



THE UNIVERSITY OF HONG KONG

**FACULTY OF ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE**

COMP7506 Smart Phone Apps Development

SHI YUHAN(3035540736)

Electrical and Electronic Engineering (General Stream)

2018.12.01

Category

Category	2
BackGround	3
Game Design.....	4
I) Scenes	4
II) Add camera and random geometry	4
III) Add motion trajectories physical effects	4
IV) Add Render Loop and Remove off-screen child nodes	5
V) Add header display panel.....	5
VI) Touch processing.....	5
VII) Explosive particle effect	6
View and Sound Design	7
I) Main Menu	7
II) DETAIL	7
III)START	8
IV)Sound	8

BackGround

As the final week is coming, everyone feels stressed for the final exam, I hope that I can find a good way to release my pressure. Nothing can do better than a game to help people escape from the real world and devoted in oneself. For that reason I decided to design an application which can help me release stress and relax.

There are hundreds of thousands of games in App Store and most of them are well designed. They provide novel and interesting rules to enable people to achieve certain goals during the game. In most cases we need to study and compare various strategies to achieve the best results. But since we need much time to review the course for the exam, I don't have so much time to study the rules carefully and maybe it can also makes me feel tired.

It's necessary to design an App which don't have so many rules and also can makes people feel pleasure.

Game Design

I) Scenes

Enter the project name Fighter, select Swift language, SceneKit game technology, Universal device type, remove the unit test hook. Clean up unwanted files and set the view type to SCNView in Main.storyboard. Put all the required assets into the resource folder. Add an icon to the game. Find game icons from Resources and drag and drop them into Assets.xcassets.



II) Add camera and random geometry

```
func setupCamera() {  
    cameraNode = SCNNode()  
    cameraNode.camera = SCNCamera()  
    cameraNode.position = SCNVector3(x: 0, y: 5, z: 10)  
    scnScene.rootNode.addChildNode(cameraNode)  
}
```

Create an empty node and assign it to the cameraNode. Then create a new SCNCamera object and assign it to the cameraNode's camera property. After that set the camera position (x:0, y:0, z:10). Finally add cameraNode to the scene as a child node of the scene root node.

```
func spawnShape()
```

- Create a placeholder geometry
- Define a switch statement to handle the shape returned in ShapeType.random().
- Create an SCNBox object and store it in the geometry.
- Create an SCNNode instance named geometryNode. The constructor uses the geometry parameter to automatically create a node and attach the geometry to it.
- Add the node to the root node of the scene.

III) Add motion trajectories physical effects

```
let randomX = Float.random(min: -2, max: 2)  
let randomY = Float.random(min: 10, max: 18)  
let force = SCNVector3(x: randomX, y: randomY, z: 0)  
let position = SCNVector3(x: 0.05, y: 0.05, z: 0.05)  
geometryNode.physicsBody?.applyForce(force, at: position,  
asImpulse: true)
```

- Create two random floating point numbers to represent the x and y components of the force.
- Use these random numbers to create a vector that represents this force.

-
- Create another vector to indicate where the force is applied. This position is deliberately slightly off center so that the object can be rotated.
 - Apply the force to the physical shape of the geometryNode by calling the applyForce(direction: at: asImpulse:) method.

After running, the object appeared in the air and was thrown into the air by the force. After flying, it was finally affected by gravity.

IV) Add Render Loop and Remove off-screen child nodes

In the GameViewController.swift, add the SCNSceneRendererDelegate protocol and implement the protocol method.

```
extension GameViewController: SCNSceneRendererDelegate {
    // 2
    func renderer(_ renderer: SCNSceneRenderer, updateAtTime time: TimeInterval)
```

The spawnShape() method keeps creating new nodes and adding them to the scene, but it's not removed, just dropping out of sight. Although SceneKit has some optimizations that keep the scene running, it doesn't get stuck, but we still need Remove unwanted nodes.

```
if node.presentation.position.y < -2 {
    node.removeFromParentNode()
}
```

V) Add header display panel

Add a header display panel to display the current game state

```
func setupHUD() {
    game.hudNode.position = SCNVector3(x: 0.0, y: 10.0, z: 0.0)
    scnScene.rootNode.addChildNode(game.hudNode)
}
```

```
=====
func updateHUD() {
    let scoreFormatted = String(format: "%0\4d", score)
    labelNode.text = "❤\lives) ⚡\l(scoreFormatted)"
```

VI) Touch processing

- Get the available touch. Here, if the player uses multiple fingers, there will be multiple touches.

-
- Convert from screen coordinates to the coordinates of scnView.
 -
 - hitTest(_: options:) returns an array of SCNHitTestResult objects representing all objects hit by the rays emitted by the user's touch point.
 - Check if the first result is available.
 - Pass the first encountered node to the touch processing method, which can calculate the increase score or reduce the health value.

```
override func touchesBegan(_ touches: Set<UITouch>, with event: UIEvent?)
```

VII) Explosive particle effect

Create a particle effect with the SceneKit Particle System template and name it Expande.scnpx.

View and Sound Design

The App mainly contain three pages.

I) Main Menu

After running the simulator, it shows the launch screen first. It's used to make people feel that the app has started quickly. After a moment, it shows the main menu view.



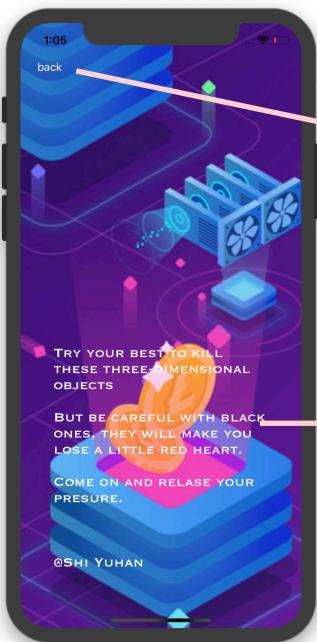
The name of the game is fighter: describe how it looks when playing a game.

Button @START: tap the button to start the game

Button @DETAIL: tap the button to read to rules of the game.

II) DETAIL

First tap the button @DETAIL to read the rules of the game.



Navigation item @back: tap the button to go back to the main menu.

Text View: shows the rules of the game and copyright.

III)START

Back to the main menu and tap button @START to start the game.



Head panel

❤️: the lives you have in the game

💥:the scores you get this time

Three objects you need to fight and they will explode into a pink fireworks when you click on them



IV)Sound

The game contain 5 sound effects.

- ExplodeBad.wav (when you click on the black objects)
- ExplodeGood.wav (when you click on the objects except the black ones)
- GameOver.wav (when the red hearts reduce to zero)
- SpawnGood.wav (when the objects except black ones shows from the bottom)
- SpawnBad.wav (when the black objects shows from the bottom)