

УДК 621.326

Ільчишин Р. – ст. гр. СНм-51

*Тернопільський національний технічний університет імені Івана Пулюя*

## **РОБОТА З ГРАФІКОЮ В ОС IOS**

Науковий керівник: к.т.н., доцент Загородна Н.В.

Ilchyshyn R.

*Ternopil Ivan Pul'uj National Technical University*

## **WORKING WITH GRAPHICS IN OS IOS**

Supervisor: Zahorodna N.

Ключові слова: UIKit, UIView, CoreGraphics, CALayer, OpenGL ES

Keywords: UIKit, UIView, CoreGraphics, CALayer, OpenGL ES

Операційна система iOS надає кілька різних API для роботи з графікою. Можна перелічити 3 основних фреймворки (в порядку зростання абстракції відносно апаратного забезпечення): OpenGL ES, Core Graphics, UIKit.

UIKit надає UIView – клас, що визначає прямокутну область екрану і інтерфейси для рендерингу контенту в цій області. Даний клас є основою для побудови користувацького інтерфейсу. Кожен UIView в основі має CALayer. У свою чергу, CALayer має буфер у вигляді піксельного бітмапу. Саме на цьому бітмапі відбувається рендеринг вмісту UIView.

Для рендерингу вмісту дочірній клас UIView повинен перевантажити метод -drawRect:. В цьому методі виконується вся робота з промальованням графічного наповнення. Для цього в UIKit є набір графічних примітивів: UIBezierPath – для створення полігонів та кривих, UIImage – для промальовки піксельних зображень та UIColor для керування кольором заливки та ліній графічних примітивів.

Для виклику промальовання інтерфейсу використовують метод setNeedsDisplay щоб повідомити систему що даний UIView потрібно перерендерити. Заборонено для виконання перемальовки викликати безпосередньо метод drawRect:.

В основі графічної частини UIKit стоїть Core Graphics – низькорівневий програмний інтерфейс, написаний мовою C. Даний API призначений для створення і промальовання високоефективних графічних примітивів на базі шляхів (тип CGPath), градієнтів (тип CGGradient) та зображень (тип CGImage).

Доступ до промальовування з використанням CoreGraphics можна отримати використавши CALayer (аналоги якого в більшості інших систем називатимуться канвас), який є в кожному UIView. Для цього для проперті layer необхідно встановити делегата, який буде відповідати за наповнення layer-а при кожній промальовці. Делегат повинен перевантажувати метод -drawLayer:inContext:. Саме в цьому методі слід використовувати функції CoreGraphics для створення наповнення CALayer.

Вищеперелічені способи створення графіки надають зручні високорівневий інтерфейси. Але для створення дійсно високопродуктивної графіки, наприклад 3d-ігор чи рендерингу руху мільйонів об'єктів на мобільних пристроях використовують спрощений аналог OpenGL – OpenGL ES. iOS надає можливість оперування функціями OpenGL ES через класи GLKView та CAEAGLLayer.