Microgame #3 Shmup

Name: Alan L. Perez

id: 004862867

Unity File: https://github.com/alanlperez/HW3Shmup

1. Create a new project, Shmup
2. Create folders under Assets: Scripts, Sprites, Prefabs, Animations
3. Download Sprites.zip from Canvas. Unzip, and drag sprites into Sprites folder.
4. Click tiles, and change Sprite mode to multiple.Graphical user interface, application

   Description automatically generated
5. Grab a small part of a specific tile from sprite editor and create Tile\_grass. A screenshot of a game

   Description automatically generated with medium confidence

A picture containing graphical user interface

Description automatically generated

1. Create a 2D GameObject, go into sprite shape, then into tile map, and click rectangle. Rename tile map as “Background”.



1. Go into Tile Palette, then create a new palette named BackgroundPalette, and save it under the tiles folder, and drag tile grass into BackgroundPalette.Graphical user interface

   Description automatically generated

Chart, histogram

Description automatically generated

1. Since our tile is 32x32, we need to change the Pixels Per Unit to 32. Then hit apply. Graphical user interface, application

   Description automatically generated
2. Go back to tile palette and drag our green tile into our grid background.A screenshot of a computer

   Description automatically generated with medium confidence
3. Change the Order in Layer to -1 In Background Inspector since we want it to be our background layer.

Graphical user interface, application

Description automatically generated

1. Player is going tpp be order in layer 0 to be above the background.A screenshot of a computer

   Description automatically generated with medium confidence
2. Add Box Collider2D and RigidBody2D to player. Set Linear Drag and Mass to 5. Set gravity as 0, and freeze rotation on the Z-axis. Graphical user interface

   Description automatically generated
3. Create PlayerController.cs and attach it to the Player game object.Graphical user interface, text, application, email

   Description automatically generated
4. Create Different Sections in the PlayerController.cs for the variables with headers.

Text

Description automatically generated

1. Take input from the user to know where the player wants to move within the x and y -axis. Add “playerRigidbody” for the player to move smoothly.Text

   Description automatically generated
2. Set speed for player in “Player” game object under the “Player Controller (Script)” to 2000.

Graphical user interface, application

Description automatically generated

1. Create two game objects under the player “BulletSpawnPosL” and “BulletSpanPosR”. From inspector create a label, blue dot for left and yellow dot for right. Place them on the wings of the player.

Graphical user interface, text

Description automatically generated

1. Go back into “PlayerController.cs” and create a section for the “shooting” variables.

Text

Description automatically generated

1. Create Bullet game object by taking the sprite “Bullet” from the sprite folder and placing it into the “Hierarchy”. Change the size of the bullet, Edit the collider to the same size of the bullet, and add a Rigidbody2D, and freeze rotation. Then create script file and attach it to bullet. Then Save Bullet as Prefab.

Graphical user interface

Description automatically generated

1. In the “PlayerController.cs” under the update function call the shoot function if a the player presses the spacebar. Then create the shoot function so that the player can shoot two bullets. One from the left position and one from the right position.

Text

Description automatically generated with low confidence

Text

Description automatically generated

1. Drag the bullet prefab to the variable “Bullet” in the Player game object. Enter 2 for the “Bullet Spawn Position in order to create a two element array and drag “BulletSpawnPosL” for the first element and “BulletSpawnPosR” for the next element.Graphical user interface, application

   Description automatically generated
2. Create a cool down time in “PlayerController.cs” for the player between shots.



Text

Description automatically generated

Text

Description automatically generatedGraphical user interface, text

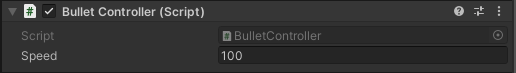
Description automatically generated

1. Initialize “Time Between Shots in Inspector.



1. Edit “BulletController.cs” to add force to the bullet. Text

   Description automatically generated
2. Set speed of Bullet prefab to 100



1. Add two enemies to the game and drag mig15 to the Hierarchy panel. Change the name to Enemy1, change its size, and change the rotation Z to 180. Add Box Collider 2D and adjust size. Then add RigidBody2D, set Mass and Linear Drag to 5. Freeze Rotation of Z. Add EnemyController.cs to Enemy1 game object. Make Enemy1 into a prefab.

Graphical user interface

Description automatically generated

1. Drag Enemy1 prefab into Hierarchy and rename it as “Enemy2”. Right click, go under prefab, and select unpack completely. This will make it a normal game object. Change image from mig15 to mig21, and change its size. Then make it into a prefab.

Graphical user interface

Description automatically generated

1. Go into EnemyController.cs and make the enemy move toward the player.Text

   Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

1. Set the x and y speed for prefab Enