

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя
(повне найменування вищого навчального закладу)

Комп'ютерно-інформаційних систем і програмної інженерії
(назва факультету)

Комп'ютерних наук
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

магістр

(назва освітнього ступеня)

на тему: Методи та засоби дослідження характеристик динамічних графічних примітивів

Виконав(ла): студент(ка) 6 курсу, групи СНм-61
спеціальності 122 «Комп'ютерні науки»

(шифр і назва спеціальності)

(підпис)

Орлінський М. В.

(прізвище та ініціали)

Керівник

(підпис)

Матійчук Л.П.

(прізвище та ініціали)

Нормоконтроль

(підпис)

Никитюк В.В.

(прізвище та ініціали)

Завідувач кафедри

(підпис)

Боднарчук І. О.

(прізвище та ініціали)

Рецензент

(підпис)

Лечаченко Т.А.

(прізвище та ініціали)

Тернопіль
2024

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет Комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра Комп'ютерних наук
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри

«_____» Боднарчук І.О.
(підпис) (прізвище та ініціали)
« » 20__ р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня магістр
(назва освітнього ступеня)

за спеціальністю 124 «Комп'ютерні науки»
(шифр і назва спеціальності)

студенту Орлінський Максим Вікторович
(прізвище, ім'я, по батькові)

1. Тема роботи Методи та засоби дослідження характеристик динамічних графічних примітивів

Керівник роботи Матійчук Любомир Павлович, д.е.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «24» листопада 2023 року № 4/7-1100.

2. Термін подання студентом завершеної роботи 27 травня 2024р.

3. Вихідні дані до роботи Наукові публікації, електронні ресурси, підручники, посібники з тематики дослідження.

4. Зміст роботи (перелік питань, які потрібно розробити) Вступ. 1. Аналіз параметрів елементів відображення інформації. 2. Моделі пікселя як динамічного примітива. 3. Розробка системи для експериментальних досліджень моделей динамічних примітивів. 4. Охорона праці та безпека в надзвичайних ситуаціях. Висновки.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

1. Тема 2. Мета роботи, об'єкт, предмет, наукова новизна. 3. Співвідношення фізичних і суб'єктивних характеристик зображень. 4. Характер зміни інтенсивності кольору динамічного примітива. Форма пікселя. 5. Зображення на екрані монітора при різних довжинах сторін пікселя. 6. Способи растрування площини пікселя. 7. Адитивний синтез нових кольорів на базі різного процентного співвідношення двох первинних кольорів 8. Графіки зміни інтенсивності, задані лінійними законами. 9. Встановлення динамічного закону на кожному із новоутворених проміжків. 10. Класифікація діаграм мови UML. 11. Діаграма прецедентів. 12. Діаграма наслідування класів. 13. Граф файлів, які включаються в модуль. 14. Блок-схема алгоритму створення динамічного примітиву та проведення його оцінювання. 15. Інтерфейс головного вікна програми. 16. Висновки. 17. Завершальний слайд.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	Сенчишин Віктор Степанович		
Безпека в надзвичайних ситуаціях	Клепчик Василь Михайлович		

7. Дата видачі завдання 24 листопада 2023 р.**КАЛЕНДАРНИЙ ПЛАН**

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	04.12.2023	
2.	Підбір наукових джерел щодо параметрів елементів відображення інформації	15.12.2023-31.11.2023	
3.	Переклад та опрацювання наукових джерел щодо параметрів елементів відображення інформації	15.01.2024-25.02.2024	
4.	Виконання дослідження щодо аналізу параметрів елементів відображення	26.02.2024-07.04.2024	
5.	Оформлення розділу «Аналіз параметрів елементів відображення інформації.»	15.04.2024-18.04.2024	
6.	Оформлення розділу «Моделі пікселя як динамічного примітива »	19.04.2024-25.04.2024	
7.	Оформлення розділу «Розробка системи для експериментальних досліджень моделей динамічних примітивів»	26.04.2024-02.05.2024	
8.	Виконання завдання до підрозділу «Охорона праці»	03.05.2024-07.05.2024	
9.	Виконання завдання до підрозділу «Безпека в надзвичайних ситуаціях»	08.05.2024-10.05.2024	
10.	Оформлення кваліфікаційної роботи	11.05.2024-14.05.2024	
11.	Нормоконтроль	15.05.2024-16.05.2024	
12.	Перевірка на плагіат	17.05.2024	
13.	Попередній захист кваліфікаційної роботи	21.05.2024	
14.	Захист кваліфікаційної роботи	30.05.2024	

Студент

(підпис)

Орлінський М. В.

(прізвище та ініціали)

Керівник роботи

(підпис)

Матійчук Л.П.

(прізвище та ініціали)

АНОТАЦІЯ

Методи та засоби дослідження характеристик динамічних графічних примітивів // Кваліфікаційна робота освітнього рівня «Магістр» // Орлінський Максим Вікторович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних наук, група СНм-61 // Тернопіль, 2024 // С.114, рис. – 50, табл. –2, додат. –3, бібліогр. –55.

Ключові слова: відображення інформації, графічні зображення, динамічні примітиви.

Кваліфікаційна робота присвячена розробці методів та засобів дослідження характеристик динамічних графічних примітивів. В першому розділі кваліфікаційної роботи освітнього рівня «Магістр» сформовані основні завдання, здійснено аналіз процесу сприйняття зображень та проведено аналіз параметрів пристроїв відображення інформації.

В другому розділі кваліфікаційної роботи здійснено обґрунтування вибору параметрів динамічних примітивів при побудові їх моделей, подається розробка математичних моделей примітива з урахуванням зміни інтенсивності кольору.

В третьому розділі кваліфікаційної роботи розроблена система для дослідження параметрів динамічних примітивів як інструментального програмного засобу, реалізовано інтерфейс програми проведено експериментальні дослідження параметрів динамічних примітивів з метою виявлення властивостей динамічних примітивів.

Об'єкт дослідження – процес відображення інформації з використанням динаміки зображень.

Предмет дослідження – методи і засоби відтворення динамічних примітивів в системах відображення інформації.

ANNOTATION

Studying methods and means of the dynamic graphic primitives characteristics // Master's degree qualification work // Orlinskyi Maksym Viktorovych // Ternopil Ivan Pulyuy National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Computer Science, SNnm-61group // Ternopil, 2024 // P.114, fig. –50, tab. -2, add. –3, bibliography -55.

Keywords: display of information, graphic images, dynamic primitives.

The qualification work is devoted to the development of methods and tools for researching the characteristics of dynamic graphic primitives. In the first section of the qualification work of the "Master's" educational level, the main tasks were formed, an analysis of the process of image perception was carried out, and an analysis of the parameters of information display devices was carried out.

In the second section of the qualification work, the justification of the choice of parameters of dynamic primitives during the construction of their models is made, the development of mathematical models of the primitive taking into account the change in color intensity is presented.

In the third section of the qualification work, a system for researching the parameters of dynamic primitives as an instrumental software tool was developed, the program interface was implemented, and experimental studies of the parameters of dynamic primitives were carried out in order to identify the properties of dynamic primitives.

The object of research is the process of displaying information using the dynamics of images.

The subject of research is methods and means of reproducing dynamic primitives in information display systems.

ЗМІСТ

ВСТУП.....	5
1 АНАЛІЗ ПАРАМЕТРІВ ЕЛЕМЕНТІВ ВІДОБРАЖЕННЯ ІНФОРМАЦІЇ	8
1.1 Аналіз процесу сприйняття зображень	8
1.2 Аналіз параметрів зображень	14
1.3 Аналіз параметрів пристроїв відображення інформації.....	18
1.4 Постановка задачі дослідження.....	27
1.5 Висновок до першого розділу	28
2 МОДЕЛІ ПКСЕЛА ЯК ДИНАМІЧНОГО ПРИМІТИВА	29
2.1 Обґрунтування вибору параметрів динамічних примітивів при побудові їх моделей	29
2.2 Розробка математичних моделей примітива з урахуванням зміни інтенсивності кольору.....	43
2.3 Висновок до другого розділу.....	60
3 РОЗРОБКА СИСТЕМИ ДЛЯ ЕКСПЕРИМЕНТАЛЬНИХ ДОСЛІДЖЕНЬ МОДЕЛЕЙ ДИНАМІЧНИХ ПРИМІТИВІВ.....	61
3.1 Вимоги до системи експериментального дослідження моделей динамічних примітивів	61
3.2 Розробка структури системи.....	65
3.3 Розробка програмних засобів системи	70
3.4 Розробка інтерфейсу системи	84
3.5 Керівництво користувача по роботі з системою.....	90
3.6 Висновок до третього розділу.....	98
4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА У НАДЗВИЧАЙНИХ СИТУАЦІЯХ.....	99
4.1 Основні принципи конструювання робочого місця користувача ЕОМ.....	99
4.2 Забезпечення захисту працівників суб'єкта господарювання від іонізуючих випромінювань	102
4.3 Висновок до четвертого розділу.....	105
ВИСНОВКИ.....	106
ПЕРЕЛІК ДЖЕРЕЛ.....	109
ДОДАТКИ.....	115

ВСТУП

Актуальність теми. Для представлення величин широко використовуються різноманітні графічні зображення. Характер зміни параметрів елементів зображень у часі може розглядатись як інформативний. У цьому випадку такі елементи можна розглядати як динамічні примітиви. Найпростішим з динамічних примітивів є піксел, що змінює свої параметри в часі.

Одним із важливих аспектів дослідження динамічних примітивів, що відтворюються реальними індикаторами, є виявлення властивостей таких індикаторів і створення моделей їх поведінки, що дає змогу провести відповідне комп'ютерне моделювання.

Застосування динамічних примітивів для відображення величин дає змогу за рахунок додаткових інформативних параметрів забезпечити скорочення апаратних витрат, вагогабаритних показників, підвищити інформативність та

Мета і задачі дослідження. Метою кваліфікаційної роботи є підвищення інформативності відображення величин за рахунок використання динаміки зображень та створення інструментального програмного засобу для дослідження параметрів динамічних примітивів.

Для досягнення мети потрібно вирішити такі задачі:

- проаналізувати процес сприйняття зображень та пристрої відображення інформації з метою виявлення їх статичних та динамічних параметрів для врахування цих параметрів при створенні моделей динамічних примітивів;
- розробити моделі динамічних примітивів з метою їх подальшого впровадження в систему дослідження параметрів динамічних примітивів;
- розробити систему для дослідження параметрів динамічних примітивів як інструментального програмного засобу;

- провести експериментальні дослідження параметрів динамічних примітивів з використанням розробленої системи з метою виявлення властивостей динамічних примітивів.

Об'єкт дослідження – процес відображення інформації з використанням динаміки зображень.

Предмет дослідження – методи і засоби відтворення динамічних примітивів в системах відображення інформації.

Наукова новизна одержаних результатів. Розроблено і застосовано принцип комплексного підходу до моделювання динамічних примітивів, що дозволило поєднати множину різновидів параметрів в одній системі.

Розроблено моделі динамічних примітивів, з урахуванням широкого спектру статичних і динамічних параметрів, які на відміну від існуючих використовують різні закони зміни інтенсивності кольору протягом періоду.

Практичне значення одержаних результатів. Отримано різновиди динамічних примітивів, які різняться за формою, розмірами, законом зміни інтенсивності кольору, що дозволяє проводити комплексний аналіз кожного статичного і динамічного параметру окремо, а також їх поєднання і взаємодію. Ці моделі можуть бути використані при побудові систем аналізу параметрів динамічних примітивів.

Розроблено програмний засіб для дослідження динамічних примітивів, який дозволяє проводити експериментальні дослідження динамічних примітивів з метою виявлення їх властивостей для подальшого застосування таких примітивів у пристроях відображення величин.

Апробація результатів кваліфікаційної роботи. Основні результати проведених досліджень обговорювались на XI науково-технічній конференції «Інформаційні моделі, системи та технології» Тернопільського національного технічного університету імені Івана Пулюя (м. Тернопіль, 13-14 грудня 2023 р.) та VII міжнародній студентській науково-технічній конференції «Природничі

та гуманітарні науки. Актуальні питання» Тернопільського національного технічного університету імені Івана Пулюя (м. Тернопіль, 25-26 квітня 2024 р.).

Публікації. Основні результати кваліфікаційної роботи опубліковано у працях конференцій(Див. додатки А, Б).

Структура й обсяг кваліфікаційної роботи. Кваліфікаційна робота складається зі вступу, чотирьох розділів, висновків, списку літератури з 55 найменувань та 3 додатків. Загальний обсяг кваліфікаційної роботи складає 114 сторінки, з них 108 сторінки основного тексту, який містить 50 рисунків та 2 таблиці.

1 АНАЛІЗ ПАРАМЕТРІВ ЕЛЕМЕНТІВ ВІДОБРАЖЕННЯ ІНФОРМАЦІЇ

1.1 Аналіз процесу сприйняття зображень

Знання про навколишній світ при безпосередньому контакті з ним одержуються не тільки через відчуття, але і через сприйняття. Обидві ці форми відображення представляють ланки єдиного процесу почуттєвого пізнання [23,55]. Вони нерозривно взаємозалежні, але кожна має також свої відмінні риси. Якщо в результаті відчуття людина одержує знання про окремі властивості, якості предмета, то сприйняття дає цілісний образ предмета чи явища [52].

Сприйняття - це відображення предметів і явищ як сукупності їх властивостей і частин при безпосередньому впливі їх на органи чуття [22]. Крім відчуттів, сприйняття включає минулий досвід людини у вигляді знань. Сприйнятий предмет співвідноситься з різними подібними, але вже знайомими предметами.

Сприйняття характеризується цілісністю. Кожна частина, яка входить в образ сприйняття, отримує певне значення лише при співставленні її з цілим і визначається ним. Сам образ сприйняття також залежить від особливостей його складових частин. Так, на основі наявного досвіду, знань, людина поєднує окремі елементи в процесі сприйняття в цілісний образ, додає їм визначену структурну оформленість.

Предмети і явища діють на людину в такому різноманітті, що вона не може їх сприймати і реагувати на них одночасно. З величезного числа впливів лише деякі ми виділяємо з великою виразністю й розумінням. Ця особливість характеризує вибірковість сприйняття.

Вибірковість сприйняття залежить від інтересів, потреб особистості [39]. Первісною і загальною можливістю сприйняття є виділення контуру предмета. І лише після того як удається виділити контур зображення з фону, починається

розрізнення форми, пропорцій, окремих деталей предмета. Виділення предмета з загального фону є необхідною умовою виразного сприйняття. Обведення контурів предмету, що сприймається, забезпечує сприйняття. Контрастність, незвичайність предметів дозволяє швидко виділити їх із фону.

Залежність змісту і спрямованості сприйняття від досвіду людини, його інтересів, ставлення до життя, установок, багатства знань називається апперцепцією [27]. Апперцепція - одне з найважливіших властивостей сприйняття. Око не сприймає само по собі. Усі види сприйняття здійснюються конкретною, живою людиною. Сприймаючи предмети, людина виражає певне відношення до них. Сприйняття буде протікати успішніше тільки при відповідній апперцепції [36].

Відображення будь-якого одиничного випадку як особливого прояву загального являє собою узагальненість сприйняття. Будь-яке явище в процесі сприйняття розуміється, осмислюється з погляду вже наявних знань, накопиченого досвіду. Ступінь узагальненості сприйняття залежить від наявних у людини системи понять, тобто від рівня й обсягу знань. Сприйняття може супроводжуватися словами. Узагальненість є вищим проявом усвідомленості людського сприйняття [34]. В сприйнятті виявляється єдність почуттєвих і логічних елементів знань, виявляється взаємозв'язок сенсорної і розумової діяльності особистості.

Під константністю розуміється відносна сталість сприйнятої величини, форми і кольору предметів при зміні відстані, ракурсу, освітленості [5]. Особливість людського сприйняття виражається в тому, що при сприйнятті предмета, який пересувається, образ, який відображується, зберігає ту саму величину. Це пояснюється константністю сприйняття. Константність сприйняття має свої межі. При великій віддаленості предметів ця властивість порушується. Константність сприйняття виявляється також при зоровому сприйнятті форми і кольору предметів. Константність сприйняття пояснюється досвідом, отриманим у процесі індивідуального розвитку особистості і має величезне практичне значення.

При сприйнятті величини і форми предметів велике значення має їх зображення на сітківці ока [38,25]. Великим предметам відповідає більше зображення, дрібним – менше. Особливість структури людського ока така, що зображення предмета, який знаходиться на далекій відстані, буде меншим, ніж зображення рівного йому предмета, розташованого близько від нас [37]. Сприйняття величини і форми предметів здійснюється при складному сполученні зорових, дотикових і мускульно-рухових відчуттів. При сприйнятті об'єму або глибини предметів основну роль відіграє бінокулярний зір [57]. При бінокулярному зорі зображення предмета падає на диспаратні, тобто на не цілком відповідні точки сітківки правого і лівого ока. Коли зображення падає на ідентичні, тобто цілком співпадаючі точки сітківки, воно сприймається як плоске. Якщо диспаратність зображення предмета занадто велика, то зображення починає двоїтися. Якщо ж диспаратність не перевищує визначеної величини, виникає сприйняття глибини. При віддаленості предметів велике значення в сприйнятті простору має взаємне розміщення світлотіней, що залежить від розташування предметів. Людина помічає ці особливості і вчиться, користуючись світлотінями, правильно визначати положення предметів у просторі.

Навколишнє середовище, впливаючи на сприйняття предмета, може стати джерелом зорової ілюзії [37]. Вона викликана психологічним законом контрасту відношення предмета і фону.

У залежності від того, наскільки цілеспрямованою буде діяльність особистості, сприйняття розділяють на навмисне і ненавмисне. Ненавмисне сприйняття може бути викликане як особливостями навколишніх предметів (їх яскравістю, розміщенням, незвичайністю), так і відповідністю цих предметів інтересам особистості. У ненавмисному сприйнятті немає заздалегідь поставленої мети. У навмисному сприйнятті людина ставить мету, додає визначені вольові зусилля, щоб краще реалізувати свій намір, довільно вибирає об'єкти сприйняття [23].

Для аналізу зображень зоровою системою людини найбільш важливими є ділянки, які відповідають його геометричним особливостям - кути, перетини, розгалуження.

Спостереження свідчать про наявність зв'язку параметрів рухів ока з геометричною структурою зображення поза залежністю від його змісту.

Кольоровий зір забезпечується колбочками в сітківці, а чутливість до малих рівнів світла - паличками. При низьких рівнях світла кольорове сприйняття втрачається [48].

Гранична реакція на зоровий подразник залежить від фонові освітленості. Мінімально помітне подразнення відповідає приблизно 60 квантам синьо-зеленого світла з довжиною хвилі 510 нм, яке падає на роговицю [53].

Палички в сітківці дають "сутінковий" зір. Він настає після адаптації ока до темряви і розрізняє максимум до $3 \cdot 10^{-5}$ ніт (кд/м^2). Для повної адаптації ока до темряви потрібно близько 45 хв.

"Світловий" зір забезпечується колбочками. Він має місце після адаптації ока до фонові яскравості мінімум у 3 ніт, тобто в стані світлової адаптації. Для цієї адаптації після збільшення яскравості потрібно близько 2 хв. [62].

Між адаптованим до світла ("світловим") і адаптованим до темряви ("сутінковим") станами чутливість ока безупинно змінюється; при зменшенні яскравості чутливість зрушується в синю область. Цей ефект відомий під назвою мезопікового стану, який існує в діапазоні від 3 до $3 \cdot 10^{-5}$ ніт. У практичному застосуванні дисплеїв частіше за все використовується "світловий" зір [56].

Сприйняття кольору характеризується трьома специфічними особливостями: світимістю або яскравістю; відтінком – помітним кольором (чисті кольори однозначно залежать від довжини хвилі); насиченістю кольору – міра інтенсивності кольору (нерозбавленість його знебарвлюючим білим світлом). Будь-яке колірне сприйняття можна відтворити належною комбінацією трьох монохромних компонентів (наприклад, червоного, зеленого

і синього кольорів) [3]. Міжнародна комісія з оптичних вимірів прийняла колориметричний стандарт, який представляє тривимірну колірну діаграму. Декартові координати цієї тривимірної діаграми введені з трьох ідеалізованих немонахроматичних основних кольорів, показаних на рисунку 1.1. Значення x , y і z розраховуються так, щоб їхня сума дорівнювала одиниці $x+y+z=1$. Це дозволяє виключити тривимірний графік.

Досить ідентифікувати x і y , щоб автоматично одержати z . Іншими словами, колірну карту можна представити у вигляді двовимірної проекції на площину xy .

Розглянемо трьохкомпонентний графік: крива, позначена z , показує інтенсивність монохроматичного світла постійної потужності, довжина хвилі якого показана на абсцисі; крива \bar{y} нормалізована відносно максимуму при 550 нм.

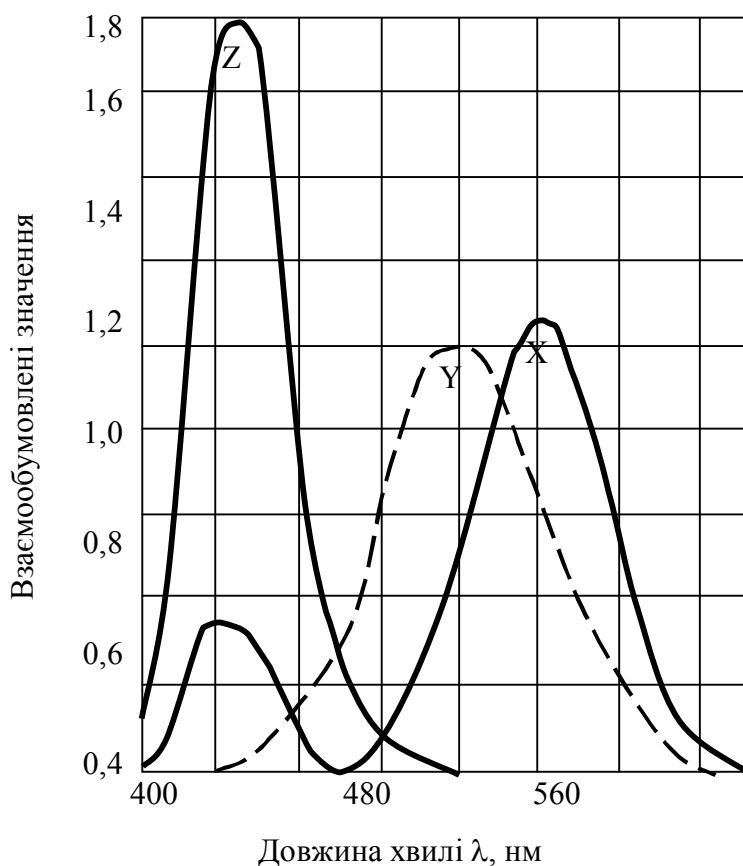


Рис 1.1. Стандартні криві змішування кольорів

Крива \bar{y} показує світлову чутливість ока. Якщо колір генерується монохроматичним джерелом, його хроматичні координати розраховуються за такими рівняннями:

$$x = \frac{X}{X + Y + Z}, y = \frac{Y}{X + Y + Z}, z = \frac{Z}{X + Y + Z}, \quad (1.1)$$

де три взаємозалежних значення X , Y і Z зчитуються при обраних довжинах хвиль з рисунку 1.1. Ці співвідношення нормалізують x , y і z таким чином, що їх сума дорівнює одиниці.

Якщо колір генерується немонохроматичним джерелом, ці три взаємообумовлених значення розраховуються інтегруванням по усьому видимому діапазону:

$$X = \int \varphi_{\lambda} x(\lambda) d\lambda, Y = \int \varphi_{\lambda} y(\lambda) d\lambda, Z = \int \varphi_{\lambda} z(\lambda) d\lambda, \quad (1.2)$$

де φ - повний спектральний потік випромінювання. Колірні координати визначаються попередніми співвідношеннями.

Розрізнення кольорів оком набагато ефективніше в нижній частині хроматичної діаграми, ніж у верхній; наприклад, сприйняття відтінку зеленого кольору отримується з більшої площі хроматичної діаграми, ніж сприйняття відтінків блакитного або рожевого кольорів [40]. Для зрівноважування цих помилок у сприйнятті кольору були запропоновані різні нові діаграми, у яких вісь x розтягнута, а верхня частина осі y стиснута.

Око здатне адаптуватися до змін рівня освітленості на 7 порядків (у 10^7 разів). Адаптація в основному відбувається в сітківці. За рахунок зміни розміру зіниці можлива адаптація до рівня освітленості лише порядку 20:1. Метою підвищення f (показника ока) у першу чергу є поліпшення різкості зображення.

Якщо зовнішнє освітлення в основному жовто-зелене (~ 570 нм), то важко сфокусуватися як на червоній ($\lambda < 650$ нм), так і на синій ($\lambda < 500$ нм) частинах зображення. Спектральна чутливість колбочок до червоного і зеленого кольору

має істотне перекриття, тоді як перекриття чутливості колбочок до синього кольору невелике [52].

1.2 Аналіз параметрів зображень

Аналіз зображення – виділення з зображення потрібної інформації за допомогою автоматичних чи напівавтоматичних приладів і систем. У літературі ця область має також інші назви: виділення даних із зображення, аналіз сцен, опис зображень, автоматичне дешифрування, розпізнавання зображень і т. ін. [42, 57].

На відміну від інших видів обробки зображень, таких, як кодування, реставрація і поліпшення якості зображень, результатом аналізу зображень звичайно є не картинка, а її числовий опис. У системах аналізу зображень часто використовуються апріорні дані про зображувані об'єкти і про їх взаємовідношення. У такій системі аналізу можуть також використовуватися методи штучного інтелекту для керування різними блоками системи й організації ефективного доступу до бази апріорних даних про об'єкти.

Ознакою зображення називається його найпростіша відмінна чи характерна властивість. Деякі ознаки є природними, тобто вони встановлюються візуальним аналізом зображення, тоді як інші, так звані штучні ознаки, отримують у результаті його спеціальної обробки або вимірювання [2]. До природних ознак відносяться яскравість і текстура різних областей зображення, форма контурів об'єктів. Гістограми розподілу яскравості і спектри просторових частот є прикладами штучних ознак.

Найважливішим параметром зображення є світлота. Світлота виражається через такі величини, як яскравість, спектральна інтенсивність випромінювання, координати кольору, які будуть називатися ознаками яскравості. Вимірювання ознак яскравості можна проводити в окремих точках зображення, або в їх межах. Наприклад, середня яскравість околиці точки (j,k) зображення розміром $(2W+1) \times (2W+1)$ елементів визначається як:

$$\bar{Y}(j, k) = \left[1 / (2W + 1)^2 \right] \sum_{m=-W}^W \sum_{n=-W}^W Y(j + m, k + n) . \quad (1.3)$$

Існує безліч різних способів визначення ознак яскравості [45,46]. Можна використовувати значення яскравості чи координат кольору безпосередньо, або перейти до нових ознак яскравості, виконавши деяке лінійне і нелінійне перетворення.

Вимірювання ознак яскравості набуває особливої важливості при виділенні об'єктів (символічний опис) і при їх маркуванні (інтерпретація).

Дискретне зображення – масив чисел $F(j, k)$ – можна представити як двовимірний випадковий процес, який описується спільними розподілами ймовірностей. Існують розроблені методи оцінки розподілів ймовірностей значень яскравості зображення. Отримані при цьому гістограми можна використовувати також для створення деякого класу ознак зображення.

Розподіл ймовірностей значень яскравості першого порядку можна визначити як:

$$P(b) \equiv \Pr\{F(j, k) = b\} , \quad (1.4)$$

де $0 \leq b \leq L - 1$ – рівні квантування. Розподіл частот першого порядку, який оцінює $P(b)$, описується простим виразом:

$$P(b) \approx N(b) / M , \quad (1.5)$$

де M – повне число елементів зображення у вікні з центром (j, k) , а $N(b)$ – число елементів у вікні, що мають рівень b . Часто, виходячи з припущення про стаціонарність, розмір вікна приймається рівним розміру всього зображення.

Форма гістограми розподілу частот дає багато інформації про властивості зображення. Наприклад, вузька гістограма вказує на низький контраст, а бімодальна гістограма припускає наявність областей різної яскравості. Існують наступні характеристики, що описують форму гістограми першого порядку:

Середнє значення:

$$\bar{b} = \sum_{b=0}^{L-1} bP(b) . \quad (1.6)$$

Дисперсія:

$$\sigma_b^2 = \sum_{b=0}^{L-1} (b - \bar{b})^2 P(b) . \quad (1.7)$$

Коефіцієнт асиметрії:

$$b_s = 1 / \sigma_b^3 \sum_{b=0}^{L-1} (b - \bar{b})^3 P(b) . \quad (1.8)$$

Коефіцієнт ексцесу:

$$b_k = 1 / \sigma_b^4 \sum_{b=0}^{L-1} (b - \bar{b})^4 P(b) - 3 . \quad (1.9)$$

Енергія:

$$b_N = \sum_{b=0}^{L-1} [P(b)]^2 . \quad (1.10)$$

Виділення ознак, представлених у виді спектральних коефіцієнтів, було досліджено в різних практичних задачах, у яких ці ознаки використовувалися в якості вхідних даних для системи розпізнавання образів.

Серед великої кількості характеристик і параметрів зображення можна виділити шість основних груп [24]. Список фізичних і психологічних (суб'єктивних) характеристик зображення приведений у табл. 1.1.

Слід зазначити труднощі виділення і вимірювання параметрів зображення і їх опис в об'єктивних фізичних величинах, що пов'язано із складністю процедур, а іноді з унікальністю апаратури, метрологічною неопрацьованістю вимірів, трудомісткістю обробки даних [24, 34].

Техніка і методика вимірювання окремих, але далеко не всіх характеристик опрацьовані зараз до рівня практичного застосування в областях телебачення, кіно і фотографії [8,25].

Практика показує, що при аналізі й оцінці зображень кращими є ті параметри і критерії, які добре співвідносяться із суб'єктивною оцінкою якості,

яку дає спостерігач. Для суб'єктивної оцінки використовуються різні види шкал, які встановлюють функціональний зв'язок між об'єктивно вимірюваною величиною і інтенсивністю її сприйняття. У зв'язку з цим постають серйозні питання психофізіологічного вимірювання оцінки якості зображень [20,29].

Таблиця 1.1 - Співвідношення фізичних і суб'єктивних характеристик зображень

Фізичні характеристики	Суб'єктивні характеристики
1. Градієнтні (на основі властивостей приграничної кривої)	1. Фокусування
2. Частотно-контрастні: - спектральні - амплітудні	2. Різкість
3. Інтенсивність (яскравість)	3. Виразність
4. Інформаційні	4. Контрастність
5. Стабільність	5. Сприйняття градації яскравості
6. Просторові: фізичні розміри лінійні і нелінійні дефекти вимірність простору	6. Щільність
	7. Тонові відтінки
	8. Колір
	9. Мінімальна (максимальна) видима яскравість
	10. Чіткість
	11. Детальність
	12. Мерехтіння
	13. Адекватність форми і величини
	14. Об'ємність

У конкретних видах зображень (наприклад, аналого-цифрові зображення) набір і комбінація характеристик (параметрів) залежать від техніки формування зображення [7], умов сприйняття, змісту і значимості інформації. Характеристики зображень мають різний зміст із позицій фізики, психології і технології. При оцінці зображення спостерігачем дослідники, згідно даним оглядів [45,46], показують, принаймні, три незалежних показника якості зображення: роздільна здатність, яка характеризує контурну різкість зображення; правильність передачі контрасту, яка характеризує ясність; тоновідтворення зображення [22,23,36].

По кожному з цих показників можна дати свою оцінку одного й того ж зображення. Вони можуть дуже відрізнятися одна від одної (наприклад,

контрастне зображення, але з поганими дрібними деталями). Постає питання про співвідношення одиничних і інтегральних (загальних, цілісних) оцінок по тому чи іншому критерію. Однак ще дотепер цілком не вирішена задача встановлення одиничних показників якості.

Таким чином, проблема якості зображень представляє великий науковий і практичний інтерес. У дослідженнях із проблеми якості зображень розвивається тенденція до об'єднання фізичного і психофізичного підходів, тому що, з одного боку, при традиційних інженерно-технічних (фізичних) підходах до оцінки зображень задовольняються потреби тільки техніки і фізичної науки, але не споживачів цих зображень. З іншого боку, психофізіологічні дослідження, спрямовані на потреби споживачів, не дозволяють перебороти труднощі узгодження параметрів техніки і людини без залучення методів і апаратури, які використовуються при фізичних дослідженнях. Простежується якісно новий етап психофізичних досліджень. Для нього характерним стане не тільки теоретичний внесок у пізнання залежностей психічних і фізичних явищ, але і велике прикладне значення, пов'язане з удосконалюванням техніки, яка використовується людиною.

1.3. Аналіз параметрів пристроїв відображення інформації

Важливим параметром світловипромінюючих пристроїв є світимість або яскравість. Цей параметр визначає наскільки успішно дисплей конкурує з зовнішнім освітленням, достатньо складна проблема – використання дисплеїв в умовах яскравого сонячного світла. Яскравість звичайно виражається в фут-ламбертах (фут·Лб), інколи в нітах, або в канделах на квадратний метр. Ці одиниці враховують спектральну чутливість усередненого людського зору. Їх називають фотометричними одиницями.

Розглянемо характеристику дисплеїв – колір. Спектральний розподіл випромінюваного світла повинен співвідноситись зі спектральною чутливістю ока, яка зображена на рис. 1.2.

