

Game Programming Project 1

Dinosaur V/S Zombie

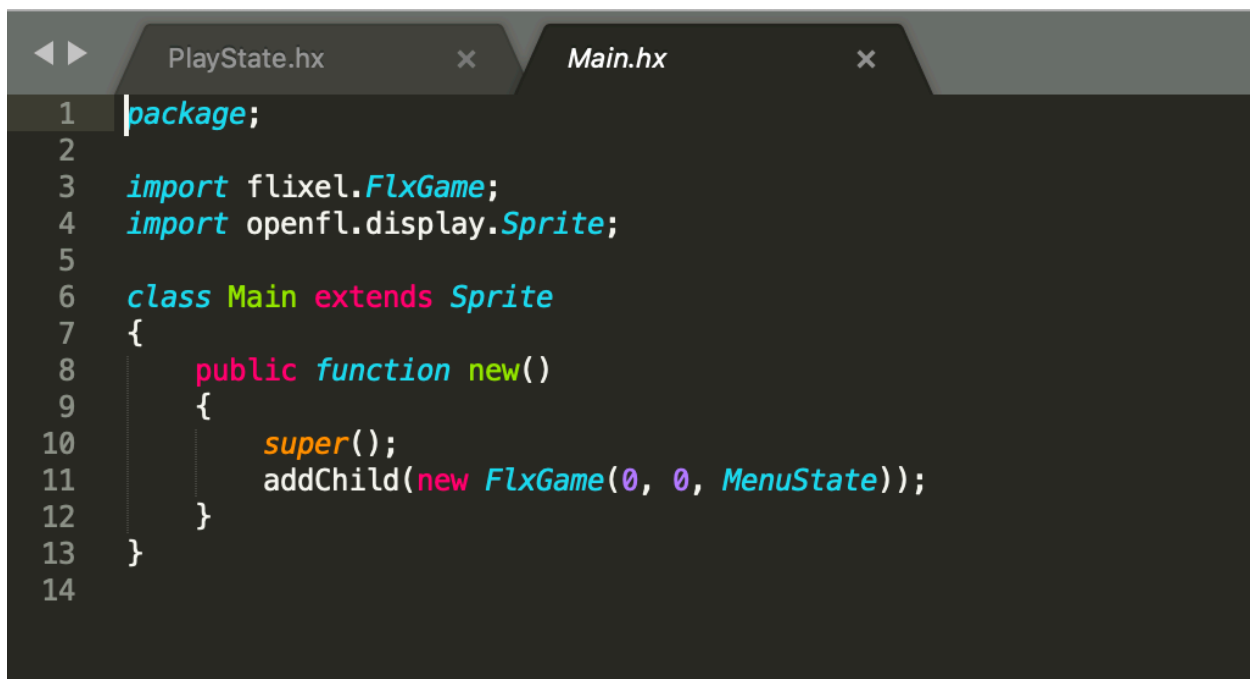
Who Will Conquer the World?

Rules of the game:

1. It is a two player game. Player 1 is Dinosaur and Player 2 is Zombie
2. It is a Shooter Game, where Dinosaur and Zombie are given unlimited bullets and to win the game one has to shoot until the other gets killed
3. When one bullet is hit the players health decreases by 20
4. The controls of the game are WASD for moving and SPACE for shooting for the dinosaur and UP, DOWN, LEFT, RIGHT and ENTER for shooting for zombie
5. The players are allowed to move around the tile map but they are not allowed to cross the walls
6. Walls sometimes play as a protective cover for the player so that he cannot get the bullets touched
7. Once, any of the players health becomes zero the opposite player wins the game

Code:

Main.hx



```
1 package;
2
3 import flixel.FlxGame;
4 import openfl.display.Sprite;
5
6 class Main extends Sprite
7 {
8     public function new()
9     {
10         super();
11         addChild(new FlxGame(0, 0, MenuState));
12     }
13 }
14
```

