COSC2543 Mobile Application programming

# TUTORIAL WEEK 6: GAME API & ANIMATION

Purpose of this tutorial

1. To get you familiar with different layers in Game API
2. Create TiledLayer and Sprite
3. Handle Sprite Collision

In order to adapt with these new topics, you must be master Event Handling in Programming 2 and Thread in Software Architect.

In this scenario, we will create the following:

|  |  |
| --- | --- |
| Grass background using TiledLayer |  |
| Mr. Smith sprite |  |
| The rock sprite |  |
| An explosion sprite |  |

Mr. Smith is standing on the grass while the rock falling down through his head. Once it hits his head, an explosion occurs and he’ll be disappeared

Below are provide classes for your implementation

GameCanvas class for creating layers and handling animation

public class BasicGameCanvas extends GameCanvas implements Runnable {

private Graphics offScreenGraphics;

private Image background, rockSpriteImg, smithSpriteImg, explodeImg;

private LayerManager layerManager;

private TiledLayer grassBackground;

private Sprite rSprite, sSprite, eSprite;

private int x = 0, y = 10;

public BasicGameCanvas(boolean bln) {

super(bln);

}

public void start() throws IOException {

//Create different sprites here

Thread t = new Thread(this);

t.start();

}

public void updateCanvas() {

Graphics g = getOffScreenGraphics();

g.setColor(255, 255, 255);

g.fillRect(0, 0, getWidth(), getHeight());

g.setColor(0, 0, 0);

//Update all layers that we created by Layer Manager

layerManager.paint(g, 0, 0);

// Flush to the display

flushGraphics();

}

//Main thread to run our game

public void run() {

}

//Create Background

public void createBackGround() throws IOException {

}

//Create Rock Sprite and add into layer

public void createRockSprite() throws IOException {

}

//Create Smith Sprite and add into layer

public void createSmithSprite() throws IOException {

}

//Create Explosion Sprite and add into layer

public void createExplosionSprite() throws IOException {

}

// Check Sprite’ collision and handle actions after collision

public void spriteCheckCollide() {

}

}

Main MIDlet

public class GameDemo extends MIDlet {

private BasicGameCanvas cv;

private Display mDisplay;

public GameDemo() throws IOException {

cv = new BasicGameCanvas(false);

}

public void startApp() {

mDisplay = Display.getDisplay(this);

mDisplay.setCurrent(cv);

try {

cv.start();

} catch (IOException ex) {

ex.printStackTrace();

}

}

public void pauseApp() {

}

public void destroyApp(boolean unconditional) {

}

}