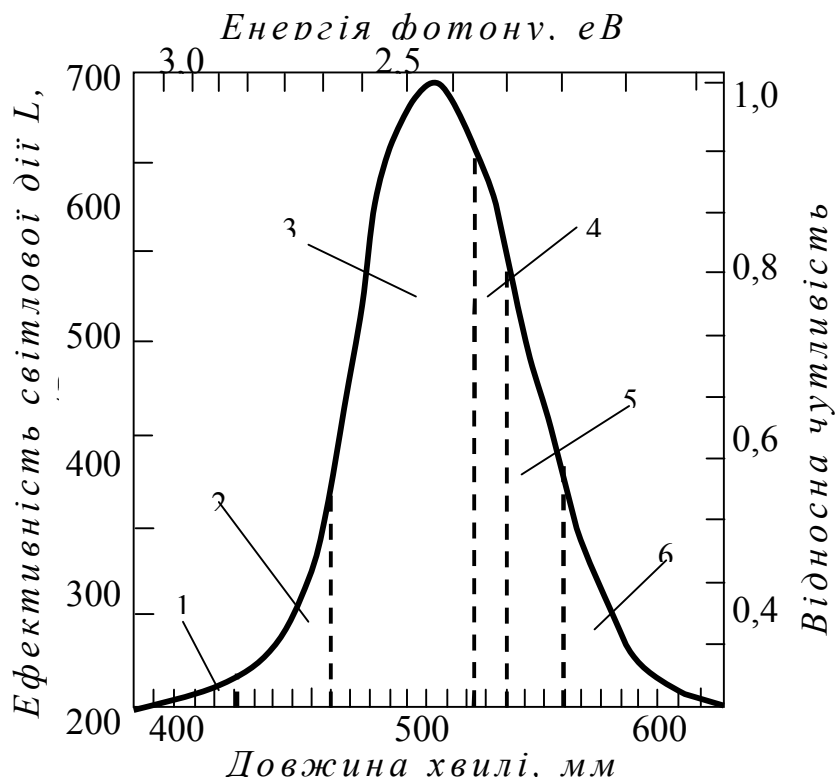


Рис 1.2. Спектральна чутливість людського ока:

1 – фіолетовий; 2 – синій; 3 – зелений; 4 – жовтий;
5 – оранжевий; 6 – червоний.

Контраст дисплея є властивістю границі світло-темно. Контраст дорівнює відношенню яскравості: $(\text{максимум} - \text{мінімум}) / \text{мінімум}$. Іноді контраст виражається у вигляді нормалізованого коефіцієнта: $(\text{максимум} - \text{мінімум}) / (\text{максимум} + \text{мінімум})$, найбільша величина якого дорівнює одиниці. Мінімальна яскравість – це яскравість дисплея у виключеному стані, залишкова яскравість за рахунок зовнішнього освітлення. Мінімальну яскравість можна зменшити за допомогою затінення фільтром, який прозорий лише для найяскравішої частини спектра. Сіра шкала визначається як число ступенів, які дають зміну інтенсивності випромінювання в $\sqrt{2}$ разів [58].

Із цієї діаграми видно, що випромінювання потужністю 100 мкВт на вузькій ділянці червоного діапазону в районі 700 нм здається менш яскравим, ніж випромінювання потужністю 100 мкВт при 556 нм, або менш яскравим ніж

широкий спектр з максимумом при 700 нм, який простягається в бік менших довжин хвиль. Червоне світло потужністю 1 мВт викликає появу болю в очах, тому існує допустимі границі потужності.

Великою перевагою є видимість дисплея у всіх напрямках. Однак часто приходится жертвувати цією можливістю для дисплеїв індивідуального користування (наприклад, у годинниках і калькуляторах). Пласмасові лінзи, які концентрують світло у вузький пучок, збільшують яскравість приладу. Контраст рідиннокристалічних дисплеїв зменшується, коли кут зору сильно відхиляється від кута нормального падіння внаслідок більшої довжини оптичного шляху в рідинному кристалі з подвійною променезаломлюваністю. Це призводить до генерації еліптично поляризованого світла, яке не вдається цілком погасити плоскими поляроїдами.

Швидкодія – швидкість відгуку дисплея. Вона визначає застосування дисплея. Очевидно, що для плоскопанельного телевізійного екрану необхідна швидкість відгуку порядку 1 мкс для послідовної адресації кожної комірки. Однак для деякого використання приймається швидкість відгуку 0,1 с, оскільки людське око не може розрізнити зміни швидше 0,1 с. Висока швидкодія пов'язана з високою потужністю розсіювання і, отже, меншою ефективністю.

Розглянемо роздільну здатність і розміри дисплеїв. Інформаційна ємність дисплея зростає зі збільшенням кількості елементів зображення, або пікселів. Отже, щільність інформації залежить від того, наскільки малим може бути розмір кожного елемента. Зовнішній вигляд і різновиди елементів широко вивчалися в телевізійній техніці [8], що призвело до визначення терміна "роздільна здатність" як кількість помітних ліній на одиницю довжини. Таким чином, роздільна здатність 20 лін./мм вважається адекватною нормальному спостереженню з відстані 20 см. Це відповідає квадратним елементам зображення зі стороною 50 мкм. Однак сприйняття зображення залежить від рівня світла і відстані до спостерігача. При збільшенні розміру елемента зображення потужність керування збільшується, а швидкодія зменшується.

Пристрої виведення інформації поділяються на декілька видів. Розглянемо найбільш поширені з них. Світловипромінюючі діоди (СВД) являють собою p - n перехід в напівпровідниковому матеріалі. При прямому зміщенні преходу має місце випромінювана рекомбінація електронів і дірок. СВД працюють при низьких напругах (<5 В) і сумісні з інтегральними схемами.

Розвиток СВД досяг досить високого рівня. Вони стали доступним комерційним виробом. Малі розміри СВД дозволяють виготовляти з однієї пластини тисячі дискретних елементів. СВД легко модулюються за допомогою зміни вхідного струму і мають час переключення менше 50 нс. Було розроблено багато напівпровідникових матеріалів, які мають емісію від червоного до синього кольору для багатофункціональних дисплеїв. СВД можуть виготовлятися в якості дискретних елементів відображення інформації різної форми (точок або сегментів), монолітних решіток індивідуально керуючих точок або частіше за все у вигляді цифрових або літерно-цифрових дисплеїв.

Для СВД існують два важливих показники якості. Першим є квантовий вихід, який визначається відношенням числа згенерованих фотонів до числа електронів, які проходять через діод, тобто ефективність перетворення приладу незалежно від зорової реакції на нього ока. Внутрішня квантова ефективність η_{int} оцінюється на p - n переході, тоді як зовнішня квантова ефективність η_{ext} оцінюється ззовні діода. Зовнішня квантова ефективність завжди менше внутрішньої внаслідок оптичних втрат, які мають місце при виході випромінювання з напівпровідника. Характерні значення η_{ext} для СВД видимого діапазону при кімнатній температурі лежать між 0,1 і 7%, тоді як при оптимальних умовах значення η_{int} можуть перевищувати 50%.

Іншим показником якості СВД є яскравість, яка є мірою зорової реакції на випромінювання, яке емітується з поверхні діода. Яскравість B в фут·ламбертах (фут·Лб), пропорційна зовнішньому квантовому виходу діода і чутливості ока, може бути розрахована виходячи з рівняння 1.11.

$$B = 1150 \cdot \frac{\eta_{ext} L J A_j}{\lambda A_s} \quad (1.11)$$

де λ - довжина хвилі емісії, мкм, J - щільність струму через перехід, А/см², а L - світлова чутливість ока, яка має максимальну величину в 680 лм/Вт для зеленого випромінювання при $\lambda=555$ нм., A_j , і A_s – площа p - n переходу і емітуючої поверхні відповідно. Відношення (A_j/A_s) дорівнює одиниці в найпростішому випадку плоского, безкорпусного, емітуючого з поверхні СВД, але в деяких випадках може бути і набагато менше одиниці (наприклад, для сферичної форми діодів, діодів із пласмасовою лінзою в корпусі і т. ін.). Оскільки яскравість діода можна змінювати зміною розміру переходу або величини струму, B звичайно нормалізується відносно щільності струму J для більш коректного порівняння характеристик різних електролюмінісцентних матеріалів. Значення яскравості, які перевищують 1000 фут·Лб, легко досягаються в комерційних СВД при звичайних щільностях струму порядку 10 А/см². Для порівняння відзначимо, що яскравість матової лампи накаливання потужністю 40 Вт складає близько 7000 фут·Лб.

З цих твержень очевидно, що необхідні ефективні СВД, які випромінюють в області максимальної чутливості ока. Оскільки верхня межа енергії емісії приблизно дорівнює ширині забороненої зони напівпровідника, то необхідні такі значення E_g , які перевищують 1,72 еВ (720 нм) і, можливо, більш близькі до 2,23 еВ (550 нм).

Важливість СВД підтверджується їх широким використанням в промислових виробках. Велика кількість дискретних СВД використовується замість неонових ламп в кабінах пілотів. Велика кількість звукової апаратури і промислових приладів мають СВД-індикатори. Для такого застосування СВД повинні випускатись різної форми для керування геометрією випромінювання: напівсферична лінза (або використання світлорозсіюючих частинок) дає широкий кут спостереження, фокусуюча лінза зручніша для отримання яскравого направленою сигналу для водія або пілота, які знаходяться в заданому положенні відносно СВД.

Основним механізмом роботи плазмової панелі є люмінесценція, яка створюється в тліючому розряді зіткненням збуджених полем іонів. При низьких тисках напруга іонізації газу складає ~ 250 В. Для підтримки розряду потрібна напруга 150 В. Цей зсув є необхідним для активації катода, оскільки регенеративний зворотній зв'язок фотонів і позитивних іонів, які бомбардують катод, збільшує електронну емісію катода.

У плазмових дисплеях постійного струму можна одержати яскравість близько 20 фут·Лб. Час ввімкнення і вимикання складає приблизно 10 мкс.

Плазмові дисплеї постійного струму без інтегрального резистора в елементі не мають пам'яті. Якщо так, то існує граничний розмір, який обмежується максимальним числом ліній, які скануються при даному рівні освітленості. Для типових значень освітленості кількість скануємих ліній виявляється рівною 200. При роботі плазмових дисплеїв постійного струму з підживленням сіру шкалу одержати легше за рахунок коефіцієнта потужності або амплітудної модуляції, ніж для плазмових дисплеїв змінного струму. Конструкція плазмових дисплеїв постійного струму набагато складніша ніж конструкція плазмових дисплеїв змінного струму.

Плазмові дисплеї змінного струму, знаходять застосування там, де необхідна плоскість корпусу, наприклад у банківському устаткуванні. Для застосування, яке вимагає високої інформативності, ідеальним є зображення дисплея, який працює протягом багатьох годин у безперервному режимі, без мерехтіння і спотворення зображення.

Газоподібний неон, який використовується у плазменній комірці змінного струму, дає випромінювання видимого діапазону. Це явище дозволяє створювати решітки комірок для дисплейних панелей видимого діапазону.

Значення яскравості в 30-75 фут · Лб типові для плазменних панелей відтворення даних з коефіцієнтом контрасту, рівним 25, і ефективністю світимості елемента 0,1- 0,5 лм/Вт.

Розглядаючи обмеження і можливості подальшого удосконалення технології плазмових панелей змінного струму розглянемо лише роздільну

здатність, колір і відтворення зображень. Розглянуті обмеження обумовлені можливостями виробництва, повною вартістю, технологічністю і сполученням усіх цих факторів.

Сучасні плазмові панелі змінного струму з високою роздільною здатністю представляють собою панель із 512×512 ліній з роздільною здатністю 2.4 лін./мм. Такі панелі звичайно виготовляються з товщиною діелектрика 40 мкм, шириною провідників 85 мкм, зазором у камері 100 мкм при значенні добутку тиску на величину зазору камери порядку 6 торр · см; у якості наповнювача використовується суміш неону 0,3% - ксенону .

Існують як монохромні, так і кольорові плазмові панелі змінного струму.

Отже, плазмові дисплеї змінного струму є плоскопанельними матричними дисплеями, розміри яких лежать у діапазоні від одного до мільйонів пікселів з роздільною здатністю до 3,7 лін./мм. Панель мініатюрна і створює зображення з високою яскравістю і контрастом без мерехтіння і похибок.

Рідиннокристалічні дисплеї (РКД) на сьогоднішній день широко розповсюджені. Рідкий кристал складається з пружного середовища з витягнутими молекулами, які можуть відбивати світло, розсіювати його й обертати площину поляризації світла. Напрямок орієнтації цих стержневидних молекул звичайно такий, що усі вони однаково орієнтуються паралельно або перпендикулярно поверхні вікна. Зображення формується введенням локального електричного поля, яке викликає зміну орієнтації молекул між електродами. У деяких нематичних кристалах обмеженої електропровідності зсув порядку 20 В викликає появу турбулентного потоку, який змінює локальну орієнтацію і приводить до динамічного розсіювання світла, тоді навколишні ділянки, які залишилися незмінними, цілком зберігають здатність світлопередачі. Скручена структура нематичних кристалів може обертати площину поляризації світла. Отже, за допомогою поляризаційних фільтрів можливе повне гасіння світла. Зсув близько 2В може повернути орієнтацію молекул у напрямку, перпендикулярному вікну; таким чином, поляризація

зникає і середовище стає прозорим. Властивості поверхні вікна досить важливі, оскільки вони керують спонтанною орієнтацією кристала.

Рідинні кристали без втрати контрасту можуть використовуватися при яскравому сонячному світлі. Потужність споживання вкрай мала, оскільки для роботи дисплея необхідна щільність струму лише декілька мкА/см². Головними недоліками РКД є обмежений кут огляду і порівняно низький контраст (5:1). Керуючу напругу, яка прикладена до РКД, можна регулювати фотопровідником, включеним послідовно з елементом. Проеціювання зображення на фотопровідник створює розподіл потенціалу на РКД, який відтворює зображення на дисплеї. Такий прилад виконує функції підсилювача світла, тому що з боку спостерігача світло може бути дуже яскравим. Кольорове зображення з високою роздільною здатністю і яскравістю 40 фут·Лб можна проектувати на великий екран.

Вдало вийшло, що найпростіший тип рідинного кристала виявився найбільш придатним для застосування в дисплеях. Усі пристрої, які послідовно з'являються в продажі, містять нематичний шар, розміщений між двома скляними підкладками, на внутрішні поверхні яких нанесені провідні плівки, які використовуються для прикладання до рідинного кристала електричного поля і зміни напрямку орієнтації молекул. Важливими характеристиками такого типу пристроїв є надзвичайно низькі рівні потужності споживання при низьких напругах, видимість дисплея при високих рівнях зовнішньої освітленості і низька собівартість.

Оптичні характеристики рідиннокристалічних приладів є наслідком фізичної анізотропії молекул і впорядкованої структури мезофази. Внаслідок макроскопічної оптичної анізотропії проходження світла через рідинний кристал керується зовнішнім електричним полем. Для більшості приладів важливим оптичним параметром служить коефіцієнт заломлення. Він має одне значення n_o (звичайний коефіцієнт) для поляризації світла перпендикулярно осі і інше значення n_e (незвичайний коефіцієнт) для світла, поляризованого паралельно осі. Оптична подвійна променезаломлюваність $n_o - n_e$ у деяких

нематиках може досягати значення 0,3, що істотно перевищує величину подвійної променезаломлюваності майже у всіх природних оптично одноосьових кристалах.

Однією з найпривабливіших цілей попередніх і сьогоднішніх робіт з рідинних кристалів є створення дешевих дисплеїв великої площі, здатних у багатьох випадках замінити катодно-променеві трубки. Принципова простота пристрою РКД і їх планарна конструкція стимулювали спроби створення великогабаритних панельних дисплеїв. Найсерйознішою проблемою в даній технології є адресація інформації до відповідної ділянки дисплея. Для малих дисплеїв можна створити окремий електричний контакт до кожного елемента дисплея. Але для дисплеїв, число елементів яких більше 50, або для двовимірних решіток з електричними перетинами створити такий окремий контакт до кожного елемента неможливо. Одне з можливих рішень – мультиплексування – має характеристики, які відповідають фізичним основам роботи рідиннокристалічних приладів.

При побудові систем відображення інформації, які використовують динамічні примітиви слід вивчити всі аспекти поведінки таких примітивів при зміні їх статичних та динамічних параметрів. Для комплексного вирішення цієї задачі може бути створена система моделей динамічних примітивів, як інструментальний програмний засіб.

При дослідженні такої системи потрібно вирішити задачі:

- розробити моделі динамічних примітивів з метою їх подальшого впровадження в систему;
- провести експериментальні дослідження параметрів динамічних примітивів з використанням розробленої системи з метою виявлення властивостей динамічних примітивів.

Характеристики, переваги і недоліки всіх розглянутих пристроїв відображення інформації наведені в табл. 1.2.

Таблиця 1.2 - Загальна таблиця характеристик дисплейних пристроїв

Тип дисплею/ Характеристика	СВД	Плазмові дисплеї		РКД
		пост. струму	змін. стр.	
Спектральний діапазон	від червоного до синього	червоний	червоний	можливі всі кольори
Яскравість	$5 \cdot 10^4$ фут·Лб	20	700	
Сіра шкала	>64		64	
Контраст	>50:1	36:1	30:1	10:1
Робоча напруга	2-20 В	250 В	250 В	2-20 В
Використана потужність	10 мВт/мм ²	мА/см ²	мА/см ²	мкА/см ²
к.к.д. (максимальний)	10			
Ефективність світимості, 1/Вт	2000	0.07	0.3	
Час відповіді ввім/вим	мкс	10 мкс		мс/0.1с
Довговічність	10 ⁴ г			2·10 ⁴ г
Розмір елемента (мін/макс)	50 мкм/10 мм	0.5 мм	0.4 мм	0.25 мм
Максимальний розмір	30 см ²	200 лін.		

1.4. Постановка задачі дослідження

Проведений аналіз дозволив виявити і систематизувати статичні і динамічні параметри зображень та пристроїв відображення інформації.

Можна перерахувати коло задач, які необхідно вирішити на теоретичному й експериментальному рівні в рамках дослідження проблеми якості зображень [24]:

- 1) провести класифікацію дисплеїв на основі специфіки зображень, які генеруються ними, розглядаючи їх як сигнальний комплекс, адресований людині;
- 2) виявити специфіку діяльності користувачів із зображеннями і важливі для них характеристики зображень;
- 3) сформулювати робоче визначення якості зображення й уточнити поняття одиничного, узагальненого й інтегрального показників якості;

- 5) виявити і опрацювати показники якості зображень і визначити їх внесок у загальну оцінку якості;
- 6) розробити досить просту і надійну методику оцінки якості зображення.

Незважаючи на розмаїття поглядів на проблему якості зображення [1,8,25,32], було прийнято робоче визначення якості зображення, яке трактується як сукупність споживчих властивостей зображення, які виявляються при його сприйнятті й оцінці спостерігачем на основі форм зорових образів. Це визначення припускає наявність у структурі якості не тільки об'єктивних, але і суб'єктивних властивостей. Показник якості зображення – це кількісна міра реакцій спостерігача при вирішенні задачі оцінки властивостей зображення, рівень якості – категоріальна міра реакцій спостерігача при вирішенні задачі оцінки властивостей зображення, а оцінка якості зображення – прояв відносин суб'єкта до об'єкта через формалізовані реакції шляхом узагальнення або диференціації властивостей зображення відповідно до поставленої задачі сприйняття.

1.5. Висновок до першого розділу

В першому розділі кваліфікаційної роботи освітнього рівня «Магістр» сформовані основні завдання, здійснено аналіз процесу сприйняття зображень та проведено аналіз параметрів пристроїв відображення інформації.

2 МОДЕЛІ ПІКСЕЛА ЯК ДИНАМІЧНОГО ПРИМІТИВА

2.1. Обґрунтування вибору параметрів динамічних примітивів при побудові їх моделей

Будь-який графічний примітив може характеризуватись статичними та динамічними параметрами . До статичних параметрів примітива можна віднести такі: форма, розміри сторін фігури, поворот (кут повороту відносно осей), масштаб, зсув відносно центру координат, колір. До динамічних параметрів, у свою чергу, відносять: закон зміни інтенсивності, тривалість періоду, скважність, кількість проміжків, характеристики проміжків [29].

В даному розділі розглянемо моделі піксела як динамічного примітива, який характеризується як статичними, так і динамічними параметрами.

За формою в рамках площини доцільно розглянути такі форми, які забезпечують регулярність при утворенні площинного інформаційного поля: еліпс (коло), трикутник (рівнобедрений, прямокутний, рівносторонній), чотирикутник (паралелепіпед, прямокутник, трапеція, квадрат), п'ятикутник, шестикутник, восьмикутник [30].

Керування характером зміни інтенсивності кольору дозволяє провести моделювання фізичних примітивів в реальних режимах їх роботи і моделювання режимів, які забезпечують найкращі показники інформативності цих примітивів. Типові випадки характеру зміни інтенсивності проілюстровані на рис. 2.1.

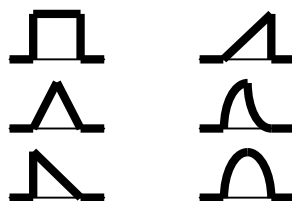


Рис.2.1 Характер зміни інтенсивності кольору динамічного примітива В світі існує багато різноманітних форм, за допомогою яких відбувається перенесення та узагальнення інформації про певний об'єкт. Форма об'єкта відіграє важливу роль при його створенні та моделюванні.

Піксел – основна елементарна складова всіх растрових зображень.

Якщо досить уважно розглянути будь-який об'єкт, то можна побачити, що він складається із набору з'єднаних між собою основних (елементарних) геометричних фігур, які забезпечують регулярність при утворенні площинного інформаційного поля. Виділимо такі елементарні геометричні фігури, які найзручніше використовувати для утворення суцільної площини. Такими фігурами є: коло, трикутник, чотирикутник, п'ятикутник та шестикутник.

В свою чергу, кожна із геометричних фігур за певними характерними ознаками поділяється на певні різновиди. Наприклад, різновидом еліпса є коло (еліпс, у якого діагоналі рівні). Коло характеризується довжиною однієї діагоналі, еліпс – довжиною двох діагоналей. Різниця в заданні цих двох фігур зумовлює необхідність їх окремого розгляду. Отже будемо розглядати форму піксела у вигляді кола і еліпса (рис. 2.2, а).

Розглянемо геометричну форму „трикутник”. Трикутники поділяють на такі підгрупи (рис. 2.2, б): прямокутні, рівнобедрені, рівносторонні. Ці види є взаємопов'язаними один з одним, оскільки прямокутний трикутник одночасно може бути і рівнобедреним.

Серед чотирикутників розрізняють такі види (рис. 2.2, в): прямокутник, квадрат, трапеція, паралелепіпед.

Розглянемо також деякі форми багатокутників: п'ятикутник, шестикутник, восьмикутник (рис. 2.2, г). Для зручності задання цих фігур оберемо форми правильних багатокутників, оскільки в такому випадку для опису фігури достатньо буде визначити одну її сторону. Форма піксела „правильний восьмикутник” є досить актуальною оскільки, при повороті на кут $+45^\circ$ або -45° отримуємо ту ж фігуру. Піксели у вигляді восьмикутника використовуються при побудові фоточутливих матриць для цифрової фото та відео техніки.



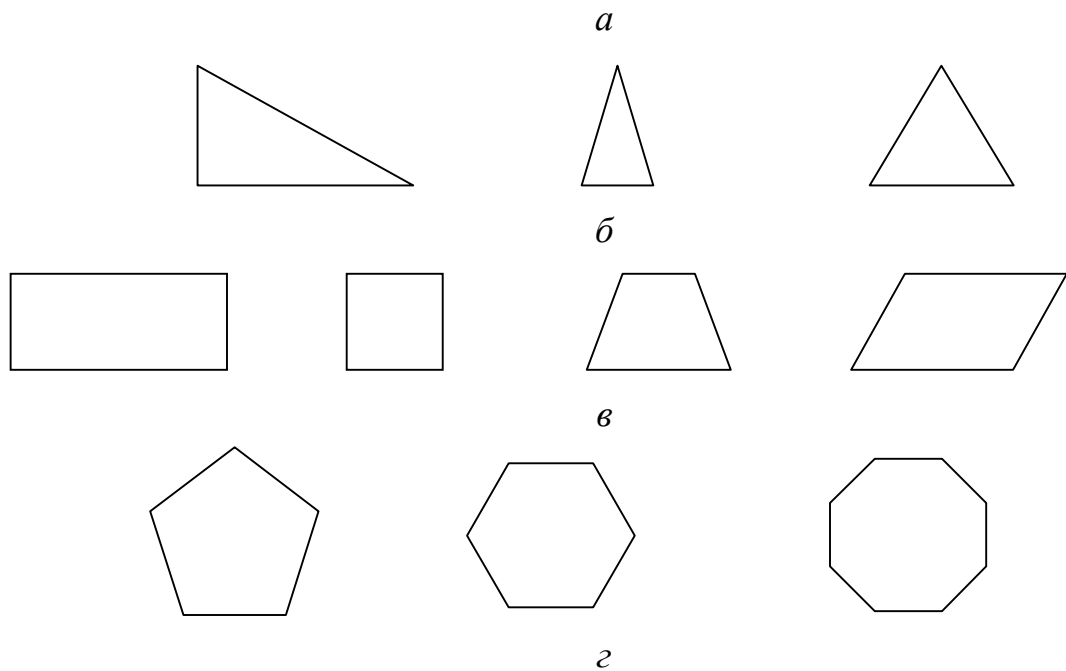


Рис. 2.2. Форми піксела:

а – коло та еліпс; б – трикутники; в – чотирикутники; г – багатокутники.

Кожна із перерахованих вище геометричних фігур має ряд параметрів, якими вона описується на площині. А саме – величина сторін, поворот, зсув, масштаб.

Величиною сторін у прямокутника будуть його довжина та ширина, які позначаються відповідно літерами a та b . Величина сторони квадрата задається довжиною однієї із його сторін a . Паралелепіпед характеризується довжиною a , шириною b , та висотою h . Величина сторони трапеції складається із довжин двох основ a та b , і висоти h . У кола величиною сторони буде радіус, який позначається r . У еліпса величинами його сторін будуть його дві діагоналі – головна a та допоміжна b . Параметри рівнобедреного трикутника задаються довжиною одного ребра a та основою b . У рівностороннього трикутника всього один параметр – довжина сторони a . Прямокутний трикутник задається двома довжинними параметрами – сторонами a та b , які є катетом і гіпотенузою. У правильного п'ятикутника, шестикутника та восьмикутника – один параметр – радіус r , кола описаного навколо фігури [51].

Величина сторін відіграла і відіграє важливу роль при побудові моделі пікселя. Про це свідчить всім відомий приклад, коли форма пікселя монітора задається у вигляді прямокутника зі сторонами a та b .

Можливі такі варіанти:

- 1) коли $a = b$, тоді піксели, із яких складається зображення, на екрані будуть мати вигляд квадрата та зображення буде пропорційним (рис. 2.3 а);
- 2) коли $a < b$, піксели, із яких складається зображення, на екрані будуть мати вигляд прямокутника, а зображення буде видовженим (рис. 2.3, б);
- 3) коли $a > b$, піксели, із яких складається зображення, на екрані будуть мати вигляд прямокутника, а зображення буде приплюснутим (рис. 2.3, в) [22].

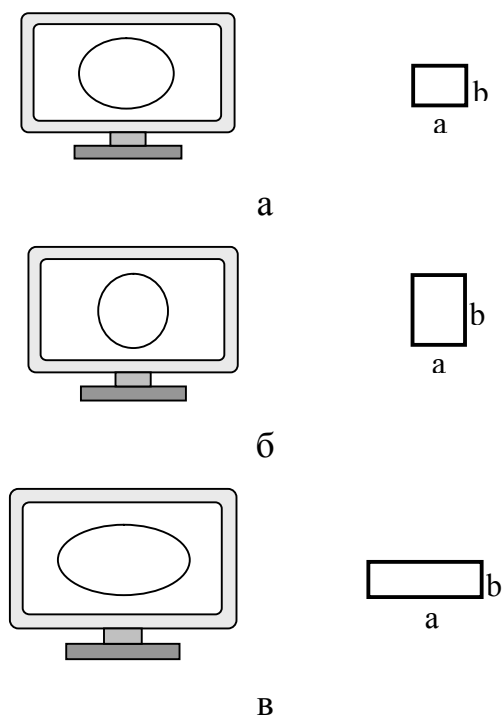


Рис. 2.3. Зображення на екрані монітора при різних довжинах сторін пікселя:
 а – довжини сторін рівні; б – довжина сторони пікселя менша за висоту;
 в – довжина сторони пікселя більша за висоту.

Коефіцієнт прямокутності пікселя характеризує реальні розміри пікселя (його відношення ширини та висоти). Розміри пікселя залежать від апаратного

та програмного забезпечення, яке використовується на різних комп'ютерних системах (платформах) – Macintosh та Windows PC. Тому для різних типів комп'ютерних платформ зображення можуть трохи відрізнятися. Наприклад, малюнок, створений на Macintosh, в системі Windows PC буде виглядати злегка видозміненим, завдяки коефіцієнту прямокутності пікселя, який впливає на диспропорцію малюнка [22]. Наприклад, розглянемо малюнок, на якому зображено коло і який створений в графічному редакторі в ОС DOS (рис. 2.3, а) та відкриємо даний графічний файл у ОС Windows (рис. 2.3, б) і на комп'ютері із платформою Macintosh (рис. 2.3, в). Всі три зображення будуть відрізнятися один від одного, оскільки у різних операційних системах та на різних платформах (різне апаратне забезпечення) будуть різні коефіцієнти прямокутності пікселя, що і спричиняє видозміну зображення [22].

Будь-яку фігуру на площині можна повернути на певний кут відносно її центру. Поворот можна виконати за годинниковою стрілкою та проти годинникової стрілки. Виконувати поворот фігур із однаковими розмірами сторін на кут, кратний 90° (наприклад, коло, квадрат), немає ніякого сенсу, оскільки ніяких змін при цьому не відбудеться, тобто форма пікселя не зміниться. А при повороті трапеції чи прямокутного трикутника можна отримати зовсім інше сприйняття геометричної форми пікселя (рис. 2.4).

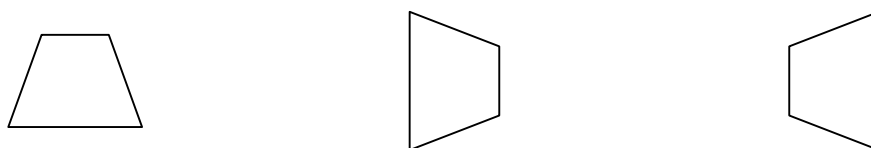


Рис. 2.4. Поворот трапеції „за” та „проти” годинникової стрілки на кут 90°

Модель динамічного примітива буде будуватися по центру області відображення примітива. Тому актуальним є дослідження сприйняття моделі пікселя при його зсуві на певну кількість одиниць відносно центру. Зсув може бути здійснений в площині у чотирьох напрямках – вліво, вправо, вгору, вниз. Зсув пікселя відіграє важливу роль при моделюванні інформаційного поля сукупності пікселів.

Масштаб використовується для візуального збільшення або зменшення графічного примітива. Масштаб задається у відносних одиницях і показує кількість пікселів на одиницю розміру. Наприклад, розмір сторони рівностороннього трикутника 2, а масштаб – 50. Тобто в одну одиницю довжини сторони буде входити 50 відносних одиниць.

Важливим статичним параметром пікселя для побудови інформаційного поля є спосіб растрівання площини пікселя. Способи растрівання імітують появу зображень в координатному напрямку (перші чотири схеми рис.2.5), в радіальному напрямку (шоста схема рис.2.5) і в діагональному напрямку (п'ята схема рис.2.5).

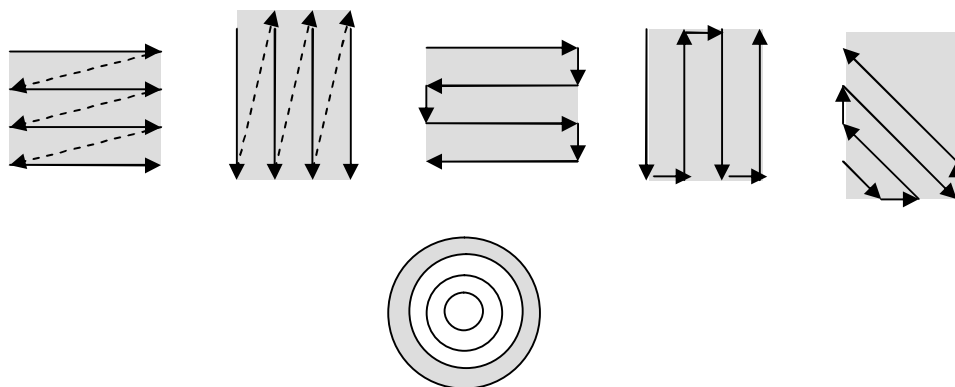


Рис.2.5. Способи растрівання площини пікселя

Останнім і досить важливим параметрами є колір примітива і колір фону. Колір примітива може бути основним і вторинним. Основний колір примітива – початковий колір пікселя при інтенсивності рівній 0 [30], вторинний колір – колір примітива при максимальній інтенсивності.

Фон – колір площини, на якій буде відображатись примітив. Всі три кольори мають бути різними, щоб не зливатися в єдине ціле.

Зорове сприйняття – сукупність процесів побудови зорового образу навколишнього світу [23]. З цих процесів простіші забезпечують сприйняття кольору, яке може зводитися до оцінки освітленості, або видимої яскравості, кольорного тону, або власне кольору, і насиченості як показника відмінності

кольору від сірого рівної із ним освітленості. При цьому основні механізми колірної сприйняття мають природній характер і реалізуються за рахунок структур, локалізованих на рівні підкіркових утворень мозку [36, 40]. Око реагує на три первинні кольори: червоний, зелений і синій. Людський мозок сприймає колір як поєднання цих трьох сигналів. Сприйняття кольору помітно змінюється залежно від зовнішніх умов. Дві людини ніколи однаково не сприйматимуть один і той же фізичний колір. Люди відрізняються один від одного навіть по чутливості до діапазону видимого світла. В даний час світло визначається як проміжне середнє при сприйнятті випромінювання об'єкту. Коли наші очі збуджуються світлом, відбитим від об'єкту, то ми сприймаємо і розпізнаємо світло як колір.