The main function of Main.hx is to transfer to the MenuState as shown in line 11

MenuState.hx

```

1  package;
2
3  import flixel.FlxState;
4  import flixel.FlxG;
5  import flixel.util.*;
6  import flixel.ui.*;
7  import flixel.text.FlxText;
8
9  class MenuState extends FlxState
10 {
11     var gameTitle:flixel.text.FlxText;
12     var btnPlay:flixel.ui.FlxButton;
13     var developer:FlxText;
14
15     function clickPlay()
16     {
17         FlxG.switchState(new PlayState());
18     }
19
20     override public function create():Void
21     {
22         FlxG.cameras.flash(FlxColor.BLACK, 3);
23         gameTitle = new FlxText(0,FlxG.height/4, FlxG.width, "Dinosaur V/S Zombie \n Who will conquer the World?");
24         gameTitle.setFormat(null,30,FlxColor.WHITE, "center");
25         add(gameTitle);
26         FlxG.sound.play("assets/sounds/heartbeat.wav",0.15,true);
27         btnPlay= new FlxButton (280, 220, "Play", clickPlay);
28         add(btnPlay);
29
30         developer = new FlxText(280,300, 100, "Developed by Tejas Wadiwala \n @tejaswadiwala");
31         add(developer);
32
33         super.create();
34     }
35
36     override public function update(elapsed:Float):Void
37     {
38         super.update(elapsed);
39     }
40 }
41

```

This file shows the menu screen of the game as shown it will display the title of the game and will show a button Play to start the game

PlayState.hx

```

42
43     override public function create():Void
44     {
45
46         FlxG.sound.play("assets/sounds/heartbeat.wav",0.15,true);
47         healthPlayer=100;
48         healthPlayer2=100;
49
50         healthText=new FlxText(0,0, FlxG.width, "Dinosaur: "+ healthPlayer);
51         healthText.setFormat(null,15, FlxColor.LIME,"left");
52
53
54         healthText2=new FlxText(0,0, FlxG.width, "Zombie: "+ healthPlayer2);
55         healthText2.setFormat(null,15, FlxColor.LIME,"right");
56
57         gameOver=new FlxText(0,FlxG.height/2-50, FlxG.width, "Thank you for playing!");
58         gameOver.setFormat(null,50, FlxColor.GREEN,"center");
59         gameOver.visible = false;
60

```

The above code shows the sound which I have added in the background of the game along with text placement and initializations

```

81     map = new FlxOgmoLoader("assets/ogmo/Level1.ogm");
82     mWalls = map.loadTilemap("assets/images/walls.png", 32, 32, "wall");
83     mWalls.follow();
84     mWalls.setTileProperties(1, FlxObject.NONE);
85     mWalls.setTileProperties(2, FlxObject.ANY);
86     add(mWalls);
87

```

The above code shows that we have used ogmoloader to load the tiles for the game. The tiles were created using the ogmo editor for windows. The add(mWalls) line creates walls through which player cannot pass.

```

138     override public function update(elapsed:Float):Void
139     {
140         super.update(elapsed);
141         FlxG.collide(player,mWalls);
142         FlxG.collide(player,leftWall);
143         FlxG.collide(player,rightWall);
144         FlxG.collide(player,topWall);
145         FlxG.collide(player,bottomWall);
146
147         FlxG.collide(bulletPlayer,mWalls, BWall);
148         FlxG.collide(bulletPlayer,leftWall, BWall);
149         FlxG.collide(bulletPlayer,rightWall, BWall);
150         FlxG.collide(bulletPlayer,topWall, BWall);
151         FlxG.collide(bulletPlayer,bottomWall, BWall);
152
153         FlxG.collide(player2,mWalls);
154         FlxG.collide(player2,leftWall);
155         FlxG.collide(player2,rightWall);
156         FlxG.collide(player2,topWall);
157         FlxG.collide(player2,bottomWall);
158
159         FlxG.collide(bulletPlayer2,mWalls, BWall2);
160         FlxG.collide(bulletPlayer2,leftWall, BWall2);
161         FlxG.collide(bulletPlayer2,rightWall, BWall2);
162         FlxG.collide(bulletPlayer2,topWall, BWall2);
163         FlxG.collide(bulletPlayer2,bottomWall, BWall2);
164
165         FlxG.collide(bulletPlayer, player2, Hit);
166         FlxG.collide(bulletPlayer2, player, Hit2);
167

```

The above code shows that we have added physics using the collide function of HaxeFlixel and the third parameter in the collide function is the function that will be redirected to when the first two parameters collide.

```

201
202     if(FlxG.keys.justReleased.ENTER && canShoot2 == true){
203         canShoot2=false;
204         bulletPlayer2.exists=true;
205         bulletPlayer2.visible=true;
206         bulletPlayer2.x = player2.x+player2.width/2;
207         bulletPlayer2.y = player2.y+player2.height/2;
208         var x = player2.x-player.x;
209         if(x<0){
210
211             bulletPlayer2.velocity.x= 1000;
212             FlxG.sound.play("assets/sounds/Gun+Silencer.wav",0.15,false);
213             x=1;
214         }
215         else{
216             bulletPlayer2.velocity.x= -1000;
217             FlxG.sound.play("assets/sounds/Gun+Silencer.wav",0.15,false);
218             x=-1;
219         }
220     }
221
222     if(FlxG.keys.justReleased.SPACE && canShoot == true){
223         canShoot=false;
224
225         bulletPlayer.exists=true;
226         bulletPlayer.visible=true;
227         bulletPlayer.x = player.x+player.width/2;
228         bulletPlayer.y = player.y+player.height/2;
229         var x = player2.x-player.x;
230         if(x>0){
231
232             bulletPlayer.velocity.x= 1000;
233             FlxG.sound.play("assets/sounds/Gun+Silencer.wav",0.15,false);
234             x=-1;
235         }
236         else{
237             bulletPlayer.velocity.x= -1000;
238             FlxG.sound.play("assets/sounds/Gun+Silencer.wav",0.15,false);
239             x=1;
240         }
241     }
242

```

The above code shows when player presses SPACE or ENTER then a bullet is shot and when the bullet is shot a sound is played as seen in lines 212, 217, 233, 238.

It will also check the direction of the opposite player and shoot the bullet according to it, so in the lines 210 to 219 it will check by the position of x by subtracting the by both players x coordinate and then go in the if loop if the condition is satisfied.

```

274     }
275
276     function BWall(Bullet:FlxObject, Wall:FlxObject):Void{
277         Bullet.exists=false;
278
279         canShoot=true;
280     }
281
282     function BWall2(Bullet:FlxObject, Wall:FlxObject):Void{
283         Bullet.exists=false;
284         canShoot2=true;
285     }
286
287
288     function Hit(Bullet:FlxObject, Player:FlxObject):Void{
289         Bullet.exists=false;
290         canShoot=true;
291         healthPlayer2-=20;
292         healthText2.text="Zombie: " + healthPlayer2;
293
294     }
295     function Hit2(Bullet:FlxObject, Player:FlxObject):Void{
296         Bullet.exists=false;
297         canShoot2=true;
298         healthPlayer-=20;
299         healthText.text="Dinosaur: " + healthPlayer;
300
301     }
302 }
303
304

