Project Documentation – Total Eclipse of the Heart

Team Members: Luke Osborne, Isaac Goldman, Nick Rodriguez, Sami Shahin

The REKTangle

## Documentation

### **User Interface**

The user interface consists of xml layouts that are passed in as content views in the java class functions. The interface consists of a menu screen, instruction screen, playing screen, and game over screen. Each screen has its own layout. The menu has buttons to initiate the game loop or go to the instructions screen. The instructions layout has a transparent button which spans the screen to allow that touching any part of the screen will initiate the gameplay. The game over screen works the same way, initiating a reset to the playing screen to start a new game.

### **Game**

Game serves as the Main Activity of the app. The GamePanel view is launched from Game, which controls the view and music of the app, as well as how the app interacts with the phone. If the home button is pressed on the phone, Game chooses appropriate behavior to pause and restart the app.

|  |  |  |
| --- | --- | --- |
| public class Game extends Activity | | |
| Private member variables | | |
| MediaPlayer | soundtrack | Main game music |
| MediaPlayer | gameover | Gameover sound |

|  |  |  |
| --- | --- | --- |
| Public member functions | | |
| void | onCreate(Bundle savedInstanceState | Sets the screen to full screen and shows the menu screen. Starts the MediaPlayers. |
| void | onClickStartGame(View v) | Starts the game and game music |
| void | onClickInstructions(View v) | Shows the Instruction screen |
| void | onStop() | Stops the sounds if home screen button is pressed. |
| void | onStart() | Restarts the music and game after being minimized. |
| void | playMusic(View view) | Plays main game music |
| void | stopMusic(View view) | Stops game music |
| void | pauseMusic(View view) | Pauses game music |
| void | resumeMusic(View view) | Resumes game music |
| void | restartMusic(View view) | Pauses and rewinds game music |
| void | playDeathSound(View view) | Plays Death sound |
| void | restartDeathSound(View view) | Pauses and rewinds Death sound |

### **GamePanel**

The GamePanel object is the view from which the game loop runs. GamePanel starts MainThread and updates and draws all GameObjects. It controls gameplay, including sound and visual effects, enemy spawning, levels, scoring, and when it is “Game Over”. As a SurfaceView, GamePanel receives touch input from the user in order to control Player.

|  |  |  |
| --- | --- | --- |
| public class MainThread extends Thread | | |
| Constants | | |
| int | WIDTH | Width of the Background, used for scaling |
| int | HEIGHT | Height of the Background, used for scaling |
| int | MAX\_ENEMIES | Maximum number of enemies allowed active at one time. Used for game control and memory constraint. |
| int | LEVEL\_SCORE | The score needed to proceed to the next level of gameplay. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Private member variables | | | | |
| MainThread | | thread | The Thread that controls the framerate and processing speed | |
| Background | | bg | Background for when game is being played | |
| Background | | gameOverBg | Background for Game Over screen | |
| Player | | player | The hero of the game | |
| Paint | | scorePaint | Used for formatting and displaying text on the screen | |
| Arrow | | arrow | Used for drawing direction arrow | |
| ArrayList<Enemy> | | enemies | All enemies in play | |
| long | | enemiesStartTime | Resets whenever an enemy is spawned, used for controlling timing of Enemy spawns | |
| long | | waveStartTime | Used for controlling waves of Enemies | |
| Random | | rand | Used for random generation of Enemy starting locations | |
| char | | side | Used to control which side the Enemies come from | |
| int | | enemyV | The speed of the enemies | |
| Boolean | | gameOver | Used to control display of game over screen | |
| Boolean | | levelScreen | Used to control display of level screen/text | |
| int | | level | Current level of gameplay | |
| Public member functions | | | | |
|  | GamePanel(Context context) | | | Constructor, takes Activity as context |
| void | surfaceDestroyed(SurfaceHolder holder) | | | Joins the Thread and stops music before destruction of surface |
| void | surfaceCreated(SurfaceHolder holder) | | | Initializes GameObjects and Paint, start Times, and music |
| Boolean | onTouchEvent(MotionEvent event) | | | Handles user input to control Player and game |
| void | update() | | | Updates all GameObjects and score |
| void | draw(Canvas canvas) | | | Draws all Objects on screen |
| void | spawn\_top() | | | Spawns an Enemy from the top of the screen |
| void | spawn\_bottom() | | | Spawns an Enemy from the bottom of the screen |
| void | spawn\_left() | | | Spawns an Enemy from the left of the screen |
| void | spawn\_right() | | | Spawns an Enemy from the right of the screen |
| Boolean | detect\_collision(GameObject g1, GameObject g2) | | | Returns true if the rectangles of the GameObjects intersect one another |
| void | choose\_side() | | | Switches between sides in a controlled manner |
| void | choose\_spawn() | | | Chooses which spawn method to call based on the side |

### **MainThread**

MainThread extends Thread (a native Android class) and controls how fast the game runs and the framerate. It is used to ensure clean, smooth graphics and that the processor of the phone isn’t taxed by unnecessary computations.