Фізичну дію кольору було багато разів підтверджено численними експериментами фізіологів і психологів. Так, М. Деревіре приводить такий опис дії кольору на психіку, даний доктором Подільським.

Зелений колір – беззаспокійливий, гіпнотичний. Він впливає на нервову систему, знімаючи дратівливість, безсоння, втому, знижує кров'яний тиск і піднімає тонус.

Блакитний колір – антисептичний. Чутливій людині блакитний допомагає більше, ніж зелений, але від його передозування виникають деяка втома і пригніченість [37].

Оранжевий колір стимулює відчуття і прискорює пульсацію крові, не впливаючи при цьому на кров'яний тиск. Він має сильну стимулюючу дію, створює відчуття благополуччя і веселості, але може спричинити втому.

Червоний колір володіє теплотою. Він стимулює мозок, ефективний при меланхолії, але в той же час легко надає дратівливу дію [22, 41].

Людина може сприймати колір двох видів: колір об'єкту, що світиться, який називають кольором свічення, і колір освітленого об'єкту, який називають кольором об'єкту. Колір об'єкту – це колір, відбитий від освітленого об'єкту. Він складається з світла, відбитого від поверхні об'єкту, а також зі світла, відбитого і розсіяного на елементах, що знаходяться під поверхнею об'єкту.

Око людини сприймає довжини хвиль в діапазоні 400 - 500 нм. як синій колір, в діапазоні 500 - 600 нм. як зелений колір і в діапазоні 600 - 700 нм. як червоний колір. У комп'ютерній індустрії ці кольори називаються трьома первинними кольорами. Для їх позначення використовується аббревіатура RGB [24, 36].

Всі кольори, що зустрічаються в природі, можна створити, змішуючи світло цих трьох довжин хвиль, варіюючи їх інтенсивності. Суміш, що складається з 100% кожного кольору, дає білий колір. Суміш 0% від кожного кольору дає відсутність світла або чорне світло [40].

Мистецтво відтворення кольору шляхом складання в різних пропорціях трьох первинних RGB кольорів називається аддитивним змішенням. Цей принцип використовується для відтворення кольору у відео і в комп'ютерних моніторах [22].

Число існуючих кольорів взагалі безмежно. Деякі пристрої, до яких можна віднести і очі людини, здатні сприймати кольори, інші пристрої здатні відтворювати кольори. Однак роблять вони це по-різному. Людське око не спроможне сприймати кольори ультрафіолетового та інфрачервоного діапазону, однак те, що воно сприймає, набагато більше, ніж може передати екран монітору, офсетний друк або фотознімок. Їх кольорове охоплення - діапазон кольорів, які можуть бути відтворені, зафіксовані або описані будь-яким чином, - менше, ніж кольорове охоплення ока людини.

Через різницю в кольорових охопленнях різних пристроїв для передачі і отримання зображень були створені декілька колірних моделей. Різноманітність була передбачена внаслідок того, що жодна з колірних моделей не є ідеальною. На екрані монітора не можна точно передати чистий блакитний та чистий жовтий кольори, а при друкуванні зовсім не передаються кольори, складові яких мають дуже низьку щільність. Існуючі колірні моделі використовуються для взаємозв'язку між пристроями з різним кольоровим охопленням [26].

У основі створення колірних моделей лежить використання універсальних мов, які дозволяють реалізувати способи точного опису кольору за допомогою стандартних математичних виразів. Без їх допомоги було б неможливо виконати жоден з етапів обробки цифрових зображень, включаючи сканування, редагування і друк.

У сучасних комп'ютерних програмах маніпуляції з кольором здійснюються за допомогою колірних моделей і режимів. Колірні моделі надають засоби для концептуального і кількісного опису кольору. Режим — це спосіб реалізації певної колірної моделі в рамках конкретної графічної програми. Колірні моделі (color model) використовуються для математичного опису певних колірних областей спектру. Більшість комп'ютерних колірних моделей заснована на використанні трьох основних кольорів, що відповідає сприйняттю кольору людським оком. Кожному основному кольору присвоюється певне значення цифрового коду, після чого вся решта кольорів визначається як комбінації основних кольорів [22].

Колірні моделі дозволяють представити колір математично. Таке уявлення завжди здаватиметься недосконалим через відмінність від нашого сприйняття. Проте вони зручні при використанні в комп'ютерних програмах для однозначного визначення кольору. Наприклад, при подачі на монітор сигналу R255 G000 B255 на будь-якому гарно відкаліброваному моніторі теоретично повинен з'явитися один і той же колір (в даному випадку пурпуровий).

Незалежно від того, що лежить в основі математичної моделі, будь-яка модель задовольняє трьома вимогам:

- реалізація визначення кольору деяким стандартним способом, незалежним від можливостей будь-якого конкретного пристрою;
- точне подання діапазону кольорів, які виводяться, оскільки ні одна множина кольорів не являється нескінченною;
- врахування механізму сприйняття кольорів – випромінювання або відбивання.

До колірних моделей відносять такі: RGB, CMY, CMYK, HSB, HSL, Lab, YIQ, YCC. За принципом дії колірні моделі можна умовно розбити на три класи:

- адитивні (RGB), основані на додаванні кольорів;
- субтрактивні (CMY, CMYK), основу яких складає операція віднімання кольорів (субтрактивний синтез);
- перцепційні (HSL, Lab, YIQ, YCC), які базуються на сприйнятті [27, 39].

У більшості колірних моделей для опису кольору використовується трьохвимірна система координат. Вона утворює колірний простір, в якому колір можна представити у вигляді точки з трьома координатами. Для опису кольору в трьохвимірному просторі Г. Грассман вивів три закони [27].

1. Тривимірність природи кольору – око реагує на три різні колірні складові.
2. Чотири кольори завжди лінійно залежні, тобто:

$$cC = rR + gG + bB, \quad (2.1)$$

де $c, r, g, b \neq 0$ — вагові коефіцієнти для кожної з складових кольору.

Для суміші двох кольорів $(cC)_1$ і $(cC)_2$ має місце рівність:

$$(cC)_1 + (cC)_2 = (rR)_1 + (gG)_1 + (bB)_1 + (rR)_2 + (gG)_2 + (bB)_2, \quad (2.2)$$

яка свідчить про те, що колір суміші випромінювань C залежить тільки від їх кольору, але не від спектрального складу.

3. Колірний простір неперервний. Якщо в суміші трьох кольорів один неперервно змінюється, а інші залишаються постійними, то колір суміші змінюватиметься неперервно.

Розглянемо переваги і недоліки адитивної моделі. Довільний адитивний колір можна отримати на основі законів Грассмана шляхом з'єднання променів світла різних кольорів. У основі цього явища лежить той факт, що більшість кольорів видимого спектру можуть бути одержані шляхом змішування в різних пропорціях трьох основних колірних компонент. Цими компонентами, які в

теорії кольору іноді називаються первинними кольорами, є червоний (Red), зелений (Green) і синій (Blue) кольори. При попарному змішуванні первинних кольорів утворюються вторинні кольори: блакитний (Cyan), пурпурний (Magenta) і жовтий (Yellow). Слід зазначити, що первинні і вторинні кольори відносяться до базових кольорів. Базовими кольорами називають кольори, за допомогою яких можна одержати практично весь спектр видимих кольорів. Для отримання нових кольорів за допомогою адитивного синтезу можна використовувати і різні комбінації з двох основних кольорів, варіювання складу яких приводить до зміни результуючого кольору. Наприклад, на базі джерел двох первинних кольорів – червоного та зеленого, інтенсивністю кожного з яких можна керувати за допомогою фільтру, можна отримати ряд нових кольорів (рис. 2.6) [27, 39].

Адитивні моделі кольорів знайшли широке використання в системах освітлення, відеосистемах, пристроях запису на фотоплівку, моніторах, сканерах та цифрових камерах [27].

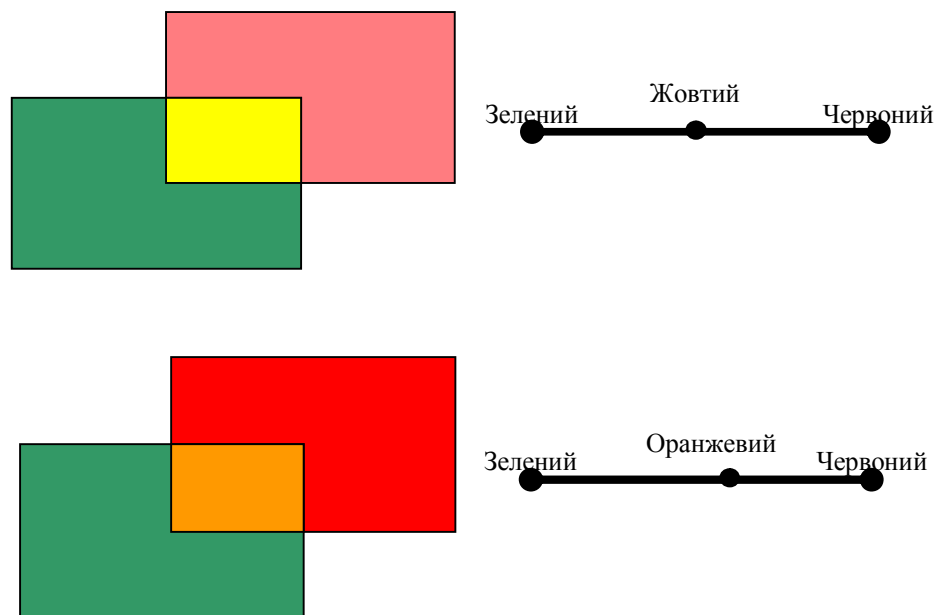


Рис. 2.6. Адитивний синтез нових кольорів на базі різного процентного співвідношення двох первинних кольорів

Розглянемо субтрактивні колірні моделі. У системі субтрактивних кольорів відбувається зворотний процес: одержання будь-якого кольору

віднімаючи інші кольори із загального променя світла. У цій системі білий колір з'являється в результаті відсутності всіх кольорів, тоді як їх присутність створює чорний колір. Система субтрактивних кольорів працює із відбитим світлом, наприклад від листа паперу. В системі субтрактивних кольорів основними є блакитний, пурпуровий та жовтий кольори (СМУ). Дані кольори є протилежними до червоного, зеленого та синього. Теоретично при змішуванні даних трьох кольорів на білому фоні (наприклад, на білому аркуші паперу) в рівних пропорціях одержуємо чорний колір. Для точного відображення чорного кольору на практиці додають ще чорний колір. А одержану колірну систему позначають аббревіатурою СМУК [3, 6].

В основі моделі RGB лежать три кольори: Red – червоний, Green – зелений і Blue – синій. Вона ідеально зручна для поверхонь, які світяться (монітор, телевізор, кольорові лампи і т. ін.). За допомогою цих трьох основних кольорів можна отримати майже весь видимий спектр (рис. 2.7). RGB називають адитивною системою змішування кольорів. Зображення на екрані монітору, а також скановані зображення кодуються в моделі RGB [1, 2, 6]. Модель можна подати у вигляді куба з ортами: $(1;0;0)$ – червоний, $(0;1;0)$ – зелений, $(0;0;1)$ – синій і початком $(0;0;0)$ – чорний. На рис. 2.8 подано куб і розподіл кольорів уздовж зазначених векторів [2].

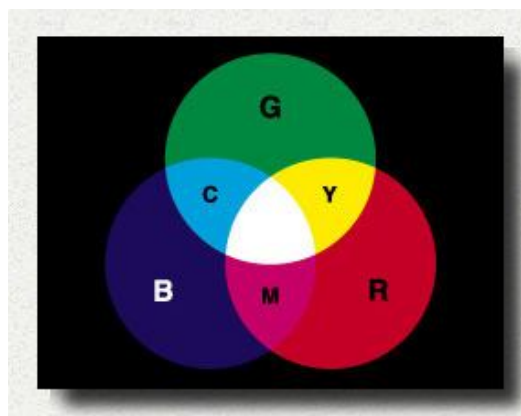


Рис. 2.7. Схема накладання кольорів в моделі RGB.

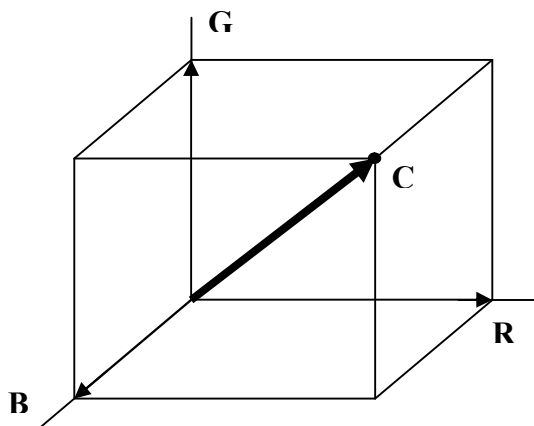


Рис. 2.8. Схема RGB моделі у вигляді куба

При використанні такої моделі будь-який колір може бути представлений у колірному просторі за допомогою вектора, який описується рівнянням:

$$cC = rR + gG + bB. \quad (2.3)$$

Рівняння ідентичне рівнянню вільного вектора в просторі, яке розглядається у векторній алгебрі. При цьому напрям вектора характеризує колір, а його модуль – яскравість. В точці початку координатних осей всі значення дорівнюють нулю (чорний колір), а в протилежній точці максимальні значення при змішуванні утворюють білий колір. Якщо дві ці точки з'єднати відрізком, то на цьому відрізку буде розташовуватись шкала відтінків від чорного до білого – сіра шкала, яку називають ахроматичною віссю. Три вершини куба дають три чисті початкові кольори. В свою чергу, кожна з трьох інших вершин між ними дає чистий, змішаний з двох основних, колір. Кожний колірний канал і сіра шкала має 256 градацій сірого.

Моделі CMY і CMYK застосовуються для типографських і принтерних фарб, плівок і т. ін. Кольори цієї моделі утворюються шляхом віднімання частини спектру падаючого світла (білого) і мають назву субтрактивні. Кольори CMY являють собою залишок віднімання основних RGB-компонентів від білого кольору. При цьому залишаються наступні кольори: Cyan – блакитний (білий колір мінус червоний), Magenta – пурпуровий (білий мінус зелений), Yellow – жовтий (білий мінус синій) (рис. 2.9) [36].

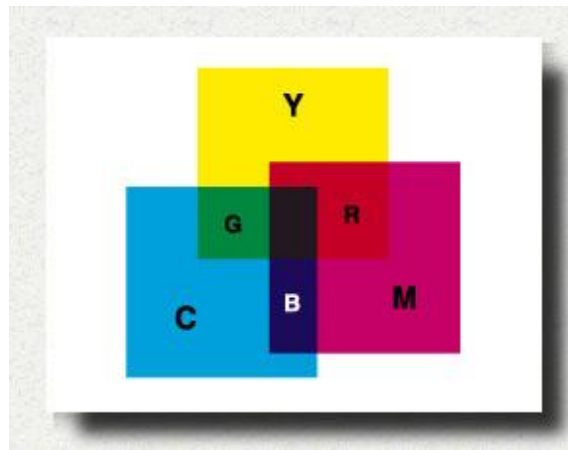


Рис. 2.9. Схема накладання кольорів в моделі СМУ

Модель СМУК є удосконаленням моделі СМУ. Вона була створена для опису процесу повнокольорового друку, наприклад, на кольоровому принтері. Пурпурова, блакитна та жовта фарби послідовно наносяться на папір в різних пропорціях. Нанесені на одне місце основні кольори змішуються, утворюючи потрібні відтінки. Однак чорний колір отримати методом змішування трьох основних кольорів не можна, в зв'язку з тим, що замість чорного виходить скоріше сіро-коричневий колір. Для отримання чистого чорного і відтінків сірого в модель СМУ був доданий новий компонент – чорний колір. Таким чином, СМУК – чотирьохканальна колірна модель. Кольорове охоплення моделі СМУК значно нижче, ніж у RGB, так як модель вона описує відображені кольори, інтенсивність яких завжди менша, ніж у випромінюючих. У СМУК білий колір слід сприймати як білий папір. При змішуванні основних компонентів з рівними значеннями виходять відтінки сірого кольору і утворюється сіра шкала. Як і модель RGB, модель СМУК є апаратно-залежною. Це означає, що при роботі з різними пристроями виводу і друку одне і те ж графічне зображення буде виглядати по-різному.

Колірна модель Lab, яка є апаратно-незалежною моделлю, оснований на сприйнятті людиною кольору. Будь-який колір в моделі Lab визначається яскравістю (brightness) і двома хроматичними компонентами - параметром "a", що змінюється в межах від зеленого до червоного, і параметром "b", що

змінюється від синього до жовтого. Яскравість в моделі Lab повністю відокремлена від кольору, що робить модель зручною для регулювання контрасту, різкості та інших тонових характеристик зображення. Ця модель є трьох каналною. Її кольорове охоплення відповідає кольоровому охопленню ока звичайної людини, а також вміщує охоплення всіх інших кольорових моделей. Цей факт дозволяє перетворювати зображення в формат Lab із, наприклад, RGB і зворотньо без зміни кольору та втрати якості зображення, що є суттєвою перевагою даної колірної моделі [36].

Провівши короткий огляд основних колірних моделей, можна зробити висновок, що для побудови системи експериментального дослідження динамічних примітивів найкраще підходить RGB-модель, оскільки вона найкраще адаптована для відображення кольорів на екрані монітору [36] та зручна для математичного представлення у програмі на мові C++ [11].

2.2. Розробка математичних моделей примітива з урахуванням зміни інтенсивності кольору

Керування характером зміни інтенсивності кольору дозволяє провести моделювання примітивів в реальних режимах їх роботи і в режимах, які в реальному житті ще не реалізовані, але забезпечують найкращі показники інформативності примітивів.

Функція інтенсивності кольору пікселя в загальному вигляді задається залежністю зміни інтенсивності від часу та описується математичною функцією [49,50]:

$$y = f(t), \quad (2.4)$$

де t – час в мілісекундах. За одиницю часу обрано мілісекунду, оскільки для визначення меж сприйнятності зміни інтенсивності динамічного примітива інтервал в секунду буде занадто довгим. Час буде встановлюватись в межах від

1 до 20000 мілісекунд. Отже, функція (2.4) буде визначена на проміжку $[a; b]$, де $a = 0$, $b \in [1..20000]$.

Інтенсивність зміни кольору примітива на проміжку $[a; b]$ може задаватися однією функцією або кількома. Для того, щоб задати декілька функцій на інтервалі, необхідно загальний проміжок $[a; b]$ розбити на підпроміжки, причому $[a; b] = [a; c_1] \cup [c_1; c_2] \cup \dots \cup [c_{n-1}; c_n] \cup [c_n; b]$, де n – загальна кількість підпроміжків. На кожному із них задається певна функція (рис. 2.10) [50].



Рис. 2.10. Розбиття інтервалу $[a; b]$ на підінтервали

Користувач системи повинен мати можливість встановлювати тривалість як цілого періоду $[a; b]$ так і підперіодів $[a; c_1]$, $[c_1; c_2]$, ... $[c_{n-1}; c_n]$, $[c_n; b]$. При встановленні часових інтервалів підперіодів потрібно, щоб обов'язково виконувалась така умова – тривалість загального періоду дорівнює сумі тривалостей підперіодів. Враховуючи це, було б доцільніше, щоб була можливість встановлювати тривалість підперіодів у відносних одиницях – частинах від загального періоду [25]. Для зручності можна встановлювати тривалість підперіодів і в абсолютних одиницях - мілісекундах, але при такому введенні повинні накладатись обмеження встановлені тривалістю всього періоду.

Скважність – це відношення тривалості проміжку, на якому задана функція інтенсивності до тривалості загального проміжку (рис. 2.11).

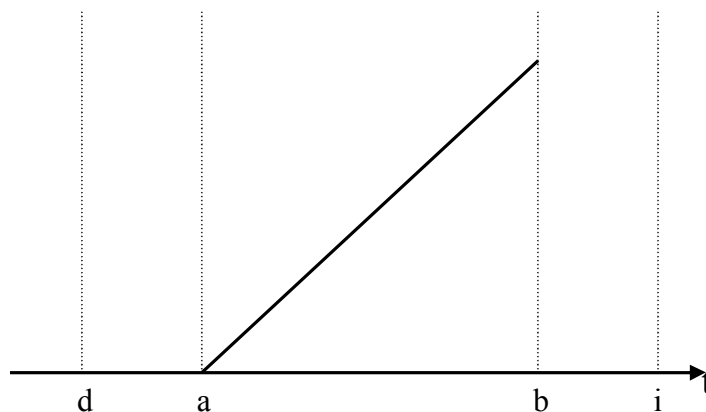


Рис. 2.11. Загальний період відображення примітива $[d,i]$

Для випадку визначення однієї функції зміни інтенсивності кольору скважність можна задати формулою:

$$S = \frac{t[a,b]}{t[d,i]}, \quad (2.5)$$

де $t[a,b]$ – тривалість проміжку, на якому задається функція зміни інтенсивності, $t[d,i]$ – загальна тривалість проміжку.

Для випадку визначення декількох функцій зміни інтенсивності кольору скважність дорівнює:

$$S = \frac{t[a,c_1] + t[c_1,c_2] \dots t[c_n,b]}{t[d,i]}, \quad (2.6)$$

оскільки,

$$t[a,b] = t[a,c_1] + t[c_1,c_2] \dots t[c_n,b]. \quad (2.7)$$

Враховуючи 2.7, не досить доцільною є можливість встановлювати час підперіодів довільно, оскільки це зумовить необхідність при введенні розраховувати загальну тривалість періоду.

Скважність будемо задавати у відсотковому форматі від 1% до 100%, оскільки таке подання є простішим для користувача. Відсоткове подання скважності є відносно абстрактним, але воно чітко пов'язує відрізок, на якому визначена функція із тривалістю періоду.

Нехай функція $y = f(t)$, яка описує закон зміни інтенсивності, визначена на всьому проміжку $[a; b]$ (рис. 2.11). Оскільки вісь X показує час, то довжина відрізка $[a; b]$ – це загальна тривалість періоду, яка задається в мілісекундах в діапазоні від 0 до 20000 мс. Користувач обирає тривалість періоду довільно, але потрібно зауважити, що при тривалості періоду менше 100 мс зміна інтенсивності сприймається дуже погано, а при тривалості періоду більше 10000 мс спостереження буде занадто виснажливим.

При встановленні значення скважності 100% – функція $y = f(t)$ буде визначена на всьому проміжку $[d; i]$. У випадку, коли значення скважності рівне 50% – функція $y = f(t)$ буде визначена лише на числовому проміжку

$$\left[\frac{d-i}{4}; \frac{3(d-i)}{4} \right] \subset [d; i].$$

На відрізку $[d, i]$, час $t(d)$ завжди рівний 0, оскільки точка d завжди буде знаходитись в початку координат, оскільки це початок періоду, а $t(i)$ – тривалість всього періоду в мілісекундах, яка встановлюється користувачем.

Виходячи із цього, проміжок, на якому функція буде визначена

$$\left[\frac{d-i}{4}; \frac{3(d-i)}{4} \right] \subset [d; i] \text{ можна подати у такому вигляді } \left[\frac{d}{4}; \frac{3d}{4} \right] \subset [d; i] \text{ (рис.}$$

2.12).

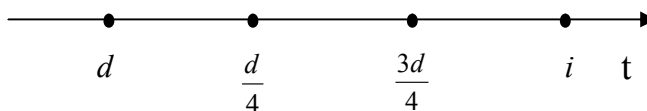


Рис. 2.12. Координати часу при скважності 50%

Аналогічно можна визначити координати періоду, на якому змінюється функція інтенсивності для будь-якого значення скважності.

Графік зміни інтенсивності завжди залежить від обраного закону, який задається певною функцією $y = f(t)$ на проміжку $[a; b]$ (рис. 2.11).

Розглянемо проміжок $[a; b]$ як одиничний відрізок $[0; 1]$. Тобто будемо вважати, що функція $y = f(t)$ визначена на одиничному числовому проміжку $[0; 1]$.

Величина відрізка $[a; b]$ – це тривалість періоду в мілісекундах. Крок переміщення по даному відрізку буде рівним 1 мс. Отже, кожне наступне значення буде більшим на 1 і його можна подати у вигляді:

$$t_n = t_{n-1} + \Delta t, \quad (2.8)$$

де $\Delta t = 1$.

Знаючи значення t в кожній точці прямої ОХ відрізка $[a; b]$ та функцію $f(t)$ задану на даному проміжку, можна знайти відповідне значення функції в точці t .

Наприклад, нехай задана лінійна функція для зміни інтенсивності $f(t) = 2t$. Тоді розрахуємо значення в будь який момент часу на проміжку $[a; b]$:

$$\begin{aligned} t_0 = 0, \text{ тоді } f(0) &= 2 \cdot 0 = 0, \\ t_1 = 1, \text{ тоді } f(1) &= 2 \cdot 1 = 2, \\ t_2 = 2, \text{ тоді } f(2) &= 2 \cdot 2 = 4, \end{aligned} \quad (2.9)$$

...

$$\begin{aligned} t_{n-1} = n-1, \text{ тоді } f(n-1) &= 2 \cdot (n-1), \\ t_n = n, \text{ тоді } f(n) &= 2 \cdot n. \end{aligned}$$

Отже, оскільки функція є або монотонно зростаючою, або монотонно спадною, свого мінімального і максимального значення вона буде набувати в точках a та b . Дана функція монотонно зростаюча, тому в точці a буде мінімум, а в точці b – максимум (рис. 2.13). Якщо функція на проміжку $[a; b]$ буде монотонно спадна, то в точці a буде максимум, а в точці b – мінімум. Мінімум функції буде задавати мінімальне значення інтенсивності кольору – основний колір примітива, а максимум – вторинний колір [51].

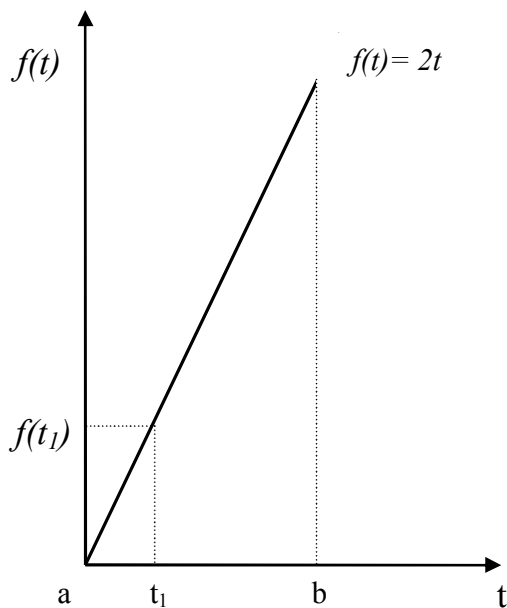


Рис. 2.13. Графік функції $f(t)=2t$, якою задається інтенсивність зміни кольору динамічного примітива

Розглянемо принцип змішування кольорів, який буде використовуватись для отримання проміжного значення кольору пікселя за певною заданою функцією. Як вже було помічено, при моделюванні динамічних примітивів буде використовуватись колірна модель RGB. Позначимо основний колір пікселя (R_1, G_1, B_1) , вторинний – (R_2, G_2, B_2) , колір пікселя в деякій проміжній точці (R_i, G_i, B_i) , функція зміни інтенсивності на проміжку відповідно $f(t)$. Значення кольору, який змінюється за певною функцією, в будь-який момент часу можна знайти, визначивши кожен складову кольору (R, G, B) окремо.

Визначимо співвідношення для розрахунку кожної компоненти кольору в залежності від основного та вторинного кольору пікселя та закону зміни інтенсивності. Нехай задані такі дані: початковий колір пікселя (R_1, G_1, B_1) , вторинний колір (R_2, G_2, B_2) , лінійна $f(t) = k \cdot t + b$ функція зміни інтенсивності кольору, тривалість інтервалу t_i . Розрахуємо колір пікселя (R_i, G_i, B_i) в деякий момент часу t_i .

Розглянемо систему рівнянь для визначення червоної компоненти кольору:

$$\begin{cases} f(t_0) = R_1; \\ f(t_n) = R_2; \\ f(t) = kt + b. \end{cases} \quad (2.10)$$

де $t_0=0$, t_n – тривалість загального періоду.

$$\begin{cases} f(t_0) = kt_0 + b; \\ f(t_0) = R_1; \end{cases} \quad (2.11)$$

$$b=R_1 \quad (2.12)$$

$$\begin{cases} f(t_n) = kt_n + b; \\ f(t_n) = R_2; \end{cases} \quad (2.13)$$

$$R_2 = kt_n + R_1 \quad (2.13)$$

$$k = \frac{R_2 - R_1}{t_n} \quad (2.14)$$

Підставивши функції (2.12) і (2.14) в рівняння прямої, отримаємо:

$$f(t_i) = \frac{R_2 - R_1}{t_n} t_i + R_1 \quad (2.15)$$

Оскільки ми розглядали інтенсивність червоного кольору в точці i то $f(t_i) = R_i$. Отже, формула для розрахунку значення червоної компоненти в будь-який момент часу буде мати вигляд:

$$R_i = \frac{R_2 - R_1}{t_n} t_i + R_1. \quad (2.16)$$

Аналогічно можна отримати функції для розрахунку значення зеленої та синьої компонент кольору:

$$G_i = \frac{G_2 - G_1}{t_n} t_i + G_1; \quad (2.17)$$

$$B_i = \frac{B_2 - B_1}{t_n} t_i + B_1. \quad (2.18)$$

Тобто складові кольору пікселя в момент часу t_i дорівнюють $(\frac{R_2 - R_1}{t_n} t_i + R_1; \frac{G_2 - G_1}{t_n} t_i + G_1; \frac{B_2 - B_1}{t_n} t_i + B_1)$.

Розглянемо розрахунок кольору пікселя на прикладі. Нехай початковий колір пікселя зелений (255;51; 0), вторинний колір – коричневий (102;153;0), функція зміни інтенсивності кольору лінійна $f(t) = k \cdot t + b$, тривалість періоду 50 с. Розрахуємо колір пікселя на другій секунді. Тобто $R_1=50$, $G_1=90$, $B_1=0$; $R_2=202$, $G_2=90$, $B_2=0$; $t_i=20$. Скориставшись рівностями (2.16), (2.17), (2.18), визначимо значення основних компонент кольору в момент часу 20 с:

$$R_i = \frac{R_2 - R_1}{t_n} t_i + R_1 = \frac{202 - 50}{50} \cdot 20 + 50 = 110. \quad (2.19)$$

Можна розглянути графік 2.14 зміни червоної компоненти кольору на інтервалі.

$$G_i = \frac{G_2 - G_1}{t_n} t_i + G_1 = \frac{90 - 90}{50} \cdot 20 + 90 = 90 \quad (2.20)$$

$$B_i = \frac{B_2 - B_1}{t_n} t_i + B_1 = \frac{0 - 0}{50} \cdot 20 + 0 = 0 \quad (2.21)$$

Функція $y = f(t)$ зміни інтенсивності кольору примітива може бути довільною. Але для подальшого дослідження розглянемо ряд елементарних математичних функцій, за якими найчастіше змінюється інтенсивність кольору в реальних пристроях.

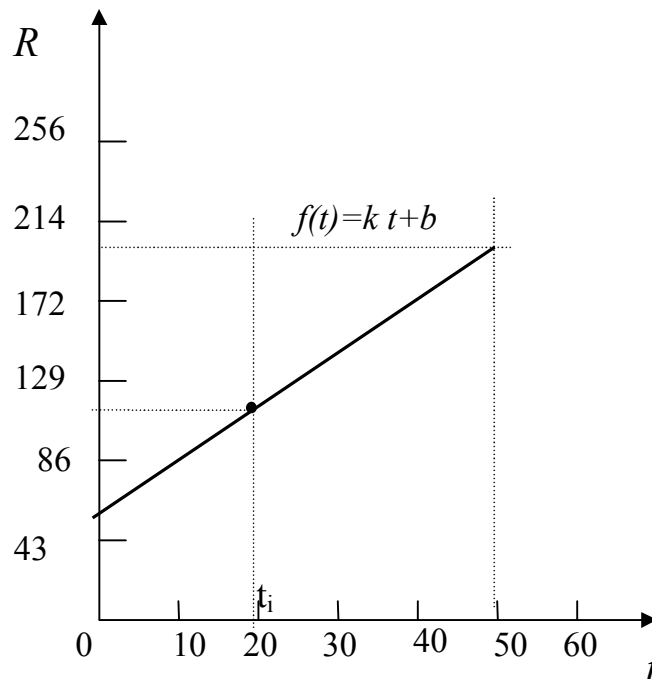


Рис. 2.14. Графік функції зміни червоної компоненти кольору примітива за лінійним законом

Отже колір піксела на 20 с буде (110;90;0) – темно салатовий.

Розглянемо спочатку найпростішу лінійну функцію $f(t) = k \cdot t + b$ [49]. Якщо коефіцієнт k дорівнює 1, то дана функція є рівномірно зростаючою, тобто із проходженням однакових проміжків часу інтенсивність збільшується на однакове значення, про що свідчить графік функції (рис. 2.15).

Коефіцієнт b характеризує зміщення графіка функції вліво чи вправо від осі Y . Якщо значення b від'ємне, то графік функції зміщується вліво, і навпаки. Розглянемо вплив коефіцієнта k на графік лінійної функції. При $k < 0$ функція є рівномірно спадаючою, тобто із проходженням однакових проміжків часу, інтенсивність зменшується на однакове значення, про що свідчить графік функції (рис. 2.16).