```

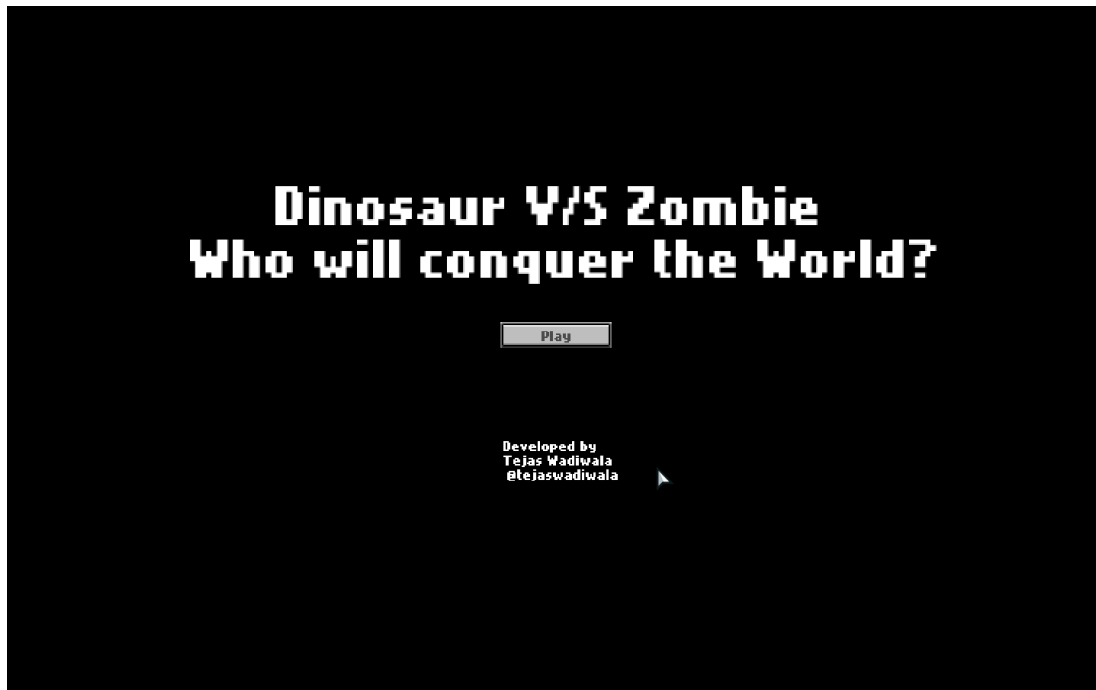
The above code shows the function which would be called when the first two parameters written in the collide function collide.

```
1 package;
2
3 import flixel.util.FlxColor;
4 import flixel.FlxSprite;
5 import flixel.util.*;
6 import flixel.util.FlxSpriteUtil;
7 import flixel.FlxG;
8 using flixel.util.FlxSpriteUtil;
9 import openfl.display.Graphics;
10
11 class Wall extends FlxSprite {
12
13     public function new (X, Y,A,B)
14     {
15         super(X,Y);
16         makeGraphic(A,B,FlxColor.TRANSPARENT);
17         immovable=true;
18
19     }
20
21
22
23     override public function update (elapsed:Float)
24     {
25         super.update(elapsed);
26     }
27
28 }
```

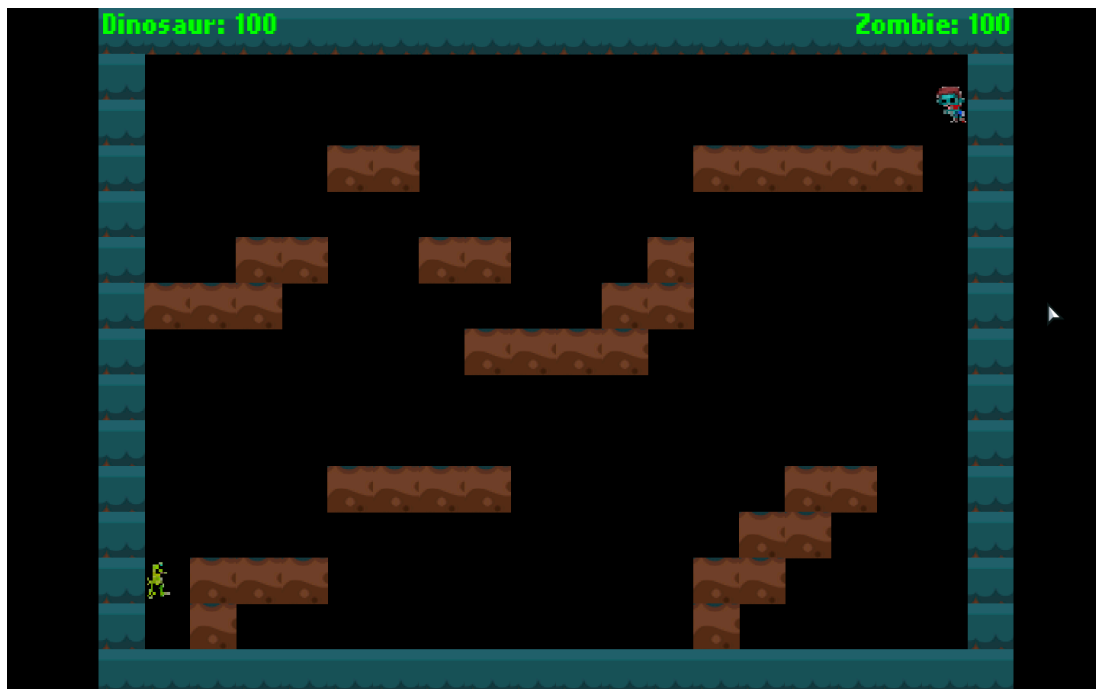
Wall.hx

The above code shows the wall.hx file which is used for creating the four walls right, left, top and bottom so that the player sprite is not able to leave the screen and I have given it the color to transparent because I already have the bricks created.

Screenshots of the Game:



The Menu Screen with one button



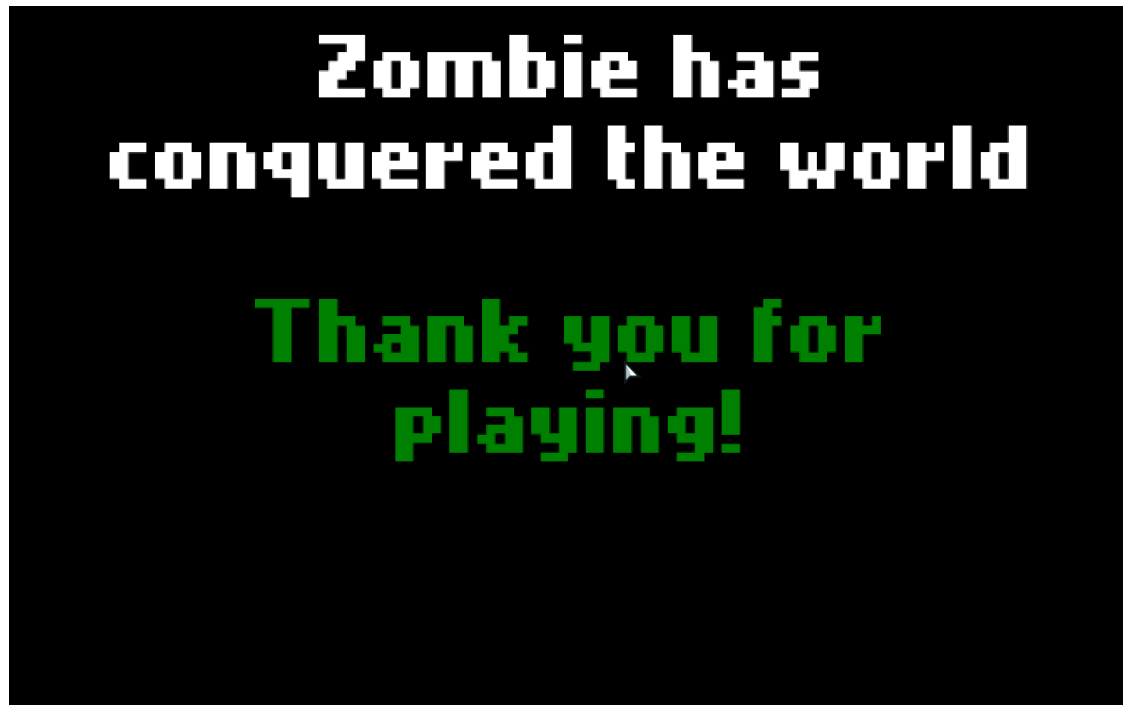
The initial Screen when both Zombie and Dinosaur have 100 Health



When the Dinosaur has 20 health and Zombie has 100 health



When the Dinosaur wins the game



When the Zombie wins the game

How to Play:

1. Extract the zip file
2. Open TheGame sublime project
3. Press Ctrl + Shift + B to build the game
4. Press Ctrl + ENTER to run the game
5. Play

Submitted By- Tejas Wadiwala
0889445