|  |  |  |
| --- | --- | --- |
| public class MainThread extends Thread | | |
| Private member variables | | |
| int | FPS | Frames Per Second, controllable by programmer |
| double | averageFPS | The average FPS, may be less than FPS if the machine runs too slowly |
| SurfaceHolder | surfaceHolder | Holds the surface, which controls display |
| GamePanel | GamePanel | The main GamePanel, used to control when update and draw methods are called |
| Boolean | running | true if the game is running |
| Canvas | canvas | The canvas is used to draw objects on screen |

|  |  |  |
| --- | --- | --- |
| Public member functions | | |
|  | MainThread(SurfaceHolder surfaceHolder, GamePanel gamePanel) | Constructor |
| void | run() | Runs the main game loop |
| void | setRunning(Boolean b) | Sets the running Boolean variable |

### **GameObject**

GameObject is an abstract class that contains basic information about the objects in the game that need to be displayed and interact with other objects.

|  |  |  |
| --- | --- | --- |
| public abstract class GameObject | | |
| Protected Member Variables | | |
| int | x | x-coordinate of the object’s center in pixels |
| int | y | y-coordinate of the object’s center in pixels |
| int | width | width of the object in pixels |
| int | height | height of the object in pixels |
| Public member functions | | |
| void | setX(int x) | Setter |
| void | setY(int y) | Setter |
| int | getX() | Getter |
| int | getY() | Getter |
| int | getHeight() | Getter |
| int | getWidth | Getter |
| Rect | getRectangle() | Gets the Rectangle of the Object, used for a “hitbox” to detect collisions |

### **Player**

Player is the player of the game, who must dodge Enemies.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| public class Player extends GameObject | | | | |
| Private Member Variables | | | | |
| Bitmap | | sprites | | A spritesheet used for animating the Player. |
| Animation | | animation | | Used to cycle through the different player frames to animate Player |
| int | | score | | Score of Player |
| double | | dx | | for Player movement |
| double | | dy | | y for Player movement |
| double | | destX | | X coordinate Player is moving towards |
| double | | destY | | Y coordinate Player is moving towards |
| Boolean | | moving | | true if the player is in motion |
| double | | v | | Speed of player |
| Boolean | | playing | | true if the player is playing |
| long | | startTime | | starts when Player starts playing |
| Public Member Functions | | | | |
|  | Player(Bitmap b, int x, int y, int w, int h, int nFrames) | | Constructor:  b: spritesheet of Player (png Bitmap)  x,y: Initial x and y position  w, h: width and height  nFrames: num. frames the spritesheet contains | |
| void | update() | | Updates player position and states | |
| int | getScore() | | Returns the score of the player | |
| void | setPlaying(Boolean b) | | Setter | |
| void | resetScore() | | Resets the score to 0 | |
| Boolean | getMoving() | | Returns true if the player is moving | |
| void | setDestX(double newDestX) | | Setter | |
| void | setDestY(double newDestY) | | Setter | |
| void | setDx(double eventX) | | Setter | |
| void | setDy(double eventY) | | Setter | |
| void | draw(Canvas canvas) | | Draws the player on the canvas | |
| void | reset\_player() | | Resets the Player | |

### **Enemy**

Enemy extends GameObject. The enemy class contains information and functions necessary to draw the Enemy class and control Enemy movement and spawning. Enemies must be able to handle coming from any direction on the screen.

|  |  |  |
| --- | --- | --- |
| public class Enemy extends GameObject | | |
| Protected Member Variables | | |
| Animation | animation | Used to cycle through the different player frames to animate Enemy |
| Bitmap | sprites | A spritesheet used for animating the Enemy. |
| int | score | The score of the Player, used to control speed |
| int | v | Speed of the Enemy |
| Random | rand | Used to generate random positions |
| long | startTime | When the enemy spawned |
| char | side | Which side of the screen the Enemy should spawn, used to control animation and speed |

|  |  |  |
| --- | --- | --- |
| Public Member Functions | | |
|  | Enemy(Bitmap b, int w, int h, char side, int s, int x, int y, int v, int nFrames) | Constructor:  b: spritesheet for Enemy  w: width  h: height  side: Which side (left = ‘l’, right = ‘r’, top = ‘t’, bottom = ‘b’)  s: score  x: initial x position  y: initial y position  nFrames: how many frames the spritesheet contains |
| float | sin\_velocity(int velocity) | Used to generate a sine-wave enemy |
| void | update() | Update Enemy position and state |
| void | draw(Canvas canvas) | Draw enemy on the Canvas |
| int | getHeight() | Getter |
| int | getWidth() | Getter |

**Problems, bugs, areas for improvement:**

High score persists while the app is running, but is not restored if the app is exited and resumed. Minimizing and then resuming the game causes a crash for this reason. Had time allowed, we wanted to implement a “Boss” enemy level.

## Resources Used

*paymon wang-lotfi – YouTube Channel Videos*

<https://www.youtube.com/channel/UCKkABMS8IVJlu0G4ipPyZaA>

This basic 2D game creation tutorial. We built his game first to learn, then adapted his class structure for our own game.

*Udacity – Developing Android Apps*

This online course offered directly from Google served as an introduction to Android Studio.