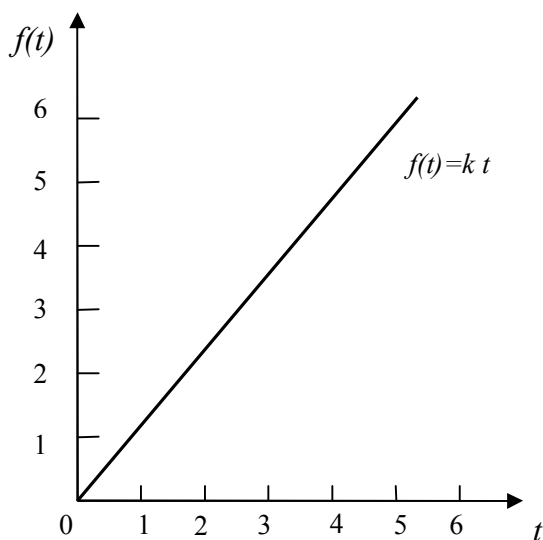


Рис. 2.15. Графік лінійної функції $f(t) = k \cdot t + b$, при $k=1$ і $b=0$

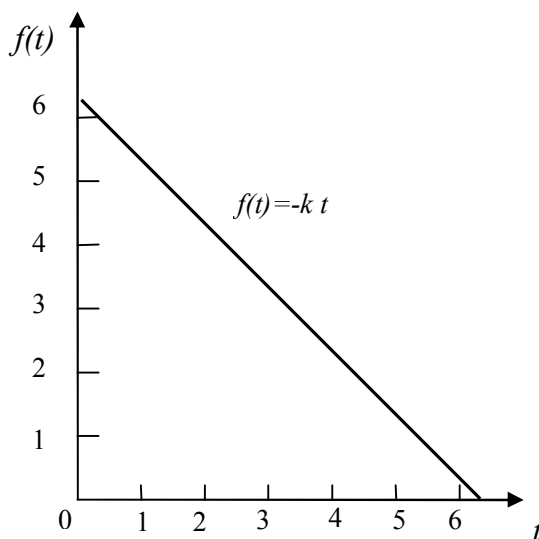


Рис. 2.16. Графік лінійної функції $f(t) = k \cdot t + b$, при $k=-1$ і $b=0$

Розглянемо функцію, яка утворена поєднанням двох лінійних функцій – монотонно зростаючої і спадаючої:

$$f(t) = \begin{cases} kt + b, & \text{якщо } t < c_1 \\ -kt + b, & \text{якщо } t > c_1 \end{cases}, \quad (2.22)$$

де c_1 – значення на осі часу, в який дана функція буде мати екстремум.

Графік цієї функції показаний на рис. 2.17.

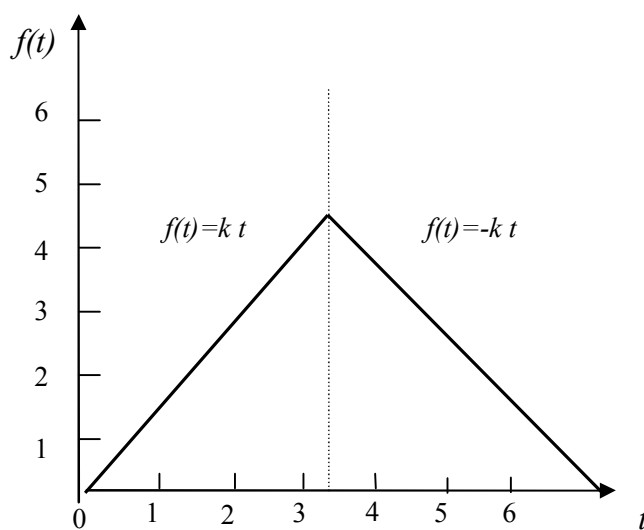


Рис. 2.17. Графік функції, яка утворена двома лінійними функціями з різним коефіцієнтом k

Всі вищеописані лінійні функції реалізовані в системі для дослідження параметрів динамічних примітивів у вигляді трьох лінійних законів зміни інтенсивності кольору.

Функція $f(t) = k \cdot t + b$ (рис. 2.15) буде описувати закон „Лінійний 1”. При його виборі графік зміни інтенсивності буде мати вигляд, який зображено на рис. 2.18, а.

Функція $f(t) = -k \cdot t + b$ (рис. 2.16) буде описувати закон „Лінійний 2”. При його виборі графік зміни інтенсивності зображений на рис. 2.18, б.

Функція 2.10 (рис. 2.17) буде описувати закон „Лінійний 3” і графік зміни інтенсивності кольору буде мати вигляд, зображений на рис. 2.18, в.

Розглянемо динамічний закон зміни інтенсивності. Він буде задаватись функцією $f(t) = \text{constant}$, тобто значення інтенсивності кольору не змінюється з часом і дорівнює певному сталому значенню.

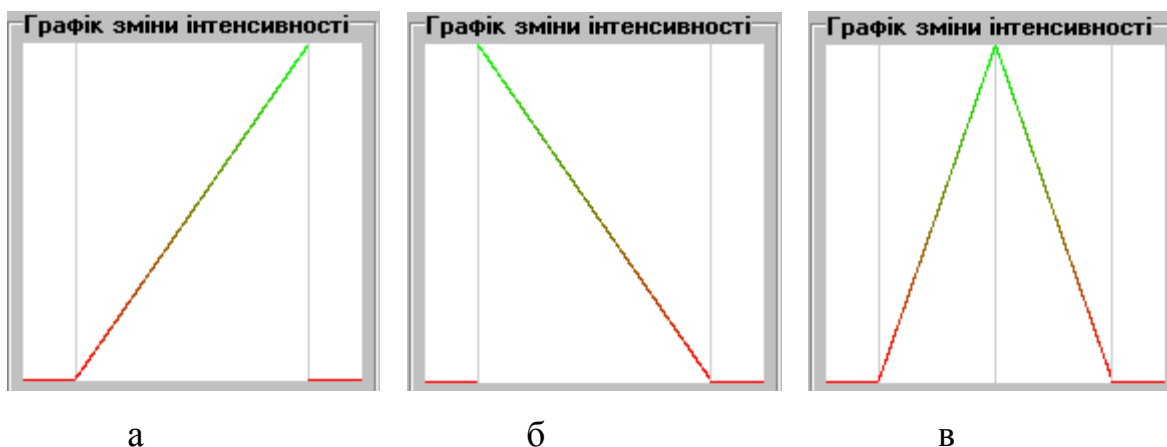


Рис. 2.18. Графіки зміни інтенсивності, задані лінійними законами:

а – закон „Лінійний 1”; б – закон „Лінійний 2”; в – закон „Лінійний 3”.

При сквапності, рівній 100%, динамічний примітив не буде змінювати кольору. Колір піксела буде дорівнювати вторинному кольору на усьому

інтервалі. При будь-якому іншому значенні скважності буде відбуватись чіткий перехід від основного до вторинного кольору піксела (рис. 2.19).

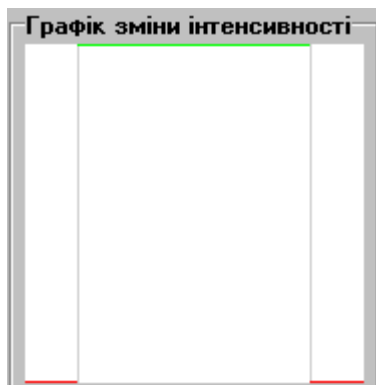


Рис. 2.19. Графік зміни інтенсивності, заданий динамічним законом

Синусоїдальному закону буде відповідати графік функції $f(t) = \sin(t)$ [50]. Дана функція визначена на всій числовій прямій і має період 2π . Тому, аналогічно до лінійної функції, будемо розглядати функцію $f(t) = \sin(t)$ лише на проміжках, де вона буде зростаючою і спадною. Відкинемо значення синусоїдальної функції, які знаходяться нижче початку координат, оскільки час не може бути від'ємним. Отже, зростає дана функція на проміжку від 0 до $\frac{\pi}{2}$

(рис. 2.20). Спадає функція $f(t) = \sin(t)$ на проміжку від $\frac{\pi}{2}$ до π (рис. 2.21).

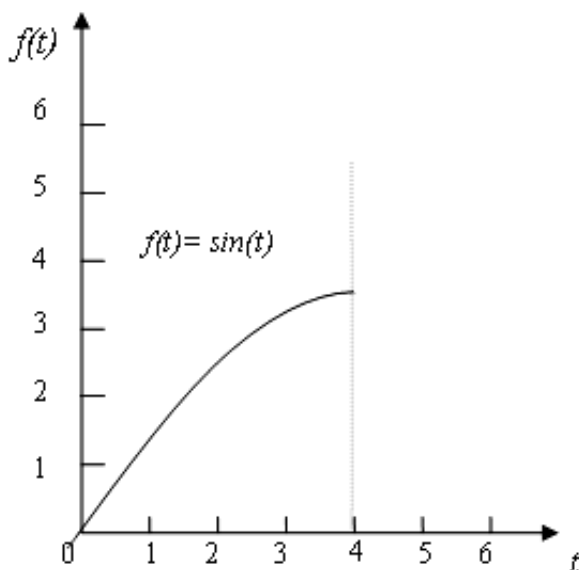


Рис. 2.20. Графік синусоїдальної функції на проміжку 0 до $\frac{\pi}{2}$

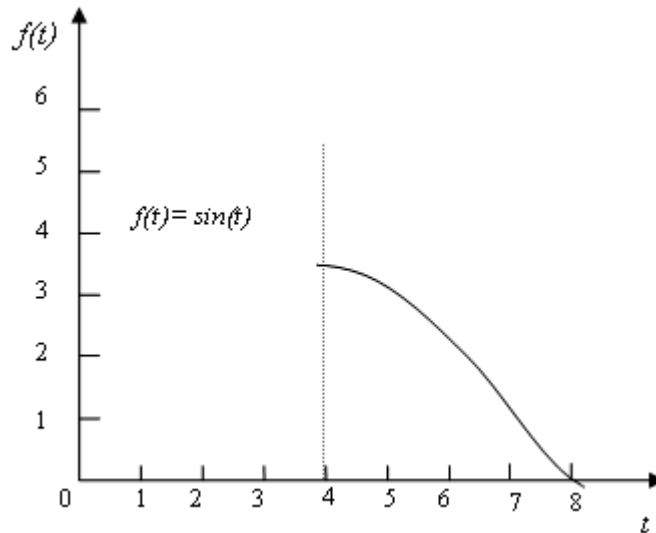


Рис. 2.21. Графік синусоїдальної функції на проміжку від $\frac{\pi}{2}$ до π

При дослідженні інтенсивності за синусоїдальним законом функція $f(t) = \sin(t)$ буде розглядатися на проміжку зростання та спадання окремо. Графік інтенсивності синусоїдального закону буде мати вигляд, зображений на рис. 2.22.



Рис. 2.22. Графіки зміни інтенсивності заданий синусоїдальним законом

Розглянемо експоненціальний закон, функція якого задається формулою [51]:

$$f(t) = \frac{e^{t \cdot s - 1}}{e^s - 1}. \quad (2.23)$$

Дана функція містить в собі коефіцієнт s , від якого залежить розлогість графіка, будемо вважати, що $s = 2$. При цьому отримаємо функцію:

$$f(t) = \frac{e^{2t-1}}{e^2 - 1}. \quad (2.24)$$

Дана функція визначена на проміжку $(-\infty; 2)$ і є на ньому завжди зростаючою. Відкинувши від'ємну частину функції (оскільки час не може мати від'ємного значення), отримаємо графік показаний на рис. 2.23.

По аналогії до попередніх функцій нам потрібно мати проміжок, на якому дана функція буде спадати. Змінюючи коефіцієнти, ми можемо утворити спадну функцію (2.25).

$$f(t) = \frac{e^{2(1-t)-1}}{e^2 - 1}. \quad (2.25)$$

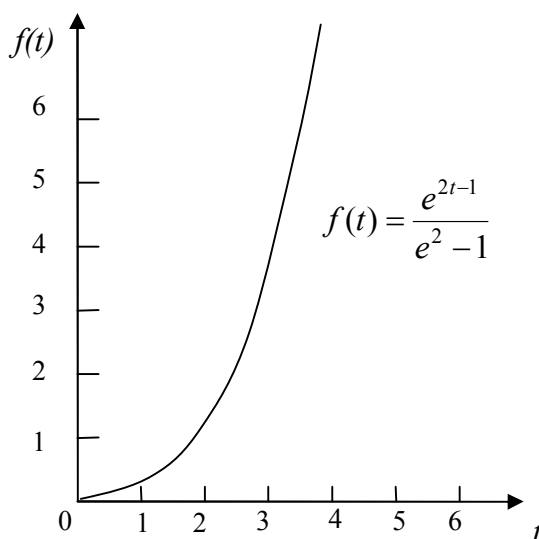


Рис. 2.23. Графік зростаючої експоненціальної функції на проміжку від 0 до 2

Дана функція визначена на проміжку $(-1; \infty)$ та є спадаючою на даному проміжку (рис. 2.24). Експоненціально зростаюча та спадна функції будуть мати такі графіки зміни інтенсивності (рис. 2.25).

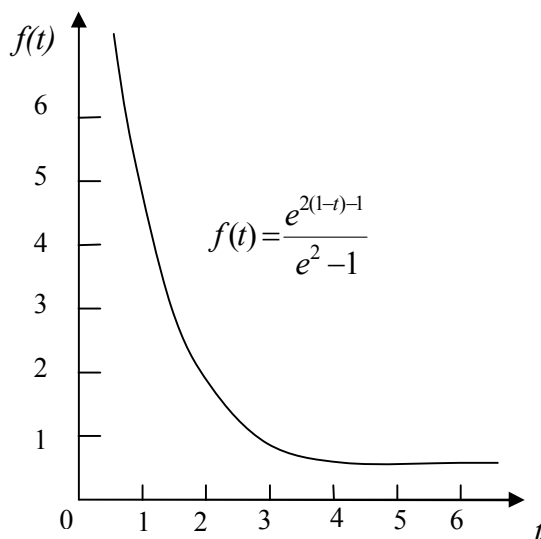


Рис. 2.24. Графік спадної експоненціальної функції на проміжку від 0 до 2

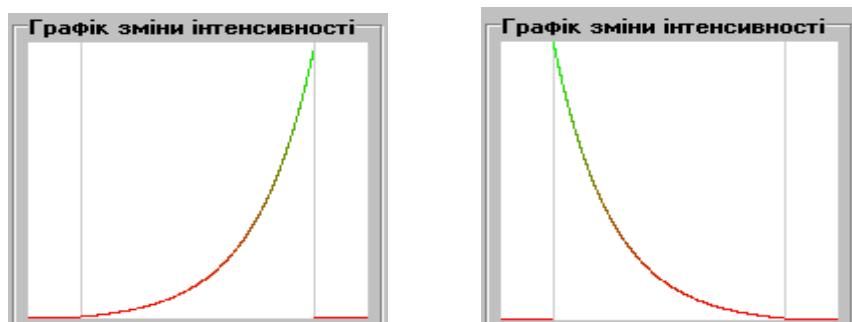


Рис. 2.25. Графіки експоненціально зростаючої та спадної функції зміни інтенсивності кольору

Доцільним також буде розглянути обернені функції до описаних вище експоненціальних. Такими функціями будуть відповідно логарифмічно зростаюча та спадна функції, які будуть мати графіки зміни інтенсивності кольору, показані на рис. 2.26.

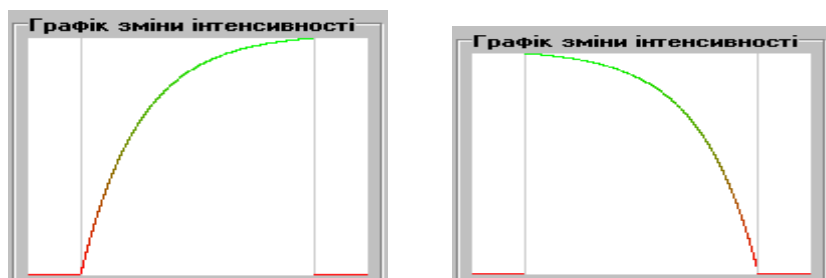


Рис. 2.26. Графіки логарифмічно зростаючої та спадної функції зміни інтенсивності кольору

Досить важливо розглянути можливість створення нової функції для інтенсивності кольору, яка б утворювалась з вище перерахованих законів. Таку можливість будемо називати „Конструктор”. Основною метою створення „Конструктора” є його використання для моделювання довільного закону зміни інтенсивності. Принцип його роботи полягає у додаванні до початкового проміжку нового інтервалу та визначення на кожному з проміжків довільної елементарної функції. Тобто, до часового проміжку $[a_1; b_1]$, буде додано ще один часовий проміжок $[b_1; a_2]$ такої ж тривалості. В такому випадку тривалість періоду зросте вдвічі за рахунок величини нового відрізка. При додаванні третього відрізка виконуються аналогічні дії. Отже, проміжок, на якому буде визначена функція, буде мати такий вигляд:

$$[a_1; b_1] + [b_1; a_2] + [a_2; b_3] \quad (2.26)$$

Його тривалість буде в три рази більшою за тривалість відрізка $[a_1; b_1]$. Для зручності, новоутворений відрізок можна подати у такому вигляді:

$$[a; c_1] + [c_1; c_2] + [c_2; b] \quad (2.27)$$

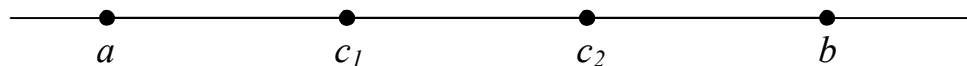


Рис. 2.27. Схема додавання проміжків

При необхідності новоутворений відрізок можна знищити. При додаванні нового відрізка його тривалість буде рівною тривалості загального проміжку, яка встановлена в даний момент. Тобто, якщо при додаванні нового відрізка період тривалості рівний 1,2 с, то новий відрізок добавиться із тривалістю 1,2 с. Тривалість кожного проміжку можна змінювати, при цьому тривалість загального проміжку буде перераховуватись.

Розділивши відрізок $[a; b]$ на проміжки, користувач на кожному із них може вибрати один із десяти динамічних законів: „Лінійній 1”, „Лінійній 2”, „Синусоїдній 1”, „Синусоїдній 2”, „Дискретний 1”, „Дискретний 2”, „Експонентний 1”, „Експонентний 2”, „Експонентний 3”, „Експонентний 4”.

Після встановлення на кожному із проміжків певного закону буде створено новий динамічний закон, який буде задаватися складною функцією виду:

$$y = \begin{cases} f_1(x), \text{ якщо } x \in [a; c_1) \\ f_2(x), \text{ якщо } x \in [c_1; c_2) \\ \dots \\ f_{n-1}(x), \text{ якщо } x \in [c_{n-2}; c_n) \\ f_n(x), \text{ якщо } x \in [c_{n-1}; b] \end{cases} \quad (2.28)$$

Наприклад, якщо тривалість загального інтервалу дорівнює 1,5 с, то він розбивається на три рівних проміжки по 0,5 с (рис. 2.27). На кожному з них можна встановити певний закон, який задається відповідною функцією (рис. 2.28). В результаті отримуємо новий закон, за яким будується графік зміни інтенсивності (рис. 2.29).

Як видно із рис. 2.28, користувач може створювати довільний динамічний закон, при цьому змінюючи величину кожного інтервалу та встановлюючи будь-яку із десяти зарезервованих елементарних функцій, які описують динамічні закони.

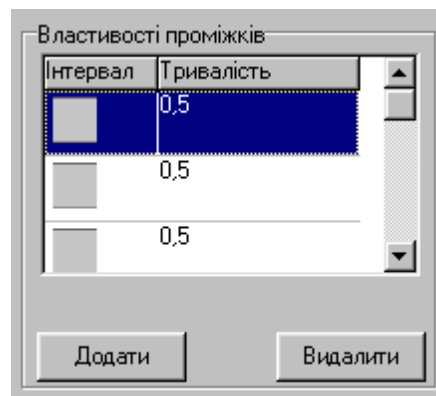


Рис. 2.27. Розбиття відрізка $[a; b]$ на три рівних проміжки

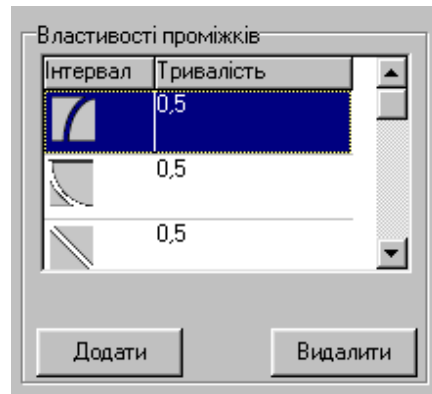


Рис. 2.28. Встановлення динамічного закону на кожному із новоутворених проміжків

2.3. Висновок до другого розділу

В другому розділі кваліфікаційної роботи здійснено обґрунтування вибору параметрів динамічних примітивів при побудові їх моделей, подається розробка математичних моделей примітива з урахуванням зміни інтенсивності кольору.

3 РОЗРОБКА СИСТЕМИ ДЛЯ ЕКСПЕРИМЕНТАЛЬНИХ ДОСЛІДЖЕНЬ МОДЕЛЕЙ ДИНАМІЧНИХ ПРИМІТИВІВ

3.1. Вимоги до системи експериментального дослідження моделей динамічних примітивів

Дослідження моделей динамічних примітивів є досить складним процесом. Спочатку необхідно задати статичні та динамічні характеристики примітива, потім проаналізувати встановлені параметри у їх поєднанні в динамічній моделі і в результаті оцінити сприйняття даної моделі у вигляді коефіцієнта сприйняття [29, 30].

Введемо визначення коефіцієнта сприйняття як коефіцієнта, який у відсотках визначає сприйнятливість динамічного примітива людиною, яка проводить експеримент, при конкретних параметрах моделі примітива.

Для експериментального дослідження сприйнятності динамічних примітивів при конкретних значеннях параметрів пристроїв відображення інформації необхідно побудувати систему, яка дозволяє моделювати динамічний примітив із сукупністю його статистичних (форма, розміри сторін фігури, поворот, масштаб, зсув відносно центру координат, колір) та динамічних (тривалість пероду, сквапність) характеристик. Також потрібно промоделювати зміну інтенсивності кольору за певними законами, які описуються математичними функціями, розглянутими у другому розділі [28, 31].

Створення такої системи дасть змогу зробити порівняльну характеристику динамічних та статистичних параметрів динамічного об'єкта, який в свою чергу є моделлю піксела, на основі відповідних коефіцієнтів сприйняття.

Проведення даного дослідження є досить актуальним, оскільки на даному етапі розвитку пристроїв виведення інформації розглядаються різноманітні форми та моделі піксела.

Піксел – це не лише мінімальна складова пристроїв для виведення інформації, таких як плазмові дисплеї змінного струму, рідиннокристалічні та електронні дисплеї, а й основна складова світлочутливих матриць, які використовуються у цифрових фото- та відеокамерах [27].

При проектуванні даної системи, у відповідності до матеріалу, який викладений в другому розділі роботи, було впроваджено всі розглянуті значення і функції для статичних та динамічних параметрів моделі піксела [31].

Основними вимогами до системи, яка забезпечує експериментальне дослідження динамічних примітивів на базі персонального комп'ютера, є:

- наявність рядка меню та панелі інструментів, які забезпечують типовий віконний інтерфейс;
- наявність інформаційного поля, в якому відтворюються динамічні примітиви;
- наявність спеціальних командних панелей, в яких настраюються відповідні параметри;
- вмонтовані моделі динамічних пікселів з відповідними законами зміни їх параметрів;
- наявність можливості введення експертних оцінок параметрів в інтерактивному режимі;
- наявність підсистеми обробки параметрів з врахуванням експертних оцінок;
- зручне графічне представлення і можливість друку результатів аналізу сприйняття статичних і динамічних характеристик;

Завдання режимів повинно бути гнучким (надавати широкі можливості) і одночасно не обтяжливим з точки зору ускладнення інтерфейсу і зручності роботи з ним.

Для зручності користування системою моделювання повинна передбачатись можливість збереження всіх вибраних і введених параметрів в

окремий файл. При необхідності цей файл можна буде відкрити для перегляду і завантажити всі параметри у систему [31].

Система експериментального дослідження містить: підсистему для дослідження статичних і динамічних параметрів динамічних примітивів, підсистему для відтворення величин динамічними примітивами та підсистему для введення і аналізу експертних оцінок параметрів.

Підсистема для дослідження параметрів включає можливість налаштування різноманітних статичних (форма, розмір, кут повороту фігури, колір) і динамічних (закон зміни інтенсивності, тривалості періоду, скважність) параметрів. Ці параметри можна застосовувати як до одного пікселя окремо, так і поєднувати різноманітні їх комбінації.

Підсистема для відтворення величин динамічними примітивами включає відтворення величин за допомогою зміни різноманітної сукупності показників на одному пікселі.

Підсистема для введення і аналізу експертних оцінок параметрів дозволяє провести аналіз параметрів моделі в залежності від їх сприйняття людиною і подати результати у графічному вигляді. Це дає змогу визначити діапазон сприйнятності кожного параметра. Важливим є також визначення значень параметрів, при яких досягається найкраща або найгірша сприйнятність, і характеру зміни коефіцієнта сприйняття в залежності від зміни одного або декількох параметрів динамічної моделі. На основі отриманих результатів створюються оптимальні для сприйняття моделі динамічних примітивів.

З метою впорядкування процесу дослідження властивостей динамічних примітивів доцільно створити підсистему обробки результатів експертного оцінювання параметрів з такими можливостями:

- наявність можливості введення значення експертної оцінки параметра (в вигляді коефіцієнта сприйнятності) при певних значеннях цього параметра;
- обробки параметрів (розрахунок проміжних значень між ключовими точками), визначення комбінацій та інтегральних оцінок;

- надання результатів у відповідному вигляді (таблиці, графіки, діаграми), з вказуванням значень кожного з параметрів.

При розробці інтерфейсу системи потрібно врахувати розташування основних блоків та елементів у вікні програми, які мають відповідати таким вимогам:

- вікно експертних оцінок необхідно розмістити біля вікна з демонстрацією примітива для того, щоб користувач, спостерігаючи примітив, не розсіював увагу і одразу після спостереження міг ввести коефіцієнт сприйняття;
- блок настроювання динамічних параметрів примітива краще розміщувати в безпосередній близькості з вікном, яке демонструє закон зміни інтенсивності. Це надасть можливість, змінюючи динамічні параметри, спостерігати зміни на графіку;
- настроювання всіх параметрів краще розмістити в одній частині вікна програми, оскільки користувач зазвичай вводить всі параметри по черзі, починаючи з статичних і закінчуючи динамічними;
- розміщуючи блоки інтерфейсу, потрібно передбачити можливість приховування блоків настроювання динаміки і статички, оскільки користувач системи, аналізуючи вхідні дані, може суб'єктивно оцінювати модель динамічного примітива;
- вікно для демонстрації примітива доцільніше розташовувати неподалік від графіка закону зміни інтенсивності;
- при виборі певного варіанту інтерфейсу потрібно також врахувати об'єм інформації який буде представляти блок „Експертна оцінка”.

Враховуючи всі перераховані вище пункти, вважаємо за доцільним варіант інтерфейсу, представлений на рис. 3.1.

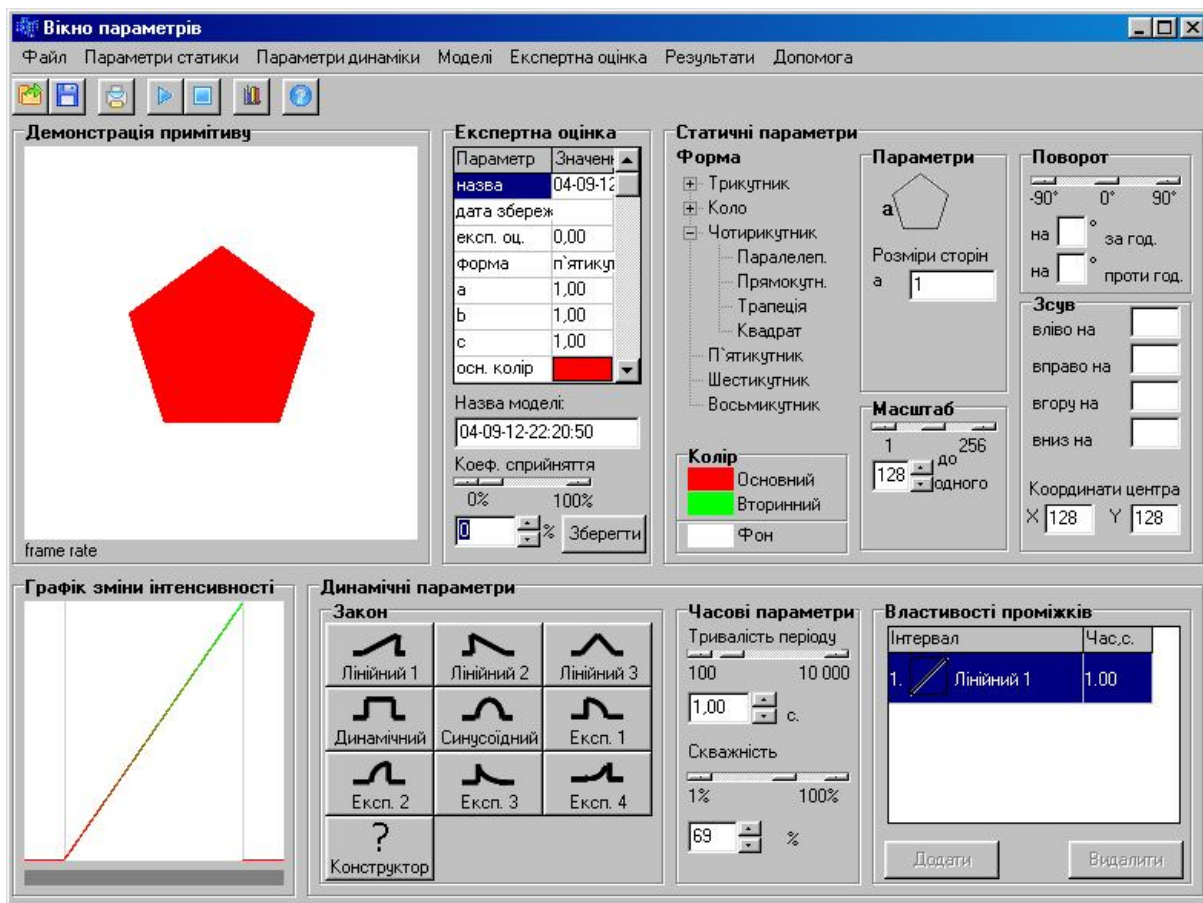


Рис. 3.1. Інтерфейс головного вікна програми

3.2. Розробка структури системи

Розробку програмного забезпечення розділимо на два етапи:

- 1) етап проектування програмного забезпечення;
- 2) реалізація розробленої моделі.

На етапі розробки програмного забезпечення здійснюється проектування моделі програмного забезпечення. У світлі розглянутих видів моделей, модель програмного забезпечення є переважно структурною.

Проектування програмного забезпечення (ПО) будемо здійснювати за допомогою уніфікованої мови моделювання UML [17,18]. UML є мовою візуального проектування, яка розроблена спеціально для специфікації, візуалізації, проектування та документування компонентів програмного забезпечення, бізнес-процесів та інших систем [21,63].

На даний час засоби проектування активно розвиваються. Проектування ПО є невід'ємною частиною якісного програмного забезпечення. Найбільшого

успіху у автоматизації процесу розробки програмного забезпечення досягла фірма Rational – вона є визначним лідером у даному напрямку [19]. Зокрема даною фірмою випущено пакет програм Rational Suite Enterprise, що складається з інтегрованих між собою програм, які дозволяють проектувати програмне забезпечення, слідкувати за його розробкою, документуванням та проводити автоматизоване тестування програмного забезпечення. Для проектування з даного пакету програм призначений продукт Rational Rose. Програмний продукт Rational Rose містить реалізацію мови UML та має графічну оболонку для проектування. Rational Rose дозволяє реалізовувати усі основні діаграми мови UML. Розглянемо класифікацію діаграм (рис. 3.2).

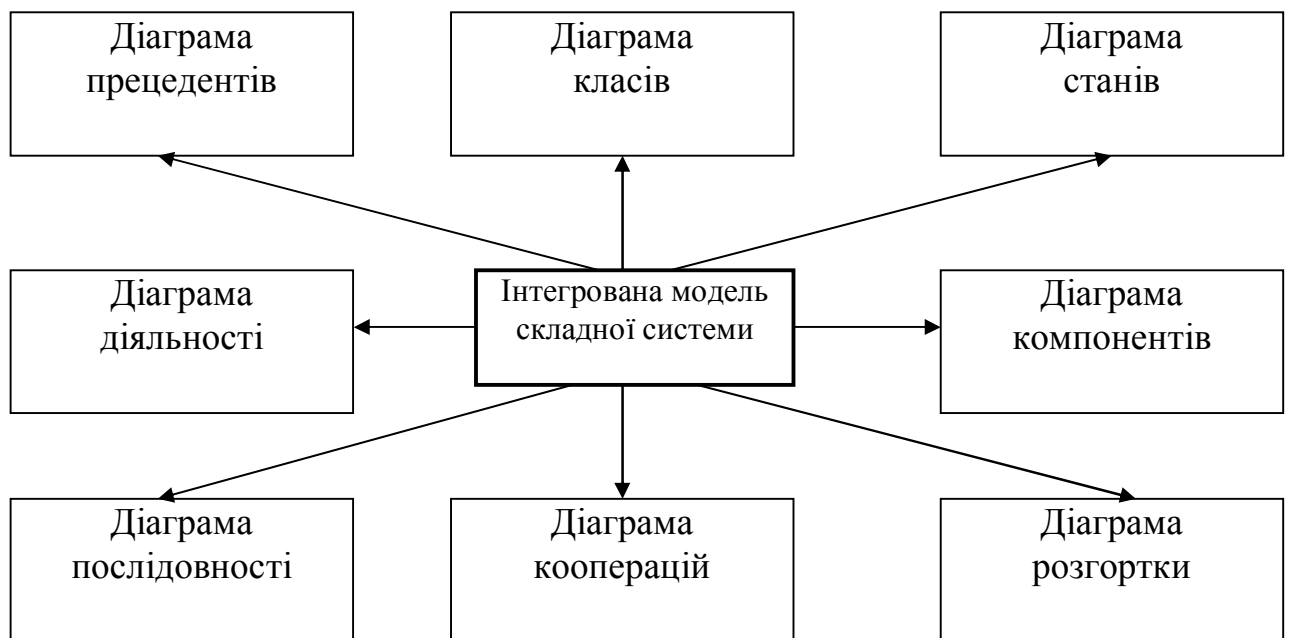


Рис. 3.2. Класифікація діаграм мови UML

Кожна з цих діаграм деталізує та конкретизує різні представлення про модель системи у термінах мови UML. Діаграма варіантів є найбільш загальною моделлю, та є вихідною для побудови усіх інших моделей.

Діаграма класів відображає статичні аспекти структури побудови складної системи. Діаграма поведінки відображає динамічні аспекти поведінки складної системи. Діаграма станів описує сукупність станів та переходів в середині системи [18]. Розробка усіх діаграм здійснюється на основі теорії мови UML [17].

Для розробки програмного забезпечення оптимізації в умовах невизначеності розглянемо наступні діаграми:

- діаграму прецедентів;
- діаграми класів;
- діаграму станів.

Діаграма прецедентів зображена на рис. 3.3. Дана діаграма призначена для опису функціонального призначення ПО. Діаграма складається з сутностей та акторів. Під актором розуміють довільна сутність, що взаємодіє з системою зовні. Прецедентом називається сервіс, який система надає актору.

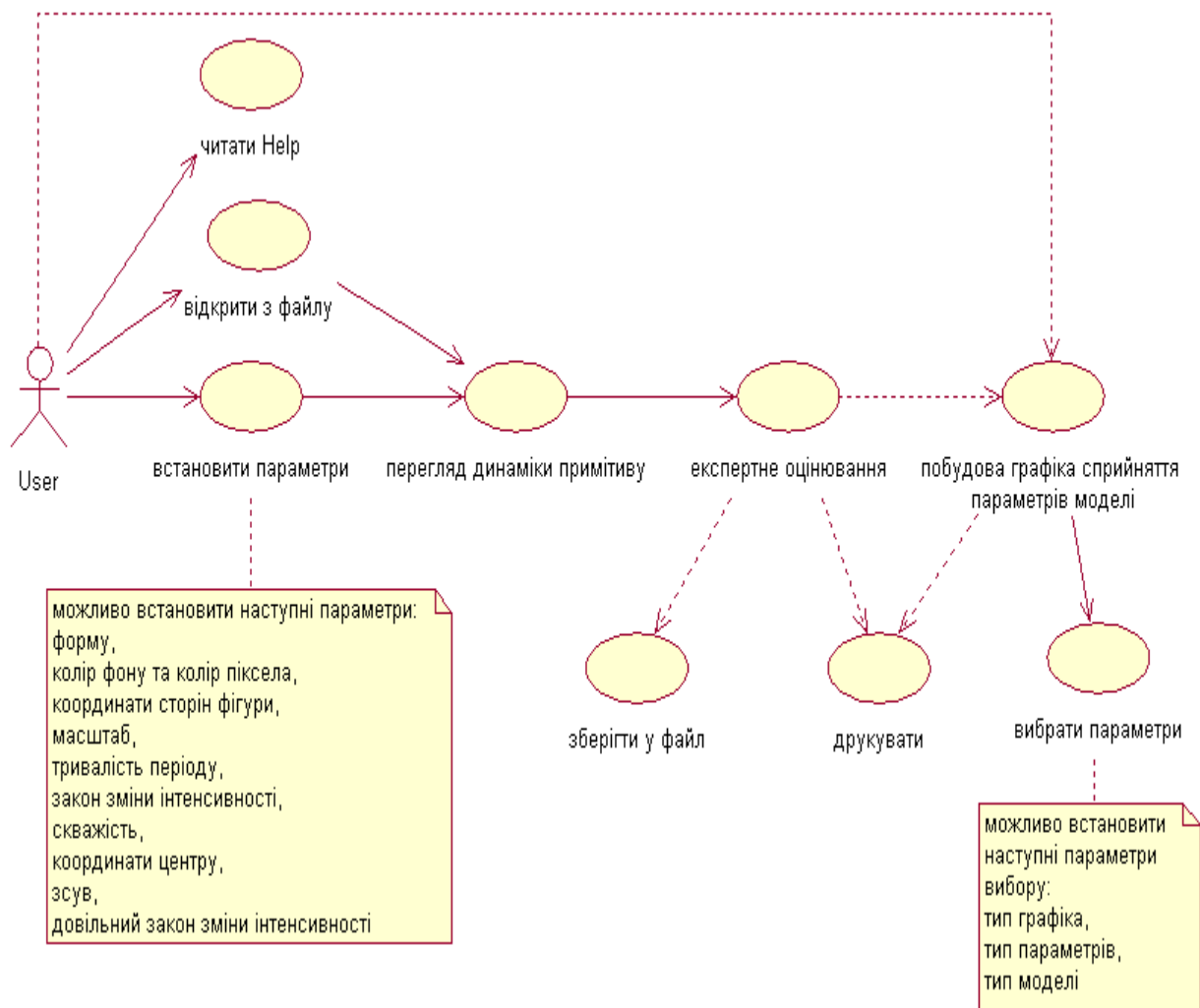


Рис. 3.3. Діаграма прецедентів

На рис. 3.3 актором є користувач ПО. Прецедентами є введення даних з клавіатури або файлів та оптимізація системи. При цьому сервіс встановлення параметрів включає в себе встановлення таких параметрів, як форма, розміри сторін примітиву, колір фону та колір пікселя, координати центру, масштаб, тривалість періоду, закон зміни інтенсивності кольору, скважність. Також надається такий сервіс як побудова графіків сприйняття параметрів моделі.

Діаграми класів складаються з класів реалізації ПО та класів стандартних бібліотек, що використовуються при розробці ПО. Розглянемо діаграми класів для розроблюваного ПО. Зауважимо, що для даних діаграм використовуються тільки імена класів і не розглядаються атрибути та операції класів.

Коротко зазначимо призначення класів:

CVertex – клас, що описує координати та колір вершини фігури;

TFormGraph – клас, який оброблює події натиску кнопки “Побудувати графік”, яка виконує побудову графіка функції або діаграми на екрані;

TFormMain – клас для промальовування “кадру”;

TFormPrimitiveSelection – клас, який оброблює події виділення елементів в таблиці з моделями;

Function – клас, який описує закон зміни інтенсивності;

Color – клас, який використовується для опису кольору в колірній моделі RGB;

Primitive – клас для опису примітива;

DynamicParams – клас для опису динамічних параметрів;

Transformation – клас для опису геометричних трансформацій;

Field – клас, який описує поля таблиці із вхідними даними про експеримент;

Interval - клас для опису інтервалів.

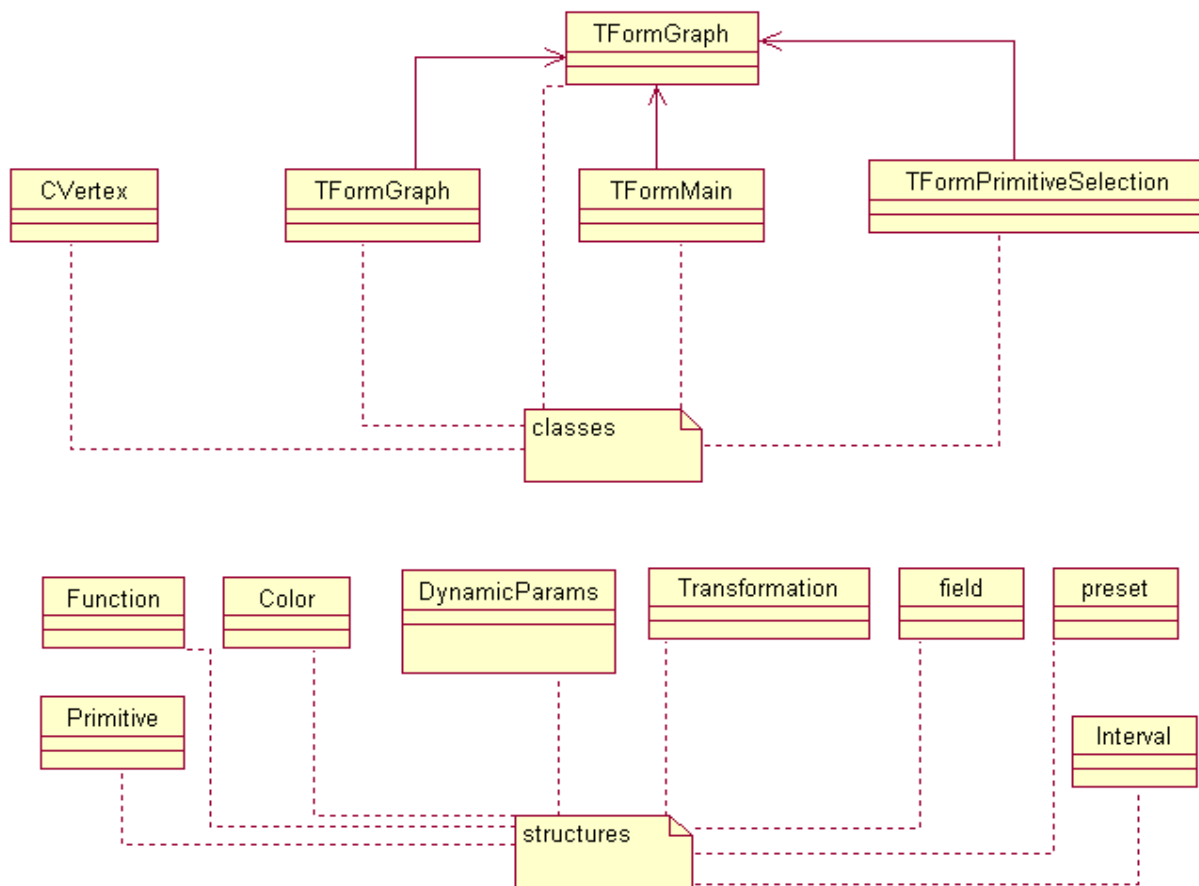


Рис. 3.4. Діаграма наслідування класів

Діаграма станів призначена для опису можливої послідовності станів та переходів, тим самим характеризуючи систему протягом її життєвого циклу [63]. Діаграма станів ПО, що розроблено у даній роботі, наведено на рис. 3.5. Опишемо діаграму станів. Після запуску програми, вона знаходиться у стані очікування наступних дій користувача. Початок роботи користувача означає введення початкових параметрів. На даній діаграмі у початкові параметри включено введення параметрів. Після цього користувач зобов'язаний переглянути динаміку примітиву і ввести експертну оцінку. При цьому користувач може зберегти обрані і введенні параметри у файл і роздрукувати їх.

Представлені діаграми дозволяють розглянути програмне забезпечення з різних боків та скласти цілісну картину програмного забезпечення як у статиці, так і у динаміці. Для більш детального розгляду системи можуть бути розроблені відсутні діаграми з переліку, що наведений на початку даного розділу.

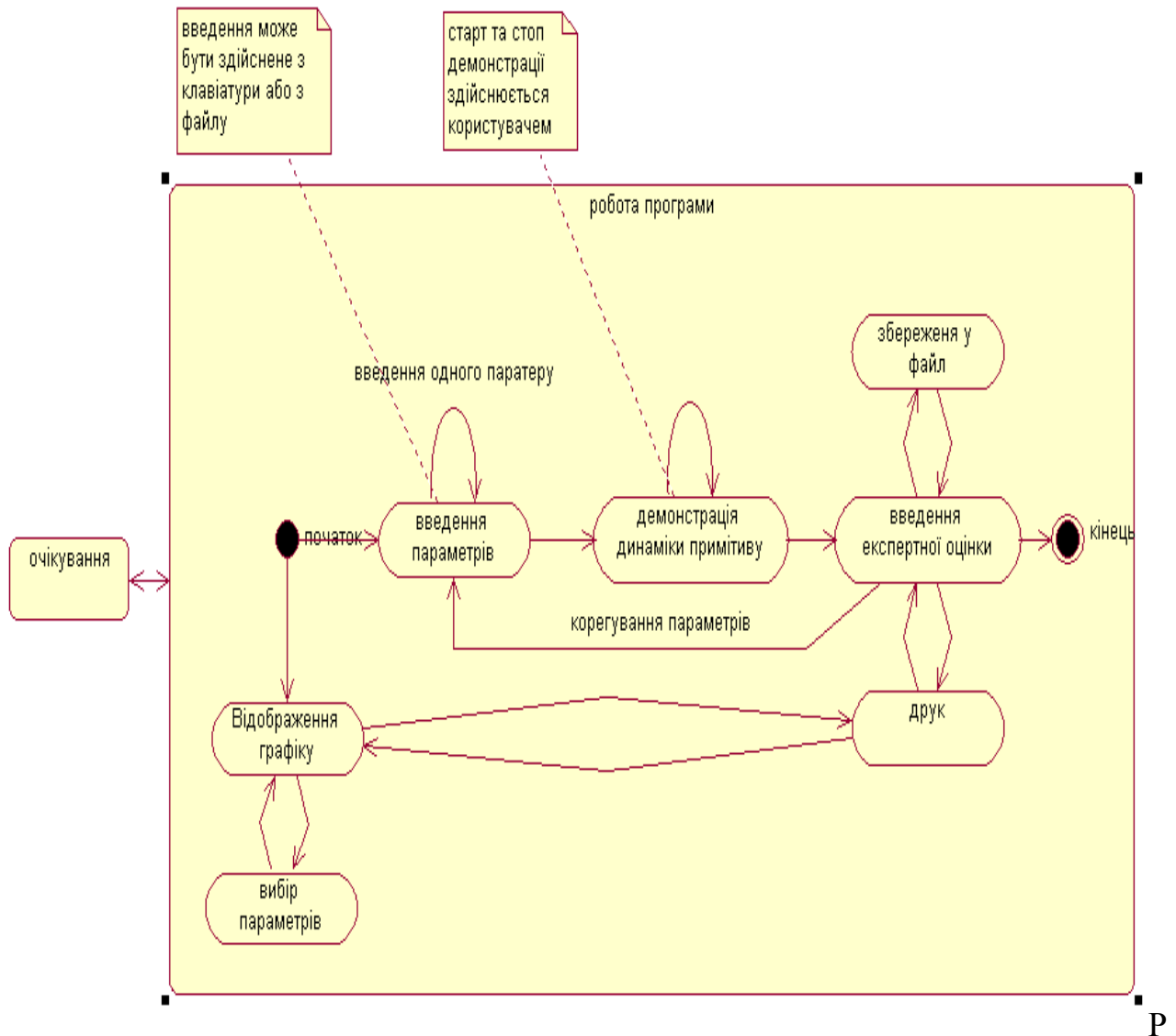


Рис. 3.5. Діаграма станів

3.3. Розробка програмних засобів системи

Для реалізації системи експериментального дослідження моделей динамічних примітивів доцільно використати мову програмування, яка відповідає таким вимогам:

- розповсюджена;
- універсальна;
- має потенціал для розширення;
- має високу швидкодію;

- має зручний інтерфейс;
- має широкий набір можливостей для роботи із графікою.

Із врахуванням всіх перерахованих критеріїв було зроблено висновок про вибір мови програмування – Borland C++ Builder версії 6.0 [9,10,13], оскільки її об'єктно-орієнтований напрямок та широкий вибір функцій та процедур задовольняє основним вимогам для побудови вищевказаної системи.

Крім цього, C++ Builder 6.0 має надзвичайно зручний компілятор для генерації коду, близький за швидкістю до програм на мові assembler. Використане середовище має потужні компоненти і можливості для швидкого створення графічного програмного забезпечення для середовищ Windows, без докладання надмірних зусиль для реалізації інтерфейсу програми [14,15,16].

Також досить важливим є те, що при написанні програми під ОС Windows, за досить короткий термін є можливість отримати програмне забезпечення із „приємним” зовнішнім виглядом, що є досить актуальним на даний момент, оскільки більшість користувачів користуються цією операційною системою та мають основні навички роботи із нею [15].

Програмний продукт вимагає підтримки операційної системи Windows 98 і вище. Крім цього обов'язково має бути встановлений DirectX версії 7.0 і вище, оскільки в даній програмі буде відбуватися побудова динамічного примітиву за допомогою процедур DirectX.

Апаратна частина має задовільняти таким мінімальним вимогам:

- процесор Pentium із тактовою частотою не нижче 166 MHz;
- не менше 16 Mb оперативної пам'яті;
- жорсткий диск розміром не менше 540 Mb;
- SVGA адаптер із 3D прискорювачем [17].

Ці апаратні засоби необхідні для роботи в операційній системі Windows тому вони визначають мінімальні вимоги для роботи програми. Слід також врахувати, що під час роботи програма потребує певного ресурсу оперативної пам'яті.

Для комфортної роботи пропонується така конфігурація персонального комп'ютера:

- процесор Pentium II із тактовою частотою не нижче 300 MHz;
- не менше 64 Мб оперативної пам'яті;
- жорсткий диск розміром не менше 1 Gb;
- SVGA адаптер із 3D прискорювачем оснащеним не менше ніж 16Мб відеопам'яті.

Система експериментального дослідження моделей динамічних примітивів складатиметься із таких модулів:

Dp.cpp – головний файл проекту.

Main.cpp – модуль реалізації всіх функцій та процедур, необхідних для роботи програми.

Graph.cpp – модуль, що містить функції та процедури для побудови порівняльного графіка за даними експертних оцінок відібраних користувачем моделей примітивів.

Dpselect.cpp – модуль, в якому реалізовані процедури для вибору моделей із файлу для побудови порівняльного графіка.

Розглянемо кожен з вищеперерахованих модулів детальніше.

В заголовок модуля Main.cpp включаються файли, наведені на рис. 3.6.

Модуль Main.cpp складається із таких функцій та процедур:

1. Процедури та функції 3D:

class CVertex – клас, що описує координати та колір вершини фігури;

#define SAFERELEASE(x) – безпечне вивільнення ресурсу;

void SetupGraph() – ініціалізація графіки в 3D;

void DoneDirect3D() – деініціалізація графіки в 3D;

void d3dtranslate(double x, double y, double z) – процедура, яка відповідає за зсув по осях X,Y,Z в 3D;

void d3dscale(double a, double b, double c) – процедура для масштабування в 3D;

void d3drotatez(double z) – процедура, яка виконує обертання зображення навколо осі Z.

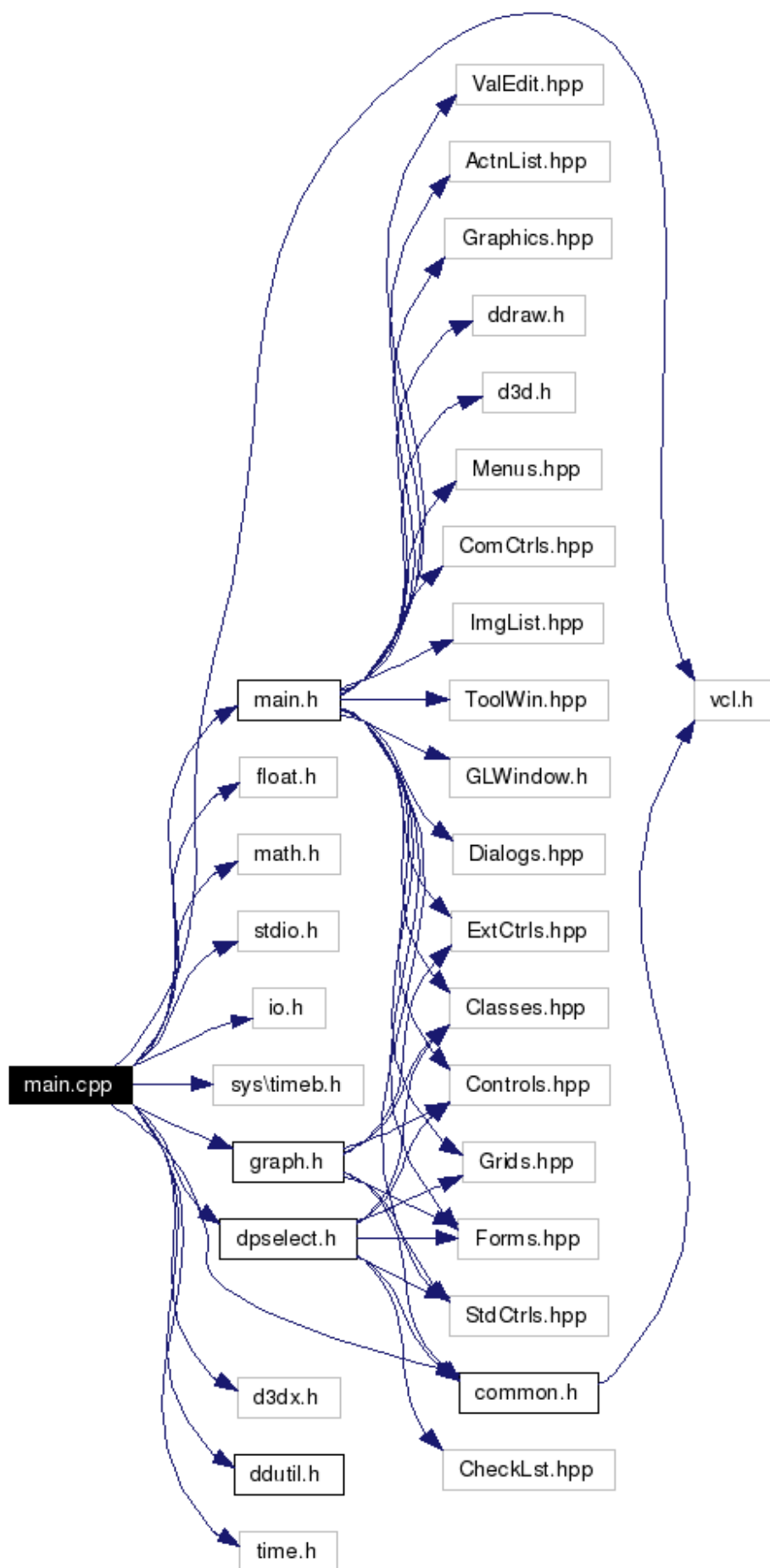


Рис. 3.6. Граф файлів, які включаються в модуль main.cpp

2. Допоміжні функції та процедури:

`double my_get_current_time()` – функція для визначення даного моменту часу;

`AnsiString rnd3(double x)` – функція для округлення числа x до двох знаків;

`AnsiString i2str(int x)` – функція для перетворення цілого числа x в рядок;

`AnsiString d2str(double x)` – функція для перетворення числа із плаваючою точкою (дійсного) в рядок;

`AnsiString fixDot(AnsiString str)` – функція для заміни крапки комою.

3. Функції та процедури для роботи із кольором:

`double mixture(double a, double b, double p)` – процедура, яка використовується для змішування двох кольорів a та b у пропорції p ;

`struct Color` – клас, який використовується для опису кольору в колірній моделі RGB.

4. Функції для створення примітиву:

`struct Transformation` – опис класу, який використовується для геометричних трансформацій;

`double normal_exp(double x, double s)` – функція для визначення експонентного закону за формулою $\frac{e^{x \cdot s} - 1}{e^s - 1}$;

`struct Function` – опис класу “Залежність”;

`double get_value(double x)` – функція для визначення інтенсивності зміни кольору в середині інтервалу за певним законом;

`struct Interval` – опис класу “інтервал”;

`struct DynamicParams` – опис класу “динамічні параметри”;

`void del_interval(int id)` – процедура для знищення інтервалу, id – номер інтервалу;

`void add_interval(int after)` – процедура для додавання нового інтервалу;

$after$ – вказує на номер інтервалу, після якого необхідно додати новий;

`double get_frame(double ctime)` – функція для знаходження „кадру” за поданим часом, який задається змінною `ctime`;

`double get_intensity_for_frame(double cframe)` – функція для знаходження інтенсивності для „кадру”, який задається змінною `cframe`;

`struct Primitive` – опис класу “примітив” ;

`void draw_Ngon(int N)` – функція для відтворення N-кутника, де N – кількість кутів многокутника;

`void draw_rect()` – процедура, яка використовується для зображення прямокутника;

`void draw_tri_r()` – процедура для відтворення зображення прямокутного трикутника;

`void Primitive::draw()` – процедура для промальовування фігури.

5. Функції та процедури головного вікна:

`void __fastcall TFormMain::IdleLoop(TObject*, bool& done)` – процедура для промальовування “кадру” ;

`void __fastcall TFormMain::UpdateFPS(double fps)` – процедура для оновлення вказівника частоти оновлення примітиву;

`int find_name(char * name_list[], char*name)` – пошук стрічки в масиві;

`struct field` – клас, який описує поля таблиці із вхідними даними про експеримент;

`void make_fields(dp::Primitive *p)` – процедура, яка створює таблицю із даними про примітив, який задається динамічною змінною `dp`, класу `Primitive`;

`void SaveToFile(dp::Primitive *p, AnsiString filename)` – процедура, яка використовується для збереження інформації про примітив, яка записана у таблиці. `dp` – примітив, `AnsiString` – стрічкова величина, яка вказує на ім'я файлу, в який заноситься інформація про примітив;

`void OpenFromFile(dp::Primitive *p, AnsiString filename)` – процедура, яка використовується для відкриття файлу в якому записана інформації про примітив. `dp` – примітив, `AnsiString` – стрічкова величина, яка вказує на ім'я файлу, в якому записана інформація про примітив;

void ReadPFromBuffer(dp::Primitive *p, char*data) – процедура, яка зчитує із буфера;

void TFormMain::FillIntervalList() – процедура для промальовування списку інтервалів;

void TFormMain::constructor() – вмикає режим конструктора;

void TFormMain::no_constructor() – вимикає режим конструктора;

void DrawIntensityGraph() – процедура, яка використовується для промальовування графіка інтенсивності;

void TFormMain::FillExpert() – процедура, яка використовується для заповнення таблиці експертних оцінок;

void TFormMain::FillControlsWithPrimitiveValues() – процедура для заповнення керуючих елементів значеннями із примітиву;

void SetupDefaultDyn() – встановлення стандартних динамічних параметрів примітиву;

void PreparePrimitive() – створення нового примітиву;

void CountFrame() – підрахування частоти оновлення примітиву;

void RenderScene() – процедура, яка використовується для промальовування нового “кадру”;

void TFormMain::StopDrawing() – процедура, яка використовується для зупинення процесу промальовування фігури;

void TFormMain::StartDrawing() – процедура, яка використовується для старшування процесу промальовування фігури;

void __fastcall TFormMain::EditScaleChange(TObject *Sender) – оброблювач події зміни керуючого елементу “EditScale”;

void __fastcall TFormMain::TrackBarScaleChange(TObject *Sender) – оброблювач події зміни керуючого елементу “TrackBarScale” ;

void __fastcall TFormMain::TrackBarRotateChange(TObject *Sender) – оброблювач події зміни керуючого елементу “TrackBarRotate” ;

void __fastcall TFormMain::EditRotateCWChange(TObject *Sender) – оброблювач події зміни керуючого елементу “EditRotateCW”;

`void __fastcall TFormMain::EditRotateCCWChange(TObject *Sender)` – оброблювач події зміни керуючого елементу “EditRotateCCW”;

`void __fastcall TFormMain::TreeViewFormClick(TObject *Sender)` – оброблювач події зміни натиску керуючого елементу “TreeViewForm”;

`void TFormMain::SetupTreeViewForm()` – процедура, яка використовується для створення дерева із основними формами примітивів.

`void TFormMain::ShowColor(TImage*img, TColor cl)` – процедура, яка використовується для промальовування кольору;

`void TFormMain::ShowMainColor()` – процедура, яка використовується для показу головного кольору;

`void TFormMain::ShowSecColor()` – процедура, яка використовується для показу додаткового кольору;

`void TFormMain::ShowBgColor()` – процедура, яка використовується для показу фонового кольору;

`void __fastcall TFormMain::ImageMainColorClick(TObject *Sender)` – процедура для вибору головного кольору, при натиску на відповідній області, відкривається палітра із кольорами;

`void __fastcall TFormMain::ImageBgColorClick(TObject *Sender)` – процедура для вибору кольору фону, при натиску на відповідній області, відкривається палітра із кольорами;

`void __fastcall TFormMain::ImageSecColorClick(TObject *Sender)` – процедура для вибору додаткового кольору, при натиску на відповідній області, відкривається палітра із кольорами;

`void __fastcall TFormMain::EditPeriodChange(TObject *Sender)` – процедура, яка використовується для вказування необхідного періоду в числовому полі;

`void __fastcall TFormMain::EditOnOffChange(TObject *Sender)` – оброблювач події зміни керуючого елементу “EditOnOff”;

`void __fastcall TFormMain::TrackBarPeriodChange(TObject *Sender)` – процедура, яка використовується для вказування необхідного періоду за допомогою елементу Track Bar;

`void __fastcall TFormMain::TrackBarOnOffChange(TObject *Sender)` – оброблювач події зміни керуючого елементу “TrackBarOnOff”;

`void __fastcall TFormMain::EditRightExit(TObject *Sender)` – процедура, яка використовується для введення числа в числове поле, на яке буде виконуватися процес зміщення примітиву вправо;

`void __fastcall TFormMain::EditLeftExit(TObject *Sender)` – процедура, яка використовується для введення числа в числове поле, на яке буде виконуватися процес зміщення примітиву вліво;

`void __fastcall TFormMain::EditUpExit(TObject *Sender)` – процедура, яка використовується для введення числа в числове поле, на яке буде виконуватися процес зміщення примітиву вгору;

`void __fastcall TFormMain::EditDownExit(TObject *Sender)` – процедура, яка використовується для введення числа в числове поле, на яке буде виконуватися процес зміщення примітиву вниз;

`void __fastcall TFormMain::EditXChange(TObject *Sender)` – процедура, яка змінює значення координати центру примітиву по значенню X;

`void __fastcall TFormMain::EditYChange(TObject *Sender)` – процедура, яка змінює значення координати центру примітиву по значенню Y;

`void __fastcall TFormMain::ProgressPaintPaint(TObject *Sender)` – процедура, яка використовується для промальовування індикатора зміни інтенсивності примітиву;

`void __fastcall TFormMain::ToolButtonStopClick(TObject *Sender)` – процедура, яка відповідає за зупинення процесу промальовування примітиву;

`void __fastcall TFormMain::ToolButtonStartClick(TObject *Sender)` – процедура, яка відповідає за запуск процесу промальовування примітиву;

`void TFormMain::SetDefaultLaw(int id)` – процедура для застосування стандартно назначеного закону;

`void __fastcall TFormMain::ToolButton1Click(TObject *Sender) –`
 оброблювач події натиску керуючого елементу “ ToolButton1” ;

`void __fastcall TFormMain::ToolButton2Click(TObject *Sender) –`
 оброблювач події натиску керуючого елементу “ ToolButton2” ;

`void __fastcall TFormMain::ToolButton3Click(TObject *Sender) –`
 оброблювач події натиску керуючого елементу “ ToolButton3” ;

`void __fastcall TFormMain::ToolButton4Click(TObject *Sender) –`
 оброблювач події натиску керуючого елементу “ ToolButton4” ;

`void __fastcall TFormMain::ToolButton5Click(TObject *Sender) –`
 оброблювач події натиску керуючого елементу “ ToolButton5” ;

`void __fastcall TFormMain::ToolButton6Click(TObject *Sender) –`
 оброблювач події натиску керуючого елементу “ ToolButton6” ;

`void __fastcall TFormMain::ToolButton7Click(TObject *Sender) –`
 оброблювач події натиску керуючого елементу “ ToolButton7” ;

`void __fastcall TFormMain::ToolButton8Click(TObject *Sender) –`
 оброблювач події натиску керуючого елементу “ ToolButton8” ;

`void __fastcall TFormMain::ToolButton9Click(TObject *Sender) –`
 оброблювач події натиску керуючого елементу “ ToolButton9” ;

`void __fastcall TFormMain::TreeViewFormExpanded(TObject *Sender, TTreeNode *Node) –` процедура, яка виводить у формі дерево із різними значеннями форм примітиву;

`void __fastcall TFormMain::ExitClick(TObject *Sender) –` оброблювач події натиску керуючого елементу “Exit” ;

`void __fastcall TFormMain::EditSideAChange(TObject *Sender) –`
 оброблювач події зміни керуючого елементу “EditSideA” ;

`void __fastcall TFormMain::EditSideBChange(TObject *Sender) –`
 оброблювач події зміни керуючого елементу “EditSideB” ;

`void __fastcall TFormMain::EditSideCChange(TObject *Sender) –`
 оброблювач події зміни керуючого елементу “EditSideC” ;

`void __fastcall TFormMain::ListViewIntervalsClick(TObject *Sender)` – оброблювач події натиску керуючого елементу “ListViewIntervals” ;

`void __fastcall TFormMain::DrawGridIntervalsDrawCell(TObject *Sender, int ACol, int ARow, TRect &Rect, TGridDrawState State)` – процедура, яка використовується для промальовування елементів списку інтервалів;

`void __fastcall TFormMain::EditDurationExit(TObject *Sender)` – оброблювач події втрати фокусу керуючого елементу “EditDuration” ;

`void __fastcall TFormMain::DrawGridIntervalsSelectCell(TObject *Sender, int ACol, int ARow, bool &CanSelect)` – оброблювач події вибору комірки елементу списку інтервалів;

`void TFormMain::UpdateDurations()` – процедура для підрахунку довжин інтервалів;

`void __fastcall TFormMain::EditDurationKeyDown(TObject *Sender, WORD &Key, TShiftState Shift)` – оброблювач події натиску кнопки при зміні довжини інтервалу;

`void __fastcall TFormMain::ButtonAddIntervalClick(TObject *Sender)` – процедура, яка використовується в „конструкторі” для додавання нового інтервалу;

`void __fastcall TFormMain::ButtonRemoveIntervalClick(TObject *Sender)` – процедура, яка використовується в „конструкторі” для знищення вказаного інтервалу;

`void int_set_style(int id)` – процедура, яка використовується для зміни закону вибраного інтервалу;

`void __fastcall TFormMain::lin11Click(TObject *Sender)` – оброблювач події натиску керуючого елементу “lin11Click” – встановлення на проміжку закону “Лінійний 1” ;

`void __fastcall TFormMain::lin21Click(TObject *Sender)` – оброблювач події натиску керуючого елементу “lin21Click” – встановлення на проміжку закону “Лінійний 2” ;

`void __fastcall TFormMain::s11Click(TObject *Sender)` – оброблювач події натиску керуючого елемента “s11Click” – встановлення на проміжку закону “Синусоїдний 1” ;

`void __fastcall TFormMain::s21Click(TObject *Sender)` – оброблювач події натиску керуючого елемента “s21Click” – встановлення на проміжку закону “Синусоїдний 2” ;

`void __fastcall TFormMain::d11Click(TObject *Sender)` – оброблювач події натиску керуючого елемента “d11Click” – встановлення на проміжку закону “Дискретний 1” ;

`void __fastcall TFormMain::d21Click(TObject *Sender)` – оброблювач події натиску керуючого елемента “d21Click” – встановлення на проміжку закону “Дискретний 2” ;

`void __fastcall TFormMain::e11Click(TObject *Sender)` – оброблювач події натиску керуючого елемента “e11Click” – встановлення на проміжку закону “Експонентний 1” ;

`void __fastcall TFormMain::e21Click(TObject *Sender)` – оброблювач події натиску керуючого елемента “e21Click” – встановлення на проміжку закону “Експонентний 2” ;

`void __fastcall TFormMain::e31Click(TObject *Sender)` – оброблювач події натиску керуючого елемента “e31Click” – встановлення на проміжку закону “Експонентний 3” ;

`void __fastcall TFormMain::e41Click(TObject *Sender)` – оброблювач події натиску керуючого елемента “e41Click” – встановлення на проміжку закону “Експонентний 4” ;

`void __fastcall TFormMain::DrawGridExpertDrawCell(TObject *Sender, int ACol, int ARow, TRect &Rect, TGridDrawState State)` – оброблювач події для перемалювання елементів таблиці експертної оцінки;

`void __fastcall TFormMain::TreeViewFormChange(TObject *Sender, TTreeNode *Node)` – оброблювач події зміни керуючого елемента `TreeViewFormChange` – зміни елементів “дерева” форм;

`void __fastcall TFormMain::ToolButtonResultsClick(TObject *Sender)` –

оброблювач події натиску кнопки “Зберегти”, яка використовується для збереження даних про експертну оцінку.

В заголовок модуля Graph.cpp включаються файли приведені на рис. 3.7.

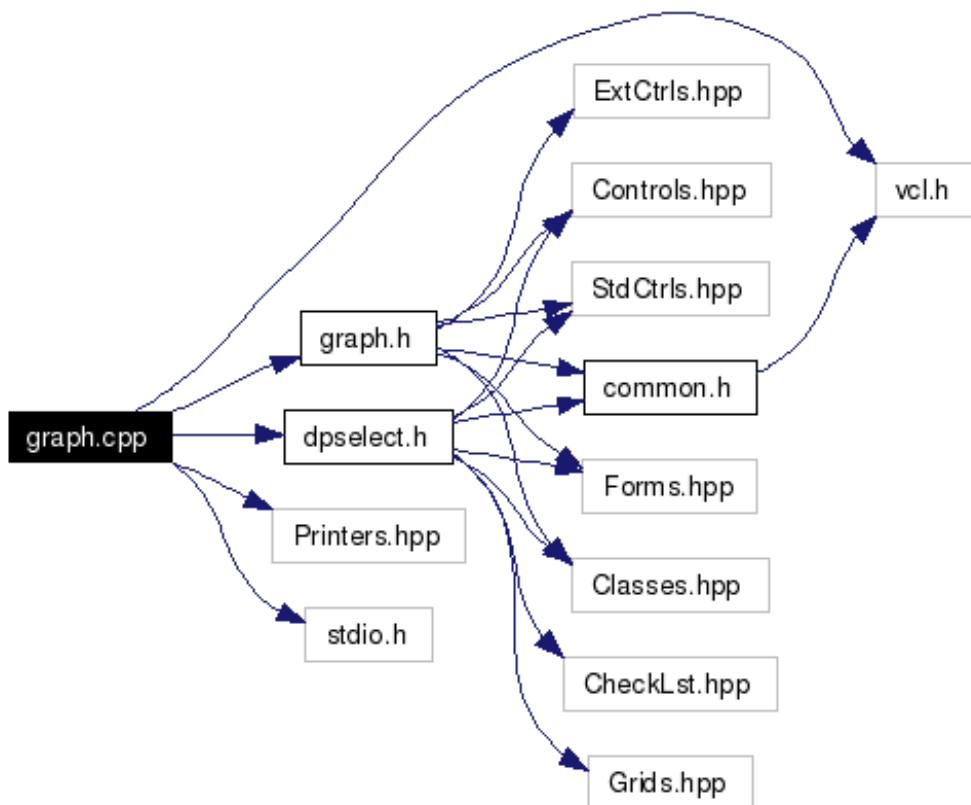


Рис. 3.7. Граф файлів, які включаються в модуль Graph.cpp

Модуль Graph.cpp складається із таких процедур:

`void __fastcall TFormGraph::ButtonDrawGraphClick(TObject *Sender)` – оброблювач події натиску кнопки “Побудувати графік”, яка виконує побудову графіка функції або діаграми на екрані;

`void __fastcall my_print_image(TImage *image)` – оброблювач події натиску кнопки “Друкувати”, яка відсилає зображення графіка на друк.

В заголовок модуля DpSelect.cpp включаються файли приведені на рис. 3.8.

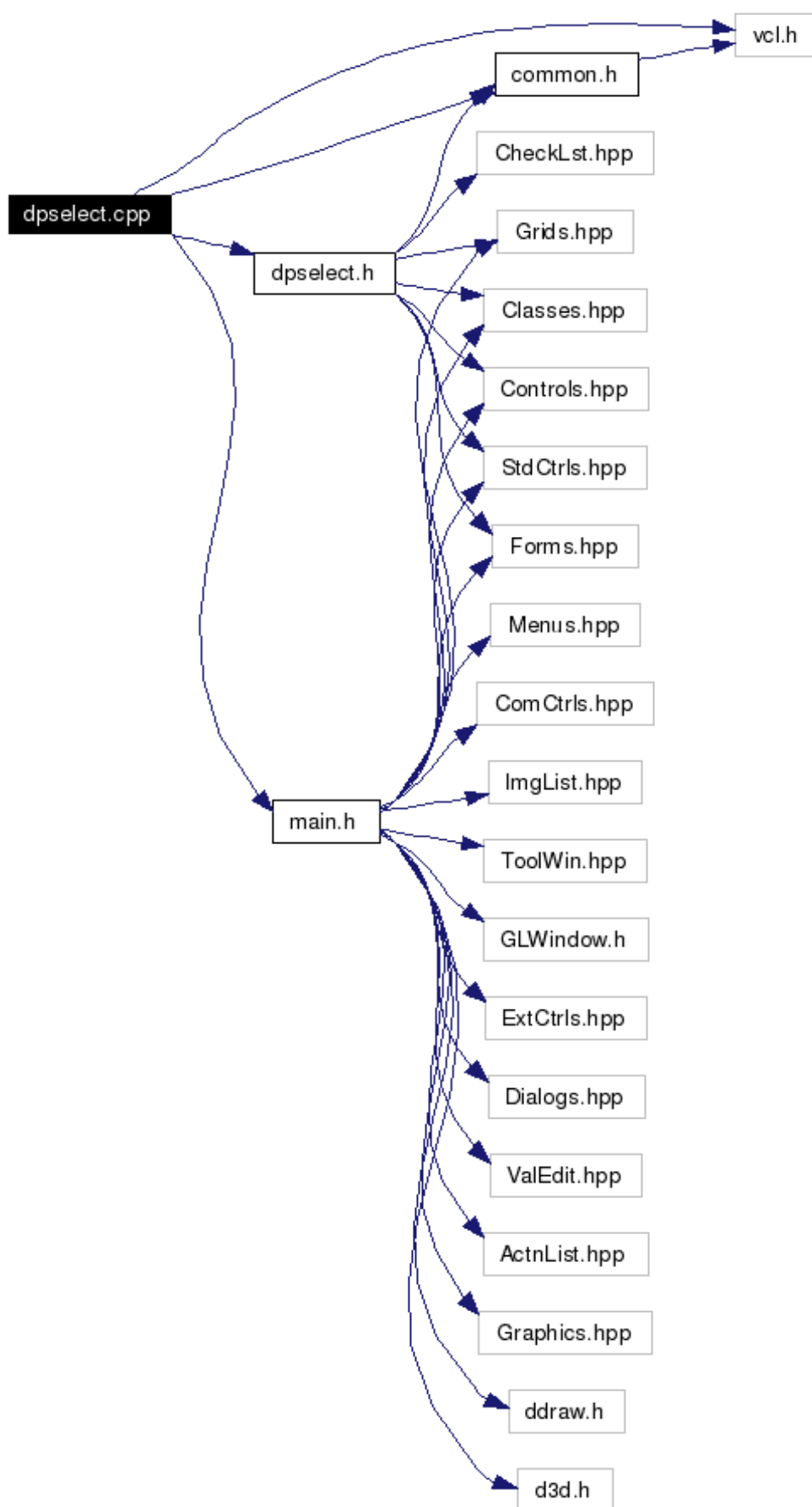


Рис. 3.8. Граф файлів, які включаються в модуль `DpSelect.cpp`

Модуль DpSelect.cpp складається із таких процедур:

`void __fastcall TFormPrimitiveSelection::PrimGridSelectCell(TObject *Sender, int ACol, int ARow, bool &CanSelect)` – оброблювач події виділення елементів в таблиці з моделями;

`void __fastcall TFormPrimitiveSelection::PrimGridClick(TObject *Sender)` – оброблювач події натиску кнопки мишки в таблиці для виділення елементів;

`void TFormPrimitiveSelection::ShowPrimList(struct dp::Primitive * plist[], int pnum)` – процедура, яка використовується для створення таблиці, в яку записуються моделі із файлу.

3.4. Розробка інтерфейсу системи

Основним призначенням системи експериментального дослідження моделей динамічних примітивів є моделювання динамічного примітиву та його дослідження.

Процес моделювання включає в себе:

- вибір форми примітиву;
- визначення розмірів фігури;
- поворот фігури;
- зсув примітиву у чотирьох напрямках відносно центру;
- встановлення центру координат;
- масштабування примітиву;
- вибір кольору пікселя(основного, вторинного);
- вибір кольору фону;
- вибір закону, за яким буде змінюватись колір примітиву;
- встановлення тривалості періоду;
- встановлення скважності.

Процес дослідження полягає у візуальній оцінці сприйняття зміни інтенсивності кольору динамічного примітиву та встановлення відповідного коефіцієнту сприйняття створеної моделі. Коефіцієнт задається у відсотковому

форматі в межах від 1 до 100% (1% - майже не сприймається, 100% - сприймається дуже добре).

Інтерфейс головного вікна програми показаний на рис. 3.1.

Вікно системи складається з таких компонентів:

- рядок меню;
- керуючі кнопки;
- блоки “Демонстрація примітиву”, “Експертна оцінка”, “Статичні параметри”, “Форма”, “Колір”, “Параметри”, “Масштаб”, “Поворот”, “Зсув”, “Графік зміни інтенсивності”, “Закон”, “Часові параметри”, “Властивості проміжків”,

Розглянемо вищеописані складові детальніше.

Для повноцінного функціонування даної системи рядок меню повинен складатися із таких пунктів:

- “Файл” – включає підпункти загального призначення, які використовуються для створення нового експерименту, збереження та відкриття даних про модель динамічного примітиву;
- “Параметри статички” – використовується для встановлення статичних параметрів примітиву;
- “Параметри динаміки” – використовується для встановлення динамічних параметрів примітиву;
- “Моделі” – використовується для завантаження моделей з необхідними для користувача параметрами;
- “Експертна оцінка” – використовується для проведення експертної оцінки примітиву;
- “Результати” – використовується для проведення статистичного аналізу по створених та оцінених моделях;
- “Допомога” – використовується для надання довідкової інформації про програму і її використання.

Пункт меню „Файл” повинен містити такі підпункти:

- “Новий” – необхідний для того, щоб розпочати створення нового експерименту;
- “Зчитати” – потрібен для зчитування динамічних і статичних параметрів динамічного примітиву і його експертна оцінка з файлу із розширенням *.CSV або *.TXT. При натиску відкривається стандартне діалогове вікно для відкриття файлу (рис. 3.9) в якому необхідно вказати ім’я файлу та натиснути кнопку „Відкрити”;
- “Зберегти” – використовується для збереження даних про експеримент та його експертну оцінку у файл із розширенням *.CSV або *.TXT:
- “Зберегти як” – необхідний для збереження даних про експеримент та його експертну оцінку у файл під іншим іменем;
- “Друк” – друк статичних, динамічних параметрів і експертної оцінки заданих в моделі;
- “Вийти” – вихід із програми.

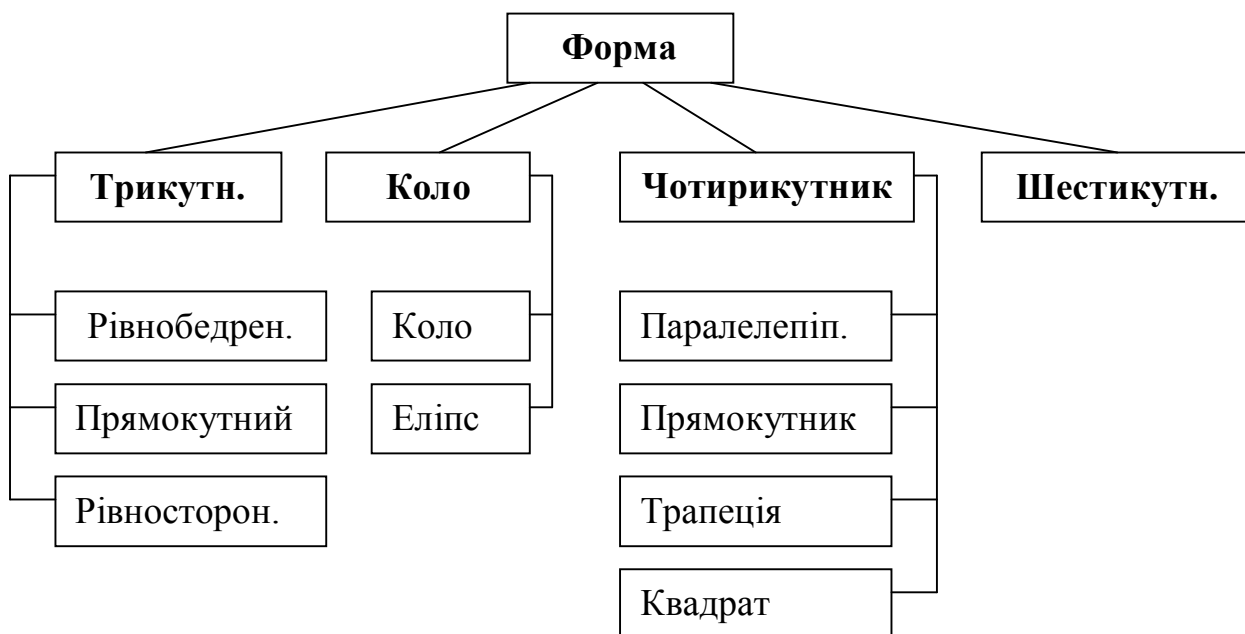


Рис. 3.10. Структура пункту підменю „Форма”

- “Зсув” – зсув примітиву від центру поля відображення примітиву на 10, 50, 100 точок в напрямках: вліво, вправо, вгору, вниз.

Пункт меню „Параметри динаміки” містить функції для встановлення основних динамічних параметрів. Він має складну будову та складається із таких підпунктів:

- “Закон” – вибір та встановлення закону, за яким буде змінюватися інтенсивність кольору примітиву. Складається із таких підпунктів: “Лінійний 1”, “Лінійний 2”, “Лінійний 3”, “Динамічний”, “Синусоїдальний”, “Експонентний 1”, “Експонентний 2”, “Експонентний 3”, “Експонентний 4”;
- “Тривалість періоду” – використовується для встановлення числового значення тривалості періоду зміни інтенсивності кольору. Основні значення: 0,1с, 1с, 5с, 10с. Дані значення обрані за основні, оскільки вони є найбільш типовими при завданні даного параметру. При необхідності, користувач може встановити довільну тривалість періоду в блоці “Часові параметри”.
- “Скважність” – встановлення значення скважності. Для користувача будуть запропоновані значення від 10% до 100% із кроком 10. Більш точне значення скважності користувач зможе встановити за допомогою блоку “Часові параметри”.

Пункт меню „Моделі” – містить перелік стандартних моделей, які створює експерт системи.

Пункт меню „Експертна оцінка” містить підпункт „Коефіцієнт сприйняття”, який використовується для встановлення відсоткового значення сприйняття моделі динамічного примітиву. Можливі значення: 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90% 100%.


Пункт меню „Результати” використовується для виведення статистичних даних – графіків, діаграм по результатам оцінки динамічних примітивів. Даний пункт меню складається із двох підпунктів:


- “Побудувати діаграму” – побудова лінійного графіка або гістограми сприйнятності по певному статичному або динамічному параметру або по всій моделі в цілому;
- “Найкращі моделі” – відкривається файл із переліком найкращих моделей динамічного примітиву, які вказав користувач.


Пункт меню „Допомога” використовується для надання необхідної інформації користувачеві. Складається він із двох підпунктів:


- “Про програму” – виводить інформацію про автора та версію програми;
- “Підказка” – використовується для надання довідкової інформації по користуванню програмою.

До складу керуючих кнопок входять такі:


 – „Завантажити файл” – використовується для завантаження файлу із даними про примітив. Дублює пункт меню „Файл” → „Зчитати”;


 – „Зберегти у файл” – використовується для збереження інформації про примітив у файл. Дублює пункт меню „Файл” → „Зберегти” ;

 – „Роздрукувати” – використовується для друку головного вікна програми;

 – „Пуск” – запуск процесу динаміки примітиву (зміна інтенсивності кольору та його демонстрація) ;

 – „Зупинити” – зупиняє процес динаміки примітиву;

 – „Графік” – використовується для побудови графіка на основі створених моделей. Дублює пункт меню „Результати” → „Побудувати діаграму” ;

 – „Допомога” – відкриває довідкову систему для надання користувачеві інформації по користуванню програмою. Дублює пункт меню „Результати” → „Побудувати діаграму”.

Розглянемо складові (блоки) головного вікна. У блоці „Демонстрація примітиву” знаходиться область, яка використовується для виведення зображення примітиву а також його динаміки.

Блок “Експертна оцінка” містить всі обрані статичні і динамічні параметри примітиву, які відображуються у вигляді таблиці. Під таблицею із статистикою міститься текстове поле для введення назви моделі при збереженні її у файл та скроллер із лічильником для встановлення відсоткового значення коефіцієнту сприйняття. В правій нижній частині блоку знаходиться кнопка „Зберегти”, яка призначена для зберігання поточної моделі у файл експерименту.

Блок “Статичні параметри” включає в себе блоки, які стосуються статистичних параметрів примітиву.

Блок “Форма” містить „Дерево” для вибору геометричної форми примітиву. Для розкриття гілок „Дерева” необхідно натиснути по значку „+”. Після відповідної форми стандартний п’ятикутник замінюється на обрану фігуру.

Блок “Масштаб” містить скролер та лічильник для встановлення масштабу примітиву. Масштабування можливе в межах від 1 до 256.

Блок “Поворот” складається із скроллера та двох полів для відображення кута повороту за годинниковою стрілкою або проти. Поворот фігури відбувається в межах від -90° до $+90^\circ$, який зразу ж відображається на екрані.

Блок “Зсув” складається із чотирьох числових полів, в які записується число точок для зсуву фігури у одному із чотирьох напрямків (вверх, вниз, вліво, вправо). Крім цього, в даному блоці є ще два числових поля X та Y для встановлення центру координат.

Блок “Графік зміни інтенсивності” містить дві області – одна для відображення графіка зміни інтенсивності за обраним законом, а інша для відображення тривалості періоду.

Блок “Динамічні параметри” відображає динамічні параметри примітиву та містить у собі три блоки: “Закон”, “Часові параметри”, “Властивості проміжків”.

Блок “Закон” містить дев’ять кнопок для вибору закону, який описує зміну інтенсивності кольору та кнопку „Конструктор” для моделювання користувачем довільної функції зміни інтенсивності кольору.

Блок „Часові параметри” створений для встановлення тривалості періоду та скважності. Для встановлення кожного із значень використовується скроллер із лічильником. Тривалість періоду може змінюватися в межах від 100 до 10000 мс, а скважність від 1 до 100%.

3.5. Керівництво користувача по роботі з системою

Перед запуском програми необхідно впевнитись у тому, що апаратна і програмна частини комп'ютера задовільняють мінімальним вимогам, описаним у другому розділі. Якщо це так, то можна запускати програму на виконання. Для цього необхідно запустити файл dr.exe. На екрані з'явиться головне вікно програми (див. рис. 3.1)

Розглянемо детально процес створення нового динамічного примітиву та проведення його оцінювання. Цей процес можна подати у вигляді покрокової деталізованої схеми (рис. 3.11).

Встановлення статичних параметрів примітиву почнемо з параметру „Форма примітиву”, оскільки це є один з обов'язкових параметрів. По

замовчуванню форма примітиву – правильний п'ятикутник. Для того, щоб змінити форму, необхідно спочатку перейти на розділ „Форма” (рис. 3.12).

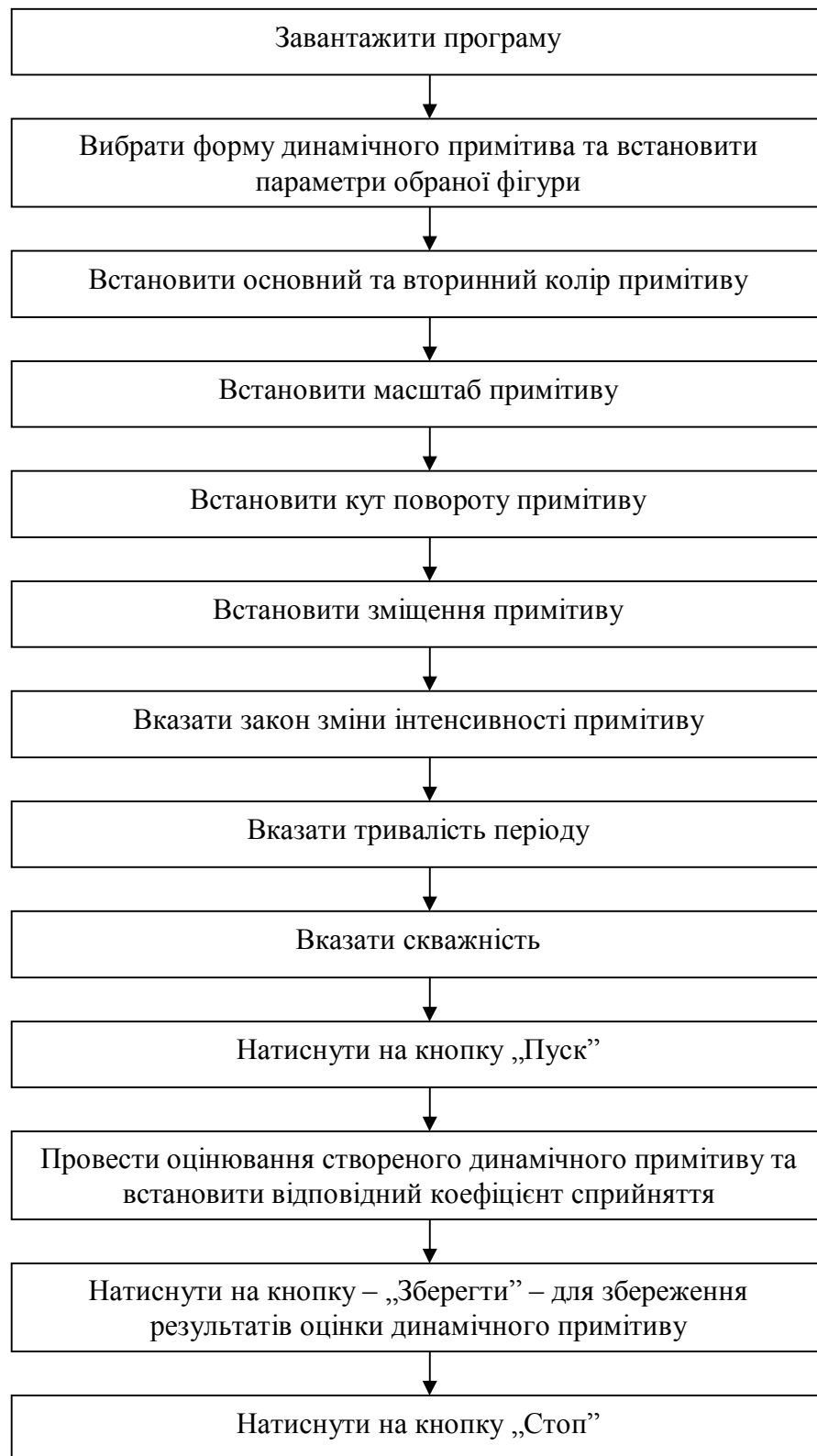


Рис. 3.11. Блок-схема алгоритму створення динамічного примітиву та проведення його оцінювання

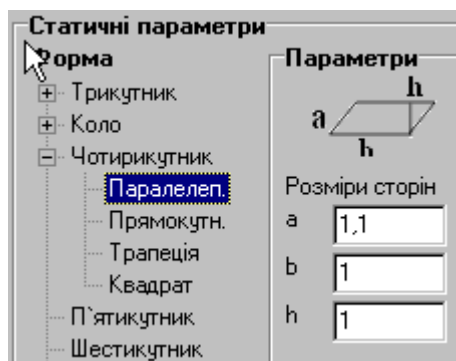


Рис. 3.12. Розділи головного вікна „Форма” і „Параметри”

Потім із дерева вибрати основну фігуру – трикутник, коло, чотирикутник або будь-яку іншу. При виборі однієї із основних фігур дерево розпадається на види цієї фігури. Натиснувши на певний вид фігури, можна форму динамічного примітиву змінити на обрану.

Після вибору розмірів піксела у розділі „Параметри” (рис. 3.12) з’являються числові поля, в які необхідно ввести основні параметри обраної фігури. Наприклад, для квадрату можна задати розмір сторони, а для трапеції – дві основи і висоту. Користувач може залишити стандартні розміри фігури, не змінюючи їх.

До моделі піксела можна застосувати афінні перетворення. Розглянемо афінне перетворення типу „Масштаб” (рис. 3.13). Початкове значення масштабу рівне 256 відносних одиниць, тобто фігура примітиву має максимально допустимі розміри, при яких вона повністю відображається на екрані. Змінюючи значення в лічильнику за допомогою скроллера, масштаб фігури можна змінювати від 1 до 256.

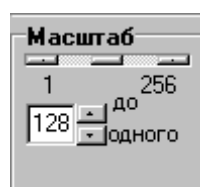


Рис. 3.13. Розділ головного вікна програми - „Масштаб”

Розглянемо можливість повертання фігури на будь-який кут (рис. 3.14). Початкове значення кута повороту фігури рівне 0° . За допомогою скролера користувач може змінювати кут повороту в межах від -90° (поворот проти годинникової стрілки) до $+90^\circ$ (поворот за годинниковою стрілкою). Також є можливість клавіатурного введення кута повороту у відповідне числове поле. При зміні кута повороту ми можемо одразу переглянути повернену фігуру у полі відображення примітиву.

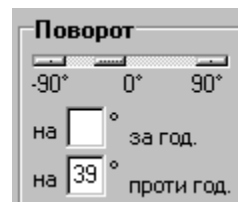


Рис. 3.14. Розділ головного вікна програми - „Поворот”

У користувача є можливість змінювати основний колір пікселя – колір пікселя при нульовому значенні інтенсивності та вторинний колір – колір пікселя при максимальній інтенсивності, а також колір фону. Для вибору кольору необхідно перейти у розділ „Колір” і один раз натиснути на прямокутнику із зображенням відповідного кольору. Після цього у допоміжному вікні з палітрою кольорів (рис. 3.15) потрібно вибрати необхідний колір і натиснути кнопку „ОК”. При необхідності є можливість вибору відтінків основних кольорів. Для цього потрібно натиснути на кнопці „Визначити колір” та вибрати потрібний відтінок.

Також існує можливість зсуву фігури відносно центру координат у чотирьох напрямках: вгору, вниз, вліво, вправо (рис. 3.16). У відповідне поле вводиться кількість точок, на яку потрібно зсунути зображення. Не рекомендовано виконувати зсув більше ніж на 250 точок, оскільки даного примітиву не буде видно на екрані. Для того, щоб фігура повернулась у початкову точку, необхідно виконати зсув у протилежному напрямку на таку ж саму кількість точок.

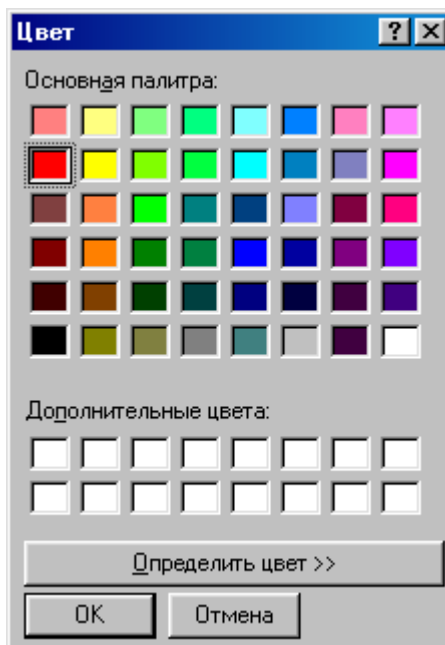


Рис. 3.15. Вікно для вибору кольору піксела або фону

Зсув фігури не є обов'язковим, оскільки за замовченням фігура розміщена в центрі поля „Демонстрація примітиву”. Координати центру розміщення фігури на екрані можна також змінити. Для цього користувачеві необхідно перейти на розділ “Координати центра” і ввести потрібні координати по осям X та Y. Стандартне значення координат центру: $X = 128$ та $Y = 128$, тобто фігура розміщена по центру, оскільки розміри поля відображення примітиву 256×256 точок.

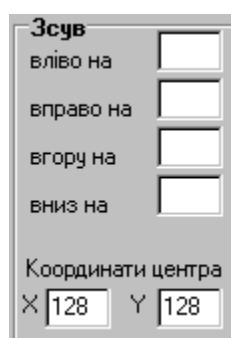


Рис. 3.16. Розділ головного вікна програми - „Зсув” і „Координати центру”

Поряд із завданням статичних параметрів користувач має можливість визначати і динамічні параметри. Розглянемо режими задання основного динамічного параметру - закону інтенсивності примітиву - закону, яким задається зміна інтенсивності кольору примітиву. Для його задання в розділі

“Закон” існує дев’ять кнопок, за їх допомогою можна обрати певний закон. Графік функції, якою описується закон, відображається на кнопці, що є досить зручним для користувача.

Для того, щоб обрати один з заздалегідь заданих законів, потрібно натиснути на одну з кнопок (рис. 3.17) - “Лінійний 1”, “Лінійний 2”, “Лінійний 3”, “Динамічний”, “Синусоїдний”, “Експонентний 1”, “Експонентний 2”, “Експонентний 3”, “Експонентний 4”.



Рис. 3.17. Розділ головного вікна програми - „Закон”

При необхідності коригування закону, або відсутності необхідного, користувач може скористатися “Конструктором”, за допомогою якого можна створити нову функцію, яка буде включати декілька простих стандартних законів.

При дослідженні динамічних характеристик піксела досить важливим є задання часових параметрів, до яких віднесемо тривалість періоду і скважність. Тривалість періоду встановлюється за допомогою скроллера, лічильника або клавіатурного введення (рис. 3.18). Тривалість періоду задається у секундах, але існує можливість вводити і мілісекунди. Параметр є обов’язковим та не може бути меншим 0,1с. Встановлення значення скважності відбувається за допомогою скроллера, лічильника або клавіатурного введення. Значення задається у відсотковому форматі від 1% до 100%. Стандартне значення – 69%. Параметр є обов’язковим.

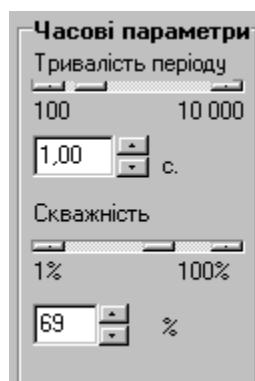


Рис. 3.18. Розділ головного вікна програми - „Часові параметри ”

Всі описані вище статичні і динамічні параметри можна задати, скориставшись відповідними пунктами меню. Це є дуже зручним моментом інтерфейсу.

Розглянемо процес створення нового закону в режимі “Конструктор”. Для ініціалізації данного режиму користувачу необхідно натиснути на відповідну кнопку – “Конструктор” у розділі “Закон”. Після цього у нього з’явиться можливість редагування проміжків у вікні “Властивості проміжків” (рис. 3.19). Існує можливість додавання нового проміжку за допомогою кнопки “Додати” і видалення будь-якого проміжок за допомогою кнопки “Видалити”. Також є можливість редагування тривалості проміжку, який вводиться в колонку “Час” для обраного проміжку. Вимірюється тривалість проміжку в секундах.

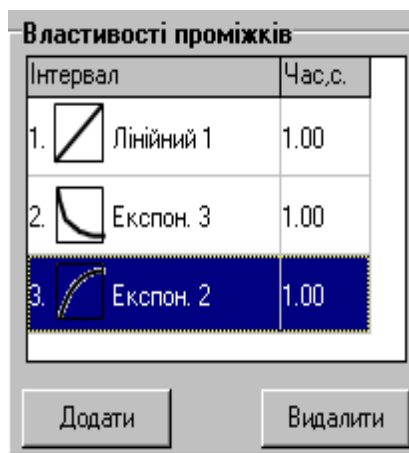


Рис. 3.19. Розділ головного вікна програми - „Властивості проміжків”

Для того щоб змінити закон на будь-якому проміжку, потрібно відкрити контекстне меню (рис. 3.20) на даному проміжку та обрати потрібний закон.

Після закінчення введення всіх статичних і динамічних параметрів для запуску експерименту потрібно натиснути кнопку „Запуск” на панелі інструментів. Для зупинки динамічного процесу і перегляду всіх встановлених параметрів необхідно натиснути кнопку „Зупинити” на панелі інструментів.

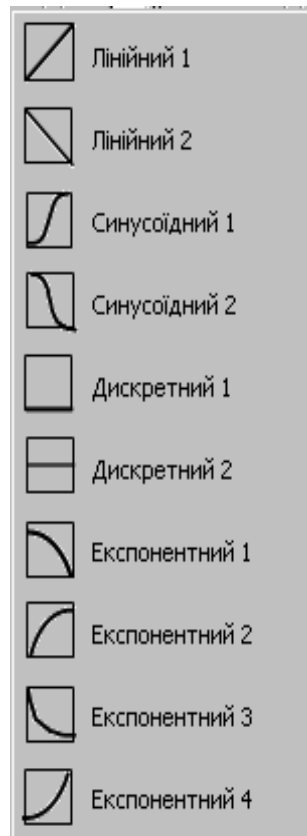


Рис. 3.20. Контекстне меню режиму “Конструктора”

Всі встановлені статичні і динамічні параметри моделі а також її експертна оцінка можуть бути збережені у файл. Програма працює із файлами двох типів: CSV-файли і текстові. CSV-файл – файли із розширенням *.CSV, який відкривається за допомогою програми Microsoft Excel. Текстовий файл – файл із розширенням *.TXT, відкривається за допомогою текстового редактора. Дані файли можуть використовуватись також для завантаження інформації в систему. Працювати з файлами (зберігати, відкривати) можна через пункт меню “Файл”, або обравши відповідну кнопку “Відкрити”, чи ”Зберегти” на панелі інструментів.

3.6. Висновок до третього розділу

В третьому розділі кваліфікаційної роботи розроблена система для дослідження параметрів динамічних примітивів як інструментального програмного засобу, реалізовано інтерфейс програми проведено експериментальні дослідження параметрів динамічних примітивів з метою виявлення властивостей динамічних примітивів.

4. ОХОРОНА ПРАЦІ ТА БЕЗПЕКА У НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1. Основні принципи конструювання робочого місця користувача ЕОМ.

Ергономіка (від грецьк. ἔργον наука про пристосування посадових– у традиційному розумінні –роботи») обов'язків, робочих місць, обладнання та комп'ютерних програм задля створення найбільш безпечних та ефективних умов праці для людини, виходячи з фізичних і психічних особливостей людського організму.

Більш широке визначення ергономіки, яке було прийняте в 2010 році Міжнародною асоціацією ергономіки (IEA) (Міжнародною ергономічною це наукова дисципліна, що вивчає–асоціацією), звучить так: «Ергономіка взаємодію людини та інших елементів системи, а також сфера діяльності щодо застосування теорії, принципів, даних і методів цієї науки для забезпечення благополуччя людини та оптимізації загальної продуктивності системи».

З цього визначення випливають такі головні завдання ергономіки:

1. Проведення досліджень, спрямованих на пристосування елементів системи "людина – трудовий процес" до природних фізичних і психічних можливостей працівника.
2. Прагнення до забезпечення таким шляхом умов для максимальної ефективності праці.
3. Прагнення запобігти всім можливим загрозам для здоров'я працівника.
4. Прагнення до оптимальної витрати біологічних ресурсів у процесі праці.

Загальні ергономічні вимоги для організації робочого місця користувача ПЕОМ (ГОСТ 12.2.049-80, ГОСТ 122032-78, ГОСТ 22269-76). Ці вимоги встановлюють основні параметри робочого місця, оснащеного дисплеєм, і враховують особливість виконуваних робіт.

Параметри робочого місця повинні бути наступними.

Площа кабінету, в якому буде проходити робота повинна бути не менш 6 м², а об'єм не менш 24 м³. Для внутрішньої обробки приміщення повинні використовуватися дифузно-відбивні матеріали з коефіцієнтами відбиття для стелі – 0,7-0,8; для стін – 0,5-0,6; для підлоги – 0,3-0,5.

Конструкція робочого столу повинна забезпечувати оптимальне розміщення на робочій поверхні використовуваного обладнання. Конструкція крісла повинна забезпечувати підтримку раціональної робочої пози під час роботи з відео-дисплейним терміналом (Далі ВДТ) і ПЕОМ, дозволяти змінювати позу з метою зниження статичного напруження м'язів шийно-плечової області і спини для попередження розвитку втоми працюючого (згідно з ГОСТ 12.2.032-78). Поверхня сидіння, спинки та інших елементів стільця (крісла) повинна бути напівм'якою, з покриттям, що не електризується, неслизьке та повітронепроникне, що забезпечує легке очищення від забруднення.

Висота робочої поверхні столу, за відсутності можливості її регулювання повинна складати 725 мм. Робочий стіл повинен мати простір для ніг висотою не менше 600 мм, шириною – не менше 500 мм, не менше 450 мм в глибину на рівні колін і на рівні простягнутої ноги – не менше 650 мм. Робоче місце має бути обладнане підставкою для ніг, має ширину не менше 300 мм, глибину не менше 400 мм, регулювання по висоті в межах 150 мм за кутом нахилу опорної поверхні підставки до 20 градусів.

Відстань від очей користувача до екрану дисплея має становити 500-700 мм. Кут зору 10-20°, але не більше 40°; кут між верхнім краєм дисплея і рівнем очей користувача має становити не менше 10°. Кращим є розташування екрану перпендикулярно до лінії зору користувача.

Робочі місця по відношенню до світлових прорізів повинні розташовуватися не ближче 3 м так, щоб природне світло падало збоку, переважно зліва. Освітленість також впливає на стан здоров'я і працездатність

людини. У відповідності зі СНіП 11-4-79 встановлені наступні вимоги до освітленості:

Для штучного освітлення:

- Комбіноване освітлення – освітленість 1500 лк;
- Загальне освітлення – освітленість 400 лк.

Для природного освітлення:

• Верхнє або комбіноване освітлення – коефіцієнт природної освітленості (далі КПО) 10%;

- Бічне освітлення – КПО 3.5%.

Для суміщеного освітлення:

- Верхнє або комбіноване освітлення – КПО 3-6%;
- Бічне освітлення – КПО 1.1-2%.

До основних показників, що визначають умови здорової роботи, належать: фон, контраст об'єкта з фоном, видимість, показник осліпленості, коефіцієнт пульсації освітленості.

Фон характеризується коефіцієнтом відбиття. Контраст об'єкта з фоном (К) характеризується співвідношенням яскравості розглянутого об'єкта (точки, лінії, знаки) і фону. Оскільки роботи користувача ПЕОМ відносяться до категорії 1а – легкі фізичні роботи (роботи проводяться сидячи і супроводжуються незначним фізичним напруженням, з енерговитратами до 120 ккал / годину), необхідно дотримуватися наступних норм: коефіцієнт відображення більше 0,4, тобто світлий фон; контраст об'єкта з фоном великий і середній при К більше 0,2 (згідно СНіП 11-4-79).

У полі зору користувача ПЕОМ має бути забезпечений відповідний розподіл яскравості. Відношення яскравості екрана до яскравості оточуючих його поверхонь не повинно перевищувати у робочій зоні 3:1 (СНіП 11-4-79). У зв'язку з цим дисплей ПЕОМ повинен відповідати наступним вимогам:

- Яскравість свічення екрану не менше 100 кд/м²;
- Мінімальний розмір світної точки для кольорового дисплея не більше 0,6 мм ;

- Контрастність зображення знаку – не менше 0,8;
- Низькочастотне тремтіння зображення в діапазоні 0,05-1,0 Гц повинно знаходитися в межах 0,1 мм;
- Екран повинен мати покриття антивідблиску;
- Відеомонітор повинен бути обладнаний поворотним майданчиком, що дозволяє переміщати відеотермінал в горизонтальній і вертикальній площинах в межах 130-220 мм і змінювати кут нахилу на 10-15 мм.

Коефіцієнт відбиття світла матеріалами і обладнанням всередині приміщень має велике значення для освітлення: чим більше світла відбивається від поверхонь, тим вище освітленість. Коефіцієнт відображення відповідно повинен бути для: стелі 60-70%, стін 40-50%, підлоги 30%, для інших поверхонь 30-40%.

Результати досліджень показують, що найбільшою мірою негативний фізіологічний вплив на операторів ПК пов'язаний з дискомфорфтними зоровими умовами через неправильно спроектоване освітлення. Згідно СНіП II-4-79 освітленість на горизонтальній площині робочого місця оператора ЕОМ повинна складати 400 лк при висоті цій площині 0,8 м над підлогою.

4.2 Забезпечення захисту працівників суб'єкта господарювання від іонізуючих випромінювань

Іонізуюче випромінювання або радіоактивність є небезпечним явищем для людського організму. При взаємодії впливу іонізаційних випромінювань у навколишнє середовище можуть відбутись різні утворення зарядів . Існують два різновиди випромінювання – «альфа» та «бета».

В залежності від носія та енергії, вони мають різну проникаючу здатність. Альфа це випромінювання яке проявляється важкими частинами складеними з протонів і нейтронів.

В свою чергу бета випромінювання являє собою ланцюг електронів та позитронів які є більшу здатність проникати у середовище. Працюючи на таких територіях, де існує радіаційна атмосфера можуть виникнути різні випадки.

На підприємстві можуть виникнути інциденти при користуванні ядерними матеріалами, зберіганні радіоактивних відходів в наслідок чого працівники можуть отримати травму у вигляді дози опромінення, використання іонізуючих джерел випромінювання.

Також у випадку такої радіаційної аварії забруднюється навколишнє середовище, люди можуть отримати травму у вигляді потужної дози опромінення. Призвести аварію на підприємстві може також якщо активна реакційна речовина знаходиться у роботі та це відбувається незаконно.

Це може привезти до опромінення жителів та перевищити межу дози опромінення. Частинки з цього випромінювання можуть залишати сліди на дихальній системі на травній системі людського організму. Також ці елементи можуть бути у водних каналах, які постачають питну воду людям.

На підприємстві де проводяться роботи з радіаційними речовинами обов'язково мають вживатись заходи проти радіації. Протирадіаційні захисти це така система правових, організаційних норм та санітарної гігієни.

До переліку таких захистів можна включити медичні заходи для забезпечення радіаційної безпеки персоналу та проектно-конструкторські. Для організації заходів проти іонізації опромінювання підприємство має ввести обов'язкові методи щоб подбати про безпеку працюючого персоналу. До таких методів можуть належати заходи які обмежують допуск працівників до джерел які випромінюють радіацію.

До таких працівників можемо віднести таких, які не підходять за віком, за статтю та працівники які вже отримали дозу випромінювання. Підприємство мусить створити сприятливі умови що дотримуються встановлених норм та вимог для працівників та застосовувати індивідуальні засоби для захисту працівника цього підприємства.

Організація повинна контролювати рівні опромінювання та вести інформаційну систему про стан радіації на підприємстві та призначених місць для праці.

На підприємстві повинні бути проведені заходи щодо організації безпеки для робіт які проводяться у радіаційних ділянках а саме: -організація роботи нарядів та розпоряджень; -організація та перевірка пропусків до робочих місць; -оформлення контролю за процесом виконання роботи; -введення примусового часу на перерву та вчасне закінчення робочого процесу.

Реалізувати заходи проти радіації за певний відрізок часу можливо, тим що працівники , які працюють з іонізованими випромінюваннями можуть виконувати вчасно свою роботу ,відповідно керівництво може за якісну роботу зменшити кількість робочих днів у тижні.

Цим самим вони застереженням вони зменшать знаходження працівників у зоні випромінювання та відповідно буде менше контактування з радіаційними приладами. Захистити працівників за допомогою відстані підприємство може шляхом доцільного розміщення приміщення, правильно розставити та розрахувати робочі місця для працівників а також забезпечити приладами, які зможуть контактувати, керувати робочим процесом з технікою яка має радіаційний вплив на відстані.

Слугувати захистом може покриття свинцем меблів які присутні у приміщенні (двері, вікна, робочі столи), створення перекриття між поверхами та перегородки. Працівникам обов'язково має бути виданий спеціальний одяг ,такі як фартухи, шапочки та рукавиці зшиті з просвинцевої тканини.

Розміщення робочих місць повинно мати правильний розрахунок на загальну кімнату, не робити перенабір та забезпечити відповідним та необхідним обладнанням робочі кабінети. При користуванні відкритими приладами іонізованого опромінення провести герметизації цих систем, при можливості використовувати роботу техніки. Підприємство повинне вжити усіх санітарно-гігієнічних заходів та соціальних, а також важливо необхідний є медичний захист робочих на об'єкті.

4.3 Висновок до четвертого розділу

В даному розділі описано основні принципи конструювання робочого місця користувача ЕОМ, зазначено діючі вимоги щодо ергономіки робочого місця. А також визначені заходи та методи із забезпечення радіаційних впливів та іонізації опромінювання на підприємствах. Описані вимоги для керівництва та підлеглих працюючих на об'єктах щодо їхніх дій в разі виникнення радіації .

ВИСНОВКИ

Основні висновки за результатами виконання кваліфікаційної роботи сформовано за пунктами:

1. В результаті аналізу параметрів зображень та елементів відображення інформації виявлено основні властивості зображень, основані на їх сприйнятності, сформульоване поняття якості зображення і визначений внесок кожного окремого показника якості у загальну оцінку зображення. Показано, що застосування конкретних індикаторних елементів в інформаційних системах, які працюють в статичному та динамічному режимах, передбачає їх попередні експериментальні дослідження, що може бути досягнуто шляхом моделювання процесів відображення інформації індикаторними елементами за допомогою спеціально створеної системи дослідження параметрів динамічних примітивів.

2. Обґрунтовано вибір параметрів динамічного примітива при побудові його моделі, таких як форма, розміри, кут повороту, основний та вторинний колір пікселя, колір фону, закон зміни інтенсивності кольору, тривалість періоду, скважність. Розглянуто різновиди функцій зміни інтенсивності, проведений їх математичний аналіз, що дозволило розробити моделі поведінки пікселя. Отримане співвідношення для визначення кольору пікселя в будь-який момент часу в залежності від значень у граничних точках проміжку і закону зміни інтенсивності, за яким колір пікселя визначається сумою його складових відповідно до RGB-моделі, кожна з яких на потоковий момент обчислюється за відповідним законом зміни інтенсивності.

3. Сформульовані вимоги до системи експериментального дослідження динамічних примітивів, які включають вимоги до інтерфейсу, апаратних та програмних засобів, складових системи. Розроблена структура та програмні засоби системи. Наявність підсистеми введення і аналізу експертних оцінок параметрів дозволяє проводити комплексний аналіз за сукупністю параметрів з

визначенням їх коефіцієнта сприйнятності. Розроблено рекомендації по роботі з системою при створенні динамічного примітива і проведенні його оцінювання.

4. Експериментальні дослідження динамічних примітивів з використанням створеної системи довели, що:

- кращою є послідовність аналізу динамічних примітивів, яка полягає в дослідженні коефіцієнта сприйняття для кожного окремого значення параметра примітива, та виборі примітива на основі найкращих параметрів;
- чим вища оцінка сприйнятності у окремих складових параметрів моделі, тим вищим є коефіцієнт сприйнятності для всієї моделі примітива;
- краще сприймаються такі форми, як квадрат, шестикутник, коло;
- краще сприймаються такі кольори примітиву і фону: червоний-салатовий-білий та чорний-білий-коричневий;
- краще сприймаються такі закони зміни інтенсивності, як дискретний, синусоїдальний, лінійний;
- діапазон кращої сприйнятності періоду лежить в межах $1 \div 5$ с.

5. Створена система для дослідження параметрів динамічних примітивів може бути використана при побудові пристроїв відображення величин з використанням динамічних примітивів для попереднього аналізу і оцінювання поведінки примітивів.

ПЕРЕЛІК ДЖЕРЕЛ

1. Цифрова схемотехніка електронних систем. Підручник / В.І., Бойко, В.Я. Жуйков, В.М. Співак, А.А. Зорі, В.В. Багрій, Т.О. Терещенко. 3-тє вид. допов. і переробл. – К.: Вища школа, 2010. – 426 с. (гриф надано МОН України – протокол №1.4/18-Г-414 від 14.02.2008 р.). ISBN 966-642-193-3.
2. Air pollution: ways to forecast and calculate it [Електронний ресурс] // OpenWeather. – 2016. – Режим доступу до ресурсу: <https://openweather.co.uk/blog/post/air-pollution-ways-forecast-andcalculate-it>.
3. Salituro E. Learn Grafana 7.0 / Eric Salituro. – Birmingham: Packt Publishing, 2020. – 410 с.
4. Why Weather Data Should Be Open And Nearly Free [Електронний ресурс] // OpenWeather. – 2020. – Режим доступу до ресурсу: <https://openweather.co.uk/blog/post/why-weather-data-should-be-open-andnearly-free>
5. Grafana vs. Kibana: The Key Differences to Know [Електронний ресурс] // logz.io. – 2020. – Режим доступу до ресурсу: <https://logz.io/blog/grafana-vs-kibana/>.
6. Analysis of Cloud Service Monitoring with Grafana [Електронний ресурс] // keeper. – 2019. – Режим доступу до ресурсу: <https://keeper.io/en/2019/11/analysis-of-cloud-service-monitoring-withgrafana/>.
7. InfluxDB line protocol tutorial [Електронний ресурс] // influxdata – Режим доступу до ресурсу: https://docs.influxdata.com/influxdb/v1.8/write_protocols/line_protocol_tutorial/.
8. Turnbull J. The Art of Monitoring / James Turnbull & Turnbull Press., 2016. – 645 с
9. Цифрова та імпульсна схемотехніка. Моделювання та аналіз / В.В. Макаренко, В.М. Співак. Навчальний посібник з грифом Вченої ради НТУУ «КПІ» (протокол . № 6 від 30 червня 2015 р). – К.: НТУУ «КПІ», 2016. – 314 с.

10. Технології захисту інформації [Електронний ресурс] : підручник для студентів спеціальності 122 «Комп'ютерні науки», спеціалізацій «Інформаційні технології моніторингу довкілля», «Геометричне моделювання в інформаційних системах» / Ю. А. Тарнавський ; КПІ ім. Ігоря Сікорського. – Електронні текстові дані (1 файл: 2,04 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2018. – 162 с. (https://ela.kpi.ua/bitstream/123456789/23896/1/TZI_book.pdf)
11. Рябенький В.М., Жуйков В.Я., Гулий В.Д. Цифрова схемотехніка: Навч. посібник. – Львів: "Новий світ-2000", 2009. – 736 с.
12. Математичні та схемотехнічні основи цифрових пристроїв: Навч. посібник. – Вінниця: УНІВЕРСУМВінниця, 2005. – 165 с.
13. Grafana vs. Zabbix [Електронний ресурс] // MetricFire. – 2019. – Режим доступу до ресурсу: <https://www.metricfire.com/blog/grafana-vs-zabbix/>.
14. Grafana vs. Datadog [Електронний ресурс] // MetricFire. – 2019. – Режим доступу до ресурсу: <https://www.metricfire.com/blog/grafana-vs-datadog/>.
15. Burnett K., Kwong Bor Ng, Park S. A comparison of the two traditions of metadata development /J. of the American Society for Information Science. Special issue on integrating multiple overlapping metadata standards, Vol. 50, Issue 13, 1999, pp.1209- 1217. [Електронний ресурс] — Режим доступу: <http://comminfo.rutgers.edu/~kbnng/publications/1999JASISPark.pdf>
16. Web Ontology Language, Use Cases and Requirements.W3C Recommendation 10 February 2004. URL: [Електронний ресурс] — Режим доступу: <https://www.w3.org/TR/webont-req/OWL>
17. Белинский В.Т. Практичний посібник із навчального конструювання - К.: “Вища школа”, 1992.
18. Конспект лекцій з дисципліни «Програмування для мобільних пристроїв» для студентів денної форми навчання спеціальності 126 «Інформаційні системи та технології» / Укладачі: Готович В.А., Михайлович Т.В. – Тернопіль : Тернопільський національний технічний університет імені Івана Пулюя, 2020. – 216 с.

19. Коноваленко І. В. Платформа .NET та мова програмування C# 8.0 : навчальний посібник / І. В. Коноваленко, П. О. Марущак. – Тернопіль : ФОП Паляниця В. А., 2020. – 320 с.

20. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин ДСанПІН 3.3.2.007-98

21. Електронний ресурс: Довідник функцій C#. <https://docs.microsoft.com/ru-ru/dotnet/csharp>

22. Електронний ресурс: Документація по Visual Studio. <https://docs.microsoft.com/ru-ru/visualstudio/?view=vs-2019>

23. Електронний ресурс: Програмно-методичний комплекс «GRAN». <https://vseosvita.ua/library/vikoristanna-prikladnogo-programnogo-paketu-gran1-naurokah-matematiki-malunki-grafikami-funkcij-80742.html>

24. Системне програмування. Системні сервісні компоненти. Навч. посібник / Дерев'янка О.С., Межеріцький С.Г., Гавриленко С.Ю., Клименко А. М. Харків: НТУ «ХПІ», 2009. 160 с.

25. Харченко В.П., Знаковська Є.А., Бородін В.А. Операційні системи та системи програмування: навч. посіб. Київ: Вид-во Нац. авіац. ун-ту «НАУдрук», 2012. 360с. 41. Шеховцов В.А. Операційні системи: Підручник. Київ: Вид. група BNV, 2005. 576 с.

26. Електронний ресурс: Програмно-методичний комплекс «ТерМ VII». http://www.kspu.edu/About/DepartmentAndServices/DSAICI/internationalprojects/NationalProjects/TerM_VII.aspx?lang=uk

27. Електронний ресурс: Поради для написання керівництва користувача. <http://analyst.by/articles/rukovodstvo-polzovatelya-sovetyi-dlya-sostavleniya>

28. Radkevych, A. V. Comparative analysis of the parameters of the strength of the subgrade at the transition to the higher axial loading up to 25 t. / A. V. Radkevych, V. D. Petrenko, O. L. Tiutkin, V. S. Andrieiev, N. A. Mukhina // IOP

Conference Series: Materials Science and Engineering, Volume 708 (2019), Number 1, 012024

29. Pshinko, O. Comparative analysis of calculation results of supporting structure of soil-cement piles / O. Pshinko, V. Petrenko, A. Tiutkin, V. Andrieiev, A. Hubar, D. Ihnatenko, R. Markul // TRANSPORT MEANS 2019. Sustainability: Research and Solutions. Proceedings of 23rd International Scientific Conference (2-4 October, 2019), Part II, pp. 820-828.

30. Petrenko V. D. Estimation of subgrade strengthening influence using soilcement elements / V. D. Petrenko, O. L. Tiutkin, I. O. Sviatko // Наука та прогрес транспорту. – 2016. – № 4 (64). – С. 161-168.

31. Petrenko, V. D. Complex analysis of subgrade stress-strain state with combined strengthening / V. D. Petrenko, O. L. Tiutkin, I. O. Sviatko, A.M.M. Alhdur // Серія: Галузеве машинобудування, будівництво. – 2017. – Вип. 1 (48). – С. 165-174.

32. Petrenko, V. D. Simulation of Subgrade Embankment on Weak Base / V. D. Petrenko, I. O. Sviatko // Наука та прогрес транспорту. – 2015. – № 4 (58). – С. 198-204.

33. S. Avidan, A. Shamir. Seam carving for content-aware image resizing./ACM Transactions on Graphics (TOG) – Proceedings of ACM SIGGRAPH, Volume 26 Issue 3, July 2007. P. 13–21

34. Perona P., Malik J. Scale-space and edge detection using anisotropic diffusion// IEEE Trans. Pattern Anal. March.Intell. – 1990. – Vol. 12, No. 5. – P. 629– 639.

35. Hanno S. Optimal operators in digital image processing: [Електроний ресурс]. – URL: <http://archiv.ub.uniheidelberg.de/volltextserver/volltexte/2000/962/>

36. Sobel I. History and Definition of the Sobel Operator. [Електроний ресурс].– URL: https://www.researchgate.net/publication/239398674_An_Isotropic_3x3_Image_Gradient_Operator

37. Микитишин А.Г., Митник М.М., Стухляк П.Д., Пасічник В.В. Комп'ютерні мережі [навчальний посібник]. Львів, «Магнолія 2006», 2013. 256с.
38. Gameplay semantics for the adaptive generation of game worlds [Електронний ресурс] – Режим доступу: <https://graphics.tudelft.nl/Publicationsnew/2014/Lop14/Lop14.pdf>.
39. How to use BSP trees to generate game maps [Електронний ресурс] – Режим доступу: <https://gamedevelopment.tutsplus.com/tutorials/how-to-use-bsp-treesto-generate-game-maps-gamedev-12268>.
40. Constrained Elastic SurfaceNets: Generating Smooth Models from Binary Segmented Data [Електронний ресурс] Режим доступу: <https://www.merl.com/publications/docs/TR99-24.pdf>
41. Rendering noise-generated terrain with ray marching [Електронний ресурс] Режим доступу: https://fileadmin.cs.lth.se/cs/Education/EDAN35/projects/16NiklasJohan_Terrain.pdf
42. Understanding Surface Nets [Електронний ресурс] Режим доступу: <https://cerbion.net/blog/understanding-surface-nets>
43. Ray Marching [Електронний ресурс] Режим доступу: <https://michaelwalczyk.com/blog-ray-marching.html>
44. Polygonising a scalar field [Електронний ресурс] Режим доступу: <http://paulbourke.net/geometry/polygonise>
45. Осадчий В.В. Основи розробки веб-додатків. Навчальний посібник / В.В. Осадчий, В.С. Круглик – Мелітополь: ТОВ «Видавничий будинок ММД», 2012. – 540 с.
46. Smooth Voxel Mapping: a Technical Deep Dive on Real-time Surface Nets and Texturing [Електронний ресурс] Режим доступу: <https://bonsairobo.medium.com/smooth-voxel-mapping-a-technical-deep-dive-on-real-50-time-surface-nets-and-texturing-ef06d0f8ca14>
47. Coding Adventure: Marching Cubes [Електронний ресурс] Режим доступу: <https://www.youtube.com/watch?v=M3iI2l0ltbE>

48. An Implementation of the Marching Cubes Algorithm [Електронний ресурс] Режим доступу: https://www.cs.carleton.edu/cs_comps/0405/shape/marching_cubes.html
49. Coding Adventure: Ray Marching [Електронний ресурс] Режим доступу: <https://www.youtube.com/watch?v=Cp5WWtMoeKg>
50. Петрик М.Р. Моделювання програмного забезпечення : науково методичний посібник / М.Р. Петрик, О.Ю. Петрик– Тернопіль : Вид-во ТНТУ імені Івана Пулюя, 2015. – 200 с.
51. Ray Marching for Dummies! [Електронний ресурс] Режим доступу: <https://www.youtube.com/watch?v=PGtv-dBi2wE>
52. Volume Rendering for Developers: Foundations [Електронний ресурс] Режим доступу: <https://www.scratchapixel.com/lessons/3d-basic-rendering/volumerendering-for-developers/ray-marching-algorithm>
53. Ковальов К.В. ПРОЄКТУВАННЯ СИСТЕМИ АДАПТИВНОЇ ГЕНЕРАЦІЇ ІГРОВОГО ПРОСТОРУ //3-я Міжнародна науково-практична конференція «SCIENTIFIC PROGRESS: INNOVATIONS, ACHIEVEMENTS AND PROSPECTS» Зб. матеріалів форуму. Т.7. Конференція «Технічні науки» – Харків: ХНУРЕ. 2022. – С.135-137.
54. Пасічник О.Г., Пасічник О.В., Стеценко І.В. Основи веб-дизайну. [Навч. посіб.]. К.: Вид. група ВHV. 2009. 336 с.
55. Коноваленко І.В., Федорів П.С. Системне програмування у Windows з прикладами на Delphi. Навч. посіб. Для тех. спец. Вищих навчальних закладів. Тернопіль: ТНТУ ім. І. Пулюя, 2012. 320 с.

ДОДАТКИ

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ
УНІВЕРСИТЕТ ІМЕНІ ІВАНА ПУЛЮЯ**

МАТЕРІАЛИ

XI НАУКОВО-ТЕХНІЧНОЇ КОНФЕРЕНЦІЇ

**«ІНФОРМАЦІЙНІ МОДЕЛІ,
СИСТЕМИ ТА ТЕХНОЛОГІЇ»**



13-14 грудня 2023 року

**ТЕРНОПІЛЬ
2023**

УДК 001
М34

ПРОГРАМНИЙ КОМІТЕТ

Голова: Приймак Микола – професор кафедри комп'ютерних систем та мереж, д.т.н., професор.

Співголови: Марущак Павло – проректор з наукової роботи, докт. техн. наук, професор.

Баран Ігор – канд. техн. наук, доцент, декан факультету ФІС.

Науковий секретар: Семенишин Галина – старший викладач.

Члени: Василь Кривень - завідувач кафедри математичних методів в інженерії д.ф.-м.н., професор; Галина Осухівська – завідувач кафедри комп'ютерних систем та мереж, к.т.н., доцент; Микола Карпінський - професор кафедри кібербезпеки, д.т.н., професор; Жанна Баб'як - завідувач кафедри української та іноземних мов, к.пед. н., доцент; Ярослав Литвиненко – професор кафедри комп'ютерних наук, д.т.н., професор; Михайло Петрик - завідувач кафедри програмної інженерії, д.ф.-м.н., професор; Наталія Загородна – завідувач кафедри кібербезпеки, к.т.н., доцент.

ОРГАНІЗАЦІЙНИЙ КОМІТЕТ

Голова: Скоренький Юрій Любомирович – канд. техн. наук, доцент кафедри фізики.

Члени: доцент кафедри комп'ютерних наук, к.т.н. В. Никитюк; доцент кафедри програмної інженерії, к.т.н. Д. Михалик; доцент кафедри кібербезпеки, к.т.н. М. Стадник; асистент Н. Шаблій; ст. викладач Л. Джиджора.

Матеріали XI науково-технічної конфіції «Інформаційні моделі, системи та технології» Тернопільського національного технічного університету імені Івана Пулюя, (Тернопіль, 13-14 грудня 2023 р.). – Тернопіль: Тернопільський національний технічний університет імені Івана Пулюя, 2023. – 257 с.

Адреса оргкомітету: ТНТУ ім. І. Пулюя, м. Тернопіль, вул. Руська, 56, 46001, тел. (0352) 52-41-33, факс (0352) 254983.

E-mail: conffis2023@gmail.com

Редагування, оформлення, верстка: Семенишин Г.М.

СЕКЦІЇ КОНФЕРЕНЦІЇ, ЯКІ ПРЕДСТВЛЕНІ В ЗБІРНИКУ

- Математичне моделювання;
- Інформаційні системи та технології;
- Комп'ютерні системи та мережі;
- Програмна інженерія та моделювання складних розподілених систем;
- Новітні фізико-технічні та освітні технології.

В збірнику надруковано тези доповідей XI науково-технічної конференції «Інформаційні моделі, системи та технології» (Тернопіль, 13-14 грудня 2023 р.) за такими науковими напрямками: математичне моделювання; інформаційні системи та технології; комп'ютерні системи та мережі; програмна інженерія та моделювання складних розподілених систем; новітні фізико-технічні та освітні технології.

Розрахований на науковців, викладачів та студентів вузів.

За зміст тез та дотримання норм академічної доброчесності відповідальність несе автор.

В.В. Никитюк, М.В. Тененський, А.В. Орловська АНАЛІЗ ВИКОРИСТАННЯ EDA ДЛЯ ВИРІШЕННЯ ПРОБЛЕМ СУЧАСНИХ ЗАСТОСУНКІВ ТА СИСТЕМ V.V. Nykytyuk, M.V. Tenenskyi, A.V. Orlovska ANALYSIS OF EDA USAGE FOR SOLVING PROBLEMS OF MODERN APPLICATIONS AND SYSTEMS	89
Олексяк В.Д., Фризм.Є. ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ АНАЛІЗУ ТА ПРОГНОЗУВАННЯ КУРСУ КРИПТОВАЛЮТ Oleksiak V.D., Mykhailo Fryz INFORMATION TECHNOLOGIES FOR THE ANALYSIS AND FORECASTING OF CRYPTOCURRENCIES	91
Максим Орлінський АНАЛІЗ ПАРАМЕТРІВ ЗОБРАЖЕНЬ Maxim Orlnskyi ANALYSIS OF IMAGES PARAMETERS	92
М. Петрошук, Я.В. Литвиненко АСПЕКТИ ЦИФРОВОЇ ТРАНСФОРМАЦІЇ ВИЩОЇ ОСВІТИ M. Petroshuk, Ia.V. Lytvynenko ASPECTS OF DIGITAL TRANSFORMATION OF HIGHER EDUCATION	93
Олег Пігур МЕТОДИ ГЛИБОКОГО НАВЧАННЯ ДЛЯ ОБРОБКИ ТЕСТОВОЇ ІНФОРМАЦІЇ З ФІНАНСОВИХ СОЦІАЛЬНИХ МЕРЕЖ Oleg Pigur METHODS OF DEEP LEARNING FOR PROCESSING FINANCIAL SOCIAL NETWORK DATA	94
Ігор Пінецький ЗАСТОСУВАННЯ ГЛИБОКИХ ЗГОРТКОВИХ НЕЙРОННИХ МЕРЕЖ У РЕСУРСНО ОБМЕЖЕНИХ ПРИСТРОЯХ ДЛЯ ВИЯВЛЕННЯ ПОЖЕЖ Ihor Pinetskyi APPLICATION OF DEEP CONVOLUTIONAL NEURAL NETWORKS IN RESOURCE- CONSTRAINED DEVICES FOR FIRE DETECTION	95
А.П. Мар'ян, П.П. Пірда, М.С. Матлага, В.П. Лехняк АВТОМАТИЗОВАНА СИСТЕМА КОНТРОЛЮ РОЗМІРІВ ПРИ МЕХАНІЧНІЙ ОБРОБЦІ A. P. Marian, P. P. Pirda, M. S. Matlaha, V. P. Lekhniak AUTOMATED DIMENSION CONTROL SYSTEM DURING MECHANICAL PROCESSING	96
О.Є. Подвисоцький; Н.Б. Стадник МЕТОДИ РОЗПІЗНАВАННЯ ОБЛИЧЬ В СИСТЕМАХ ІДЕНТИФІКАЦІЇ КОРИСТУВАЧІВ РОЗУМНОГО БУДИНКУ O.E. Podvysotskyi; N.B. Stadnyk METHODS OF FACE RECOGNITION IN SMART HOUSE USER IDENTIFICATION SYSTEMS	98
Т.І. Кужда, А.В. Поливода РОЛЬ ТА ЗНАЧЕННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ УПРАВЛІННЯ ТА КОНТРОЛЮ У ВІЙСЬКОВІЙ СПРАВІ T. Kuzhda; A. Polyvoda THE ROLE AND SIGNIFICANCE OF THE AUTOMATED MANAGEMENT AND CONTROL SYSTEM IN MILITARY AFFAIRS	99
О.З. Порохняк, Я.А. Бойчук, М. М. Егреші, О.В. Тотосько АНАЛІЗ ВИБОРУ ПРОМІСЛОВИХ РОБОТІВ ДЛЯ ОПЕРАЦІЙ ФРЕЗЕРУВАННЯ O. Z. Porokhniak, Y. A. Boichuk, M. M. Ehreshi, O. V. Totosko ANALYSIS OF THE SELECTION OF INDUSTRIAL ROBOTS FOR MILLING OPERATIONS	101

УДК 004.031.6

Максим Орлінський

Тернопільський національний технічний університет імені Івана Пулюя

АНАЛІЗ ПАРАМЕТРІВ ЗОБРАЖЕНЬ

Maxim Orlinskyi

ANALYSIS OF IMAGES PARAMETERS

Аналіз зображення – виділення з зображення потрібної інформації за допомогою автоматичних чи напівавтоматичних приладів і систем. У літературі ця область має також інші назви: виділення даних із зображення, аналіз сцен, опис зображень, автоматичне дешифрування, розпізнавання зображень і т. ін.

На відміну від інших видів обробки зображень, таких, як кодування, реставрація і поліпшення якості зображень, результатом аналізу зображень звичайно є не картинка, а її числовий опис. У системах аналізу зображень часто використовуються апріорні дані про зображувані об'єкти і про їх взаємовідношення. У такій системі аналізу можуть також використовуватися методи штучного інтелекту для керування різними блоками системи й організації ефективного доступу до бази апріорних даних про об'єкти.

Ознакою зображення називається його найпростіша відмінна чи характерна властивість. Деякі ознаки є природними, тобто вони встановлюються візуальним аналізом зображення, тоді як інші, так звані штучні ознаки, отримують у результаті його спеціальної обробки або вимірювання.

Найважливішим параметром зображення є світлота. Світлота виражається через такі величини, як яскравість, спектральна інтенсивність випромінювання, координати кольору, які будуть називатися ознаками яскравості. Вимірювання ознак яскравості можна проводити в окремих точках зображення, або в їх межах. Наприклад, середня яскравість околиці точки (j, k) зображення розміром $(2W+1) \times (2W+1)$ елементів визначається як:

$$\bar{Y}(j, k) = \left[1 / (2W + 1)^2 \right] \sum_{m=-W}^W \sum_{n=-W}^W Y(j + m, k + n) \cdot$$

Існує безліч різних способів визначення ознак яскравості [45,46]. Можна використовувати значення яскравості чи координат кольору безпосередньо, або перейти до нових ознак яскравості, виконавши деяке лінійне і нелінійне перетворення.

Вимірювання ознак яскравості набуває особливої важливості при виділенні об'єктів (символічний опис) і при їх маркуванні (інтерпретація).

Дискретне зображення – масив чисел $F(j, k)$ – можна представити як двовимірний випадковий процес, який описується спільними розподілами ймовірностей. Існують розроблені методи оцінки розподілів ймовірностей значень яскравості зображення. Отримані при цьому гістограми можна використовувати також для створення деякого класу ознак зображення.

Розподіл ймовірностей значень яскравості першого порядку можна визначити як:

$$P(b) \equiv \Pr\{F(j, k) = b\},$$

де $0 \leq b \leq L - 1$ – рівні квантування. Розподіл частот першого порядку, який оцінює $P(b)$, описується простим виразом:

$$P(b) \approx N(b) / M,$$

де M – повне число елементів зображення у вікні з центром (j, k) , а $N(b)$ – число елементів у вікні, що мають рівень b . Часто, виходячи з припущення про стаціонарність, розмір вікна приймається рівним розміру всього зображення.

Міністерство освіти і науки України
Тернопільський національний технічний університет
імені Івана Пулюя
Маріборський університет (Словенія)
Технічний університет в Кошице (Словаччина)
Каунаський технологічний університет (Литва)
Львівський національний університет
імені Івана Франка,
Гірничо-металургійна академія ім. Станіслава Сташиця (Польща)
Луцький національний технічний університет,
Чернівецький національний університет
імені Юрія Федьковича,
Вроцлавський економічний університет (Польща)
Університет технологій та економіки
імені Хелени Ходковської (Польща)
Донбаська державна машинобудівна академія



*Студентське наукове
товариство*



VII МІЖНАРОДНА
студентська науково - технічна конференція
"ПРИРОДНИЧІ ТА ГУМАНІТАРНІ
НАУКИ.

АКТУАЛЬНІ ПИТАННЯ"

25-26 квітня 2024 р.

(збірник тез конференції)

Тернопіль 2024

ББК 72+34 (Укр)
МЗ4

Матеріали VII Міжнародної студентської науково - технічної конференції / Тернопіль: Тернопільський національний технічний університет ім. І. Пулюя (м. Тернопіль, 25-26 квітня 2024 р.), 2024.- 308 с.

В збірнику друкуються матеріали VII Міжнародної студентської науково-технічної конференції. Тернопіль. – ТНТУ ім. І. Пулюя (25-26 квітня 2024 р.) за наступними науковими напрямками:

культура і мистецтво; гуманітарні науки; соціальні та поведінкові науки; управління та адміністрування; природничі науки; математика та статистика; інформаційні технології; механічна інженерія; електрична інженерія; автоматизація та приладобудування; хімічна та біоінженерія; електроніка та телекомунікації; виробництво та технології; архітектура та будівництво; аграрні науки та продовольство; сфера обслуговування; транспорт.

Редакційна колегія:

д.е.н. Богдан Андрушків, д.т.н. Олег Ляшук, д.т.н. Ігор Стадник, д.ф.н. Анатолій Довгань, д.ф.н. Андрій Криськов, д.т.н. Володимир Андрійчук, д.т.н. Анатолій Лупенко, к.ф.-м.н. Михайло Михайлишин, д.т.н. Михайло Пилипець, д.т.н. Роман Рогатинський, д.т.н. Петро Стухляк, д.т.н. Михайло Паламар, д.е.н. Наталія Кирич, д.т.н. Микола Підгурський, д.т.н., Микола Приймак, д.т.н. Василь Васильків, д.б.н. Володимир Юкало, д.в.н. Микола Кухтин, д.т.н. Богдан Яворський, к.ф.-м.н. Борис Шелестовський, д.ф.-м.н. Василь Кривень, д.т.н. Павло Марущак, д.е.н. Олена Панухник, д.е.н. Володимир Фалович, д.т.н. Тетяна Вітенько, д.т.н. Володимир Дзюра, д.т.н. Віктор Барановський, д.ф.-м.н. Михайло Петрик, д.е.н. Роман Шерстюк.

Комп'ютерний набір, верстка та редагування:
науковий секретар Ігор Окіпний

Адреса конференції:

46001, м. Тернопіль, вул. Руська, 56

Тернопільський національний технічний університет ім. Івана Пулюя

e-mail: snt@tntu.edu.ua

Тернопільський національний технічний університет ім. Івана Пулюя

Мац О. ВИБІР ТЕХНОЛОГІЙ ДЛЯ РОЗРОБКИ АВТОМАТИЗОВАНОЇ СИСТЕМИ ДЛЯ СУПРОВОДУ ПРОЦЕСІВ РЕЄСТРАЦІЇ ТА ЖИТТЄВОГО ЦИКЛУ ДОМЕННИХ ІМЕН	94
Озіранець В.С.В., Карнаухов А.К., Орловська А.В. АНАЛІЗ ЕФЕКТИВНОСТІ СИСТЕМ НАКЛАДАННЯ МАТЕРІАЛІВ З ВИКОРИСТАННЯМ МЕТОДУ СААТІ	96
Озіранець В.С.В., Орловська А.В. ГІБРИДНИЙ ПІДХІД ДО СТВОРЕННЯ МАТЕРІАЛІВ У ТРИВИМІРНІЙ ГРАФІЦІ	98
Онищук В. СТРАТЕГІЯ АТАКИ НА МЕРЕЖУ SYN-FLOOD ПРИ ІНСТРУМЕНТІ NPING3	100
Орлінський М. АНАЛІЗ ПАРАМЕТРІВ ПРИСТРОЇВ ВІДОБРАЖЕННЯ ІНФОРМАЦІЇ	102
Возьна Х., Яцишин В. КОНЦЕПТУАЛЬНА АРХІТЕКТУРА АВТОМАТИЗОВАНОЇ СИСТЕМИ УПРАВЛІННЯ БІБЛІОТЕКОЮ	104
Пишний М. «МОДЕЛЬ УЧНЯ» В КОМП'ЮТЕРНИХ НАВЧАЛЬНИХ СИСТЕМАХ	105
Пік М. ПОБУДОВА ЛІНІЇ ПЕРЕТИНУ ПОВЕРХОНЬ З ВИКОРИСТАННЯМ ГРАФІЧНОГО ПАКЕТУ AUTOCAD	107
Полевий В. ЯК FRV ТЕХНОЛОГІЇ ЗМІНЮЮТЬ ОБЛИЧЧЯ ВІЙНИ	108
Приймаченко М. SIMULATION TOOLS FOR NETWORK TECHNOLOGY RESEARCH	109
Рокош М. РОЗРОБКА БРАУЗЕРНОГО РОЗШИРЕННЯ ДЛЯ ГЕНЕРАЦІЇ ЕМЕЙЛІВ З ДОПОМОГОЮ ШТУЧНОГО ІНТЕЛЕКТУ	110
Рокош М. ГЕНЕРАТИВНИЙ ШТУЧНИЙ ІНТЕЛЕКТ І ЙОГО ВПЛИВ НА СВІТОВУ ЕКОНОМІКУ ТА ЕФЕКТИВНІСТЬ БІЗНЕСУ	112
Марціяш Г. Я., Сербін В. С., Смага І. В. РЕВОЛЮЦІЯ ВИРОБНИЦТВА: ВПЛИВ ШТУЧНОГО ІНТЕЛЕКТУ НА ОПТИМІЗАЦІЮ ТА АВТОМАТИЗАЦІЮ ВИРОБНИЧИХ ПРОЦЕСІВ	114

УДК 004.031.6

Орлінський М. – ст. гр. СНм-61

Тернопільський національний технічний університет імені Івана Пулюя

АНАЛІЗ ПАРАМЕТРІВ ПРИСТРОЇВ ВІДОБРАЖЕННЯ ІНФОРМАЦІЇ

Науковий керівник: д.е.н., доцент Матійчук Л.П.

Orlinskyi M.

Ternopil Ivan Puluj National Technical University

ANALYSIS OF THE PARAMETERS OF INFORMATION DISPLAY DEVICES

Supervisor: Matiichuk Liubomyr

Важливим параметром світло випромінюючих пристроїв є світимість або яскравість. Цей параметр визначає наскільки успішно дисплей конкурує з зовнішнім освітленням, достатньо складна проблема – використання дисплеїв в умовах яскравого сонячного світла. Яскравість звичайно виражається в фут-ламбертах (фут·Лб), інколи в нітах, або в канделах на квадратний метр. Ці одиниці враховують спектральну чутливість усередненого людського зору. Їх називають фотометричними одиницями.

Швидкодія пристроїв виведення характеризує максимально можливий темп прийому, відображення і зміни інформації. Одна з характеристик швидкодії – час відтворення знака, який вимірюється від моменту надходження кодової посилки до моменту повного відображення знака в заданому місці екрану. Час відтворення знака залежить від типу дисплею, схемних рішень, елементів, застосованих у схемі, і виду індикаторних елементів. Швидкодія систем відображення характеризується часом виклику і часом відновлення даних. Під часом виклику розуміють час, який вимірюється з моменту подачі команди на відображення потрібної інформації до моменту її відтворення. Цей час визначається в основному часом вибірки необхідної інформації з пам'яті ЕОМ і часом відтворення її на пристрої виведення.

Розглянемо характеристику дисплеїв – колір. Спектральний розподіл випромінюваного світла повинен співвідноситись зі спектральною чутливістю ока, яка зображена на рис. 1.

Контраст дисплея є властивістю границі світло-темно. Контраст дорівнює відношенню яскравості: (максимум–мінімум)/мінімум. Іноді контраст виражається у вигляді нормалізованого коефіцієнта: (максимум–мінімум) / (максимум + мінімум), найбільша величина якого дорівнює одиниці. Мінімальна яскравість – це яскравість дисплея у виключеному стані, залишкова яскравість за рахунок зовнішнього освітлення. Мінімальну яскравість можна зменшити за допомогою затінення фільтром, який прозорий лише для найяскравішої частини спектра. Сіра шкала визначається як число ступенів, які дають зміну інтенсивності випромінювання в $\sqrt{2}$ разів.

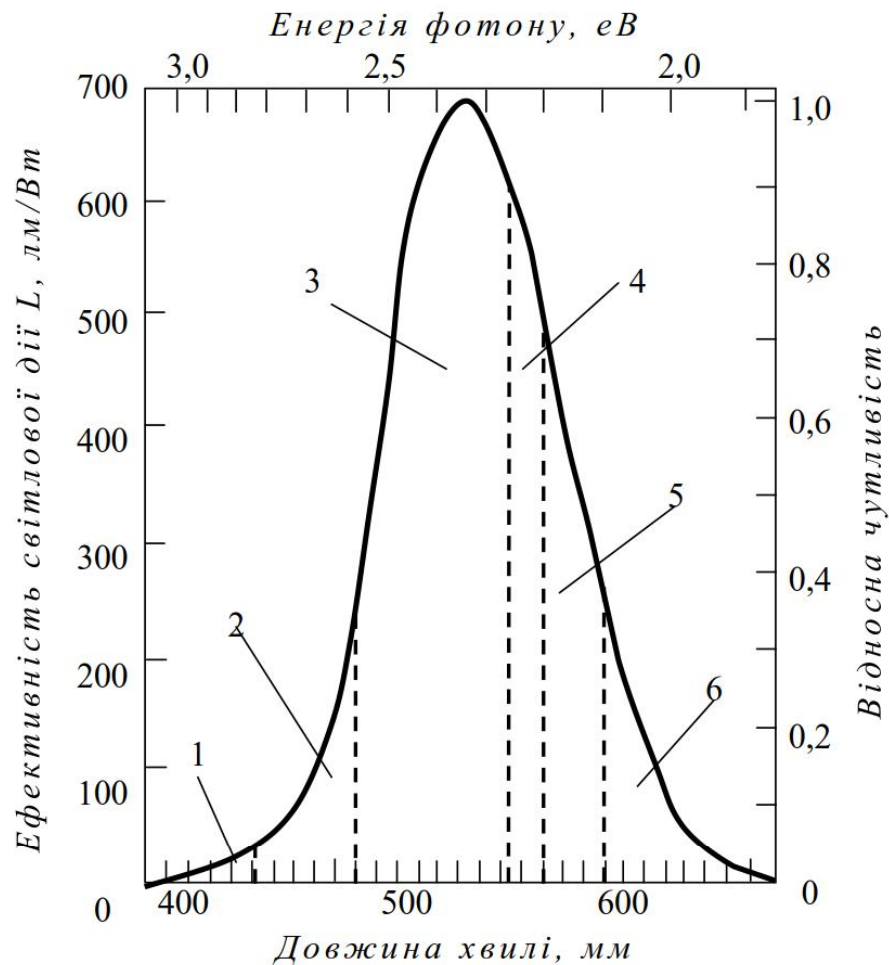


Рис 1. Спектральна чутливість людського ока:
 1 – фіолетовий; 2 – синій; 3 – зелений; 4 – жовтий;
 5 – оранжевий; 6 – червоний.

Із цієї діаграми видно, що випромінювання потужністю 100 мкВт на вузькій ділянці червоного діапазону в районі 700 нм здається менш яскравим, ніж випромінювання потужністю 100 мкВт при 556 нм, або менш яскравим ніж широкий спектр з максимумом при 700 нм, який простягається в бік менших довжин хвиль. Червоне світло потужністю 1 мВт викликає появу болю в очах, тому існує допустимі границі потужності.

Великою перевагою є видимість дисплея у всіх напрямках. Однак часто приходится жертвувати цією можливістю для дисплеїв індивідуального користування (наприклад, у годинниках і калькуляторах). Пластмасові лінзи, які концентрують світло у вузький пучок, збільшують яскравість приладу. Контраст рідино кристалічних дисплеїв зменшується, коли кут зору сильно відхиляється від кута нормального падіння внаслідок більшої довжини оптичного шляху в рідинному кристалі з подвійною променезаломлюваністю. Це призводить до генерації еліптично поляризованого світла, яке не вдається цілком погасити плоскими поляроїдами.

ЛІСТИНГ ОСНОВНИХ МОДУЛІВ ПРОГРАМИ

Main.cpp

```

namespace dp {
/* BEGINNING OF NAMESPACE*/

double mixture(double a, double b, double p) { return a+p*(b-a); }

/* ----- Color functions ----- */

struct Color
{
    /* RGB */
    double components[3];
    AnsiString to_string()
    {
        AnsiString res;
        res.printf("%.2f, %.2f, %.2f", components[0],
            components[1], components[2]);
        return res;
    }
    void from_string(char *s) {
        sscanf(s, "%lf,%lf,%lf", &components[0], &components[1],
            &components[2]);
    }
    inline double & r() { return components[0]; }
    inline double & g() { return components[1]; }
    inline double & b() { return components[2]; }
    Color() { components[0]=components[1]=components[2]=0; };
    Color(TColor c) {
        import( ((c & 0x0000ff) >> 0) / 255.0,
            ((c & 0x00ff00) >> 8) / 255.0,
            ((c & 0xff0000) >> 16) / 255.0);
    };
    Color(double c1,double c2,double c3) {
        import(c1,c2,c3);
    };
    TColor get_TColor() {
        return TColor( int(r()*255) + (int(g()*255)<<8) + (int(b()*255)<<16));
    }
    void import(double c1,double c2,double c3) {
        components[0]=c1;components[1]=c2;components[2]=c3;
    };
    Color blend_with(Color c, double proportion) {
        return Color(
            mixture(r(),c.r(),proportion),
            mixture(g(),c.g(),proportion),

```

```

        mixture(b(),c.b(),proportion)
    );
};

/* ----- Primitive functions ----- */

struct Transformation
{
    double
        scale_x, scale_y,
        offset_x, offset_y,
        rotation;
    Transformation() : scale_x(0.5), scale_y(0.5),
        offset_x(0), offset_y(0), rotation(0) {};
};

struct Function
{
    enum Dependency {
        Linear, Sinus, Exponent2, Exponent4, Exponent10,
        IExponent2, IExponent4, IExponent10,
    } function;
    static char * DependencyNames [] ;

    double start_value, end_value;

    static const int preset_num=10;
    struct preset {
        double start_value, end_value; Dependency d;
        char * title;
    };
    static struct preset presets[preset_num+1];

    void set_style(int id) {
        start_value=presets[id].start_value;
        end_value=presets[id].end_value;
        function=presets[id].d;
    }
    int get_preset_id()
    {
        int i;
        for(i=0;i<preset_num;++i)
            if(start_value==presets[i].start_value
                && end_value==presets[i].end_value &&
                function==presets[i].d)
                break;
        return i;
    }
    Function() : start_value(0.0), end_value(1.0) {
};
AnsiString as_text() {
    char *func="";
    switch(function) {

```

```

    case Linear: func="L"; break;
    case Sinus: func="S"; break;
    case Exponent2: func="E2"; break;
    case Exponent4: func="E4"; break;
    case Exponent10: func="E10"; break;
    case IExponent2: func="IE2"; break;
    case IExponent4: func="IE4"; break;
    case IExponent10: func="IE10"; break;
}
AnsiString res;
res.printf("[%.1lf:%.1lf]: %s",
    start_value, end_value, func);
return res;
};
double get_value(double x) {
    double value;
    switch(function) {
        case Linear:
            value=x; break;
        case Sinus:
            value=sin(M_PI*(x-0.5))*0.5+0.5; break;
        case Exponent2:
            value=normal_exp(x,2); break;
        case Exponent4:
            value=normal_exp(x,4); break;
        case Exponent10:
            value=normal_exp(x,10); break;
        case IExponent2:
            value=normal_exp(1.0-x,2); break;
        case IExponent4:
            value=normal_exp(1.0-x,4); break;
        case IExponent10:
            value=normal_exp(1.0-x,10); break;
        default: value=0;
    }
    return mixture(start_value, end_value, value);
};
};

char * dp::Function::DependencyNames [] = {
    "лінійний", "синус", "експ2", "експ", "експ10",
    "об. експ2", "об. експ", "об. експ10",
};

dp::Function::preset dp::Function::presets[preset_num+1] = {
    { 0.0, 1.0, Function::Linear, "Лінійний 1"},
    { 1.0, 0.0, Function::Linear, "Лінійний 2" },
    { 0.0, 1.0, Function::Sinus, "Синусоїдний 1"},
    { 1.0, 0.0, Function::Sinus, "Синусоїдний 2"},
    { 0.0, 0.0, Function::Linear, "Дискретний 1"},
    { 1.0, 1.0, Function::Linear, "Дискретний 2"},

    { 1.0, 0.0, Function::Exponent4, "Експон. 1"},
    { 1.0, 0.0, Function::IExponent4, "Експон. 2"},
    { 0.0, 1.0, Function::IExponent4, "Експон. 3"},
    { 0.0, 1.0, Function::Exponent4, "Експон. 4"},
    { 0, 0, Function::Linear, 0},
};

```



```
};
```

```
struct Interval
```

```
{
```

```
    Function f;  
    double duration;
```

```
    AnsiString as_text() {
```

```
        return AnsiString((double)duration) + AnsiString(" s., ") + f.as_text();
```

```
    };
```

```
};
```

```
struct DynamicParams
```

```
{
```

```
    /* конструктор */
```

```
    DynamicParams() : interval_number(0), start_time(0),  
        on_off_ratio(0.7), time_scale(1.0) {};
```

```
    int interval_number;
```

```
    double on_off_ratio, time_scale;
```

```
    Interval intervals[max_intervals];
```

```
    double start_time;
```

```
    void del_interval(int id)
```

```
    {
```

```
        if(id >= interval_number) return;
```

```
        if(interval_number == 1) return;
```

```
        interval_number--;
```

```
        for(int i=id; i < interval_number; ++i)
```

```
            memcpy(&intervals[i], &intervals[i+1], sizeof(Interval));
```

```
    }
```

```
    void add_interval(int after)
```

```
    {
```

```
        if(interval_number == max_intervals) return;
```

```
        for(int i=interval_number; i > after; --i)
```

```
            memcpy(&intervals[i], &intervals[i-1], sizeof(Interval));
```

```
        interval_number++;
```

```
    }
```

```
    double get_frame(double ctime) {
```

```
        return fmod((double)(ctime-start_time)/time_scale, 1.0);
```

```
    }
```

```
    int get_interval_for_fraction(double cfrac) {
```

```
        return get_interval_for_frame(cfrac);
```

```
    }
```

```
    int get_interval_for_frame(double cframe) {
```

```
        cframe = (cframe - (1 - on_off_ratio) * 0.5) / on_off_ratio;
```

```
        if(cframe < 0) return -1;
```

```
        if(cframe > 1) return -1;
```

```
        for(int i=0; i < interval_number; ++i) {
```

```
            if(cframe > intervals[i].duration)
```

```
                cframe -= intervals[i].duration;
```

```
            else {
```

```
                return i;
```

```
            }
```

```

    }
    /* UNREACHANBLE */
    return -1;
}
int get_interval(double ctime) {
    if(interval_number==0) return 0;
    return get_interval_for_frame(get_frame(ctime));
}

double get_intensity_for_fraction(double cframe) {
    return get_intensity_for_frame(cframe);
}

double get_intensity_for_frame(double cframe) {
    cframe=(cframe-(1-on_off_ratio)*0.5)/on_off_ratio;

    if(cframe<0.0) return 0.0;
    if(cframe>1.0) return 0.0;

    for(int i=0;i<interval_number;++i) {
        if(cframe > intervals[i].duration)
            cframe-=intervals[i].duration;
        else {
            return intervals[i].f.get_value(
                cframe/intervals[i].duration);
        }
    }
    /* UNREACHANBLE (?) */
    return 1.0;
}

double get_intensity(double ctime) {
    if(interval_number==0) return 1.0;
    return get_intensity_for_frame(get_frame(ctime));
};

};

dp::Color current_c;

/* Основнмй клас -- примітив */
struct Primitive
{
    bool changed;
    double exp_coef;
    AnsiString name, save_date;
    enum Forms {
        None, Triangle_EL, Triangle_R, Triangle_ISO, /*3*/
        Quadrangle, Pentagon, Hexagon, /*6*/
        Circle, Ellipse, Rectangle, /*9*/
        Parallelepiped, Trapezium, Octagon,
    } form;

    static bool show_abc[][3];
    static double default_data[][3];
    static char * FormNames [];

```

```

/* стороны */
double a,b,c;
Color main_color, background_color, sec_color;
Transformation trans;
Primitive() : form(Primitive::None), a(1), b(1), c(1),
exp_coef(0), changed(true) { SetNameFromTime(); };
DynamicParams dyn;

void draw();

Color get_color(double ctime) {
return get_color_for_frame(dyn.get_frame(ctime));
};

Color get_color_for_frame(double cframe) {
double i=dyn.get_intensity_for_frame(cframe);
int interval=dyn.get_interval_for_frame(cframe);
Color c1=main_color;
return main_color.blend_with(sec_color, i);
};

AnsiString GetTime()
{
struct tm *time_now;
time_t secs_now;
char str[80];
tzset();
time(&secs_now);
time_now = localtime(&secs_now);
strftime(str, 80, "%y-%m-%d-%H:%M:%S", time_now);
return AnsiString(str);
}
void SetNameFromTime()
{
name=GetTime();
}

private:
void draw_Ngon(int N) {
double x=0,y=1;
for(int i=0;i<N;++i) {
dp::Color c=current_c;
CVertex Points[3];
Points[0].Create( 0.0, 0.0, 0.0, c.r(), c.g(), c.b());
Points[1].Create( x, y, 0.0, c.r(), c.g(), c.b());
x=sin(2*M_PI*(double)(i+1)/N);
y=cos(2*M_PI*(double)(i+1)/N);
Points[2].Create( x, y, 0.0, c.r(), c.g(), c.b());

LastHr = lpDevice->DrawPrimitive(D3DPT_TRIANGLELIST, D3DFVF_XYZ |
D3DFVF_DIFFUSE,
Points, 3, 0);
}
};
void draw_4points(double x1,double y1, double x2, double y2,
double x3, double y3, double x4, double y4) {

```

```

CVertex Points[6];
dp::Color c=current_c;
Points[0].Create(x1,y1, 0.0, c.r(), c.g(), c.b());
Points[1].Create(x2,y2, 0.0, c.r(), c.g(), c.b());
Points[2].Create(x3,y3, 0.0, c.r(), c.g(), c.b());

Points[4].Create(x3,y3, 0.0, c.r(), c.g(), c.b());
Points[5].Create(x4,y4, 0.0, c.r(), c.g(), c.b());
Points[3].Create(x1,y1, 0.0, c.r(), c.g(), c.b());

lpDevice->DrawPrimitive(D3DPT_TRIANGLELIST, D3DFVF_XYZ | D3DFVF_DIFFUSE,
    Points, 6, 0);
}

void draw_rect() {
    CVertex Points[6];
    dp::Color c=current_c;
    Points[0].Create(-1.0,-1.0, 0.0, c.r(), c.g(), c.b());
    Points[1].Create( 1.0, 1.0, 0.0, c.r(), c.g(), c.b());
    Points[2].Create( 1.0,-1.0, 0.0, c.r(), c.g(), c.b());
    Points[3].Create(-1.0,-1.0, 0.0, c.r(), c.g(), c.b());
    Points[4].Create(-1.0, 1.0, 0.0, c.r(), c.g(), c.b());
    Points[5].Create( 1.0, 1.0, 0.0, c.r(), c.g(), c.b());

    lpDevice->DrawPrimitive(D3DPT_TRIANGLELIST, D3DFVF_XYZ | D3DFVF_DIFFUSE,
        Points, 6, 0);
}

void draw_tri_r() {
    CVertex Points[3];
    dp::Color c=current_c;
    Points[0].Create(-1.0,-1.0, 0.0, c.r(), c.g(), c.b());
    Points[1].Create(-1.0, 1.0, 0.0, c.r(), c.g(), c.b());
    Points[2].Create( 1.0,-1.0, 0.0, c.r(), c.g(), c.b());
    lpDevice->DrawPrimitive(D3DPT_TRIANGLELIST, D3DFVF_XYZ | D3DFVF_DIFFUSE,
        Points, 3, 0);
};
};

char * dp::Primitive::FormNames [] = {
    "", "трик. прав.", "трик. прямок.", "трик. рівноб.", "квадрат",
    "п`ятикутник", "шестикутник", "коло", "еліпс", "прямокутник",
    "паралелепіпед", "трапеція", 0,
};

bool Primitive::show_abc[][3]={
    {1,1,1}, // None,
    {1,0,0}, // Triangle_EL,
    {1,1,0}, // Triangle_R,
    {1,1,0}, // Triangle_ISO, /*3*/
    {1,0,0}, // Quadrangle,
    {1,0,0}, // Pentagon,
    {1,0,0}, // Hexagon, /*6*/
    {1,0,0}, // Circle,
    {1,1,0}, // Ellipse,
    {1,1,0}, // Rectangle, /*9*/
};

```

```

    {1,1,1}, // Parallelepiped,
    {1,1,1}, // Trapezium,
    {1,0,0}, // Octagon
    {1,1,1},
    {1,1,1},
};

double Primitive::default_data[][3]={
    {1,1,1}, // None,
    {1,0,0}, // Triangle_EL,
    {1,0.5,0}, // Triangle_R,
    {1,0.5,0}, // Triangle_ISO, /*3*/
    {1,0,0}, // Quadrangle,
    {1,0,0}, // Pentagon,
    {1,0,0}, // Hexagon, /*6*/
    {1,0,0}, // Circle,
    {1,0.5,0}, // Ellipse,
    {1,0.5,0}, // Rectangle, /*9*/
    {1.1,1,1}, // Parallelepiped,
    {1.1,1,1}, // Trapezium,
    {1,0,0}, // Octagon
    {1,1,1},
    {1,1,1},
};

void Primitive::draw()
{
    const int circle_subdivide=50;
    d3dtranslate(trans.offset_x, trans.offset_y, 0.0);
    d3drotatez(trans.rotation);
    d3dscale(trans.scale_x,trans.scale_y,1.0);

    switch(form) {

        case Triangle_EL:
            d3dscale(a,a,1.0);
            draw_Ngon(3); break;
        case Triangle_R:
            d3dscale(a,b,1.0);
            draw_tri_r(); break;
        case Triangle_ISO:
            d3dscale(a,b,1.0);
            draw_Ngon(3); break;
        /* квадрат */
        case Quadrangle:
            d3dscale(a,a,1.0);
            d3drotatez(45);
            draw_Ngon(4); break;
        case Rectangle:
            d3dscale(a,b,1.0);
            draw_rect(); break;
        case Pentagon:
            d3dscale(a,a,1.0);
            draw_Ngon(5); break;
        case Hexagon:
            d3dscale(a,a,1.0);

```

```

    draw_Ngon(6); break;
case Octagon:
    d3dscale(a,a,1.0);
    draw_Ngon(8); break;
case Circle:
    d3dscale(a,a,1.0);
    draw_Ngon(circle_subdivide); break;
case Ellipse:
    d3dscale(a,b,1.0);
    draw_Ngon(circle_subdivide); break;

case Parallelepiped: {
    double x2=double(a*a-c*c);
    if (x2<0) break;
    double x=sqrt(x2), h=c/2;
    draw_4points(
        b/2.0+x, h,
        b/2.0, -h,
        -x-b/2.0, -h,
        -b/2.0, h );
    break;
}
case Trapezium: {
    double x2=double(a*a-c*c);
    if (x2<0) break;
    double x=sqrt(x2), h=c/2;
    draw_4points(
        b/2.0, h,
        b/2.0+x, -h,
        -x-b/2.0, -h,
        -b/2.0, h );
    break;
}
}
}

```

```

/* END OF NAMESPACE*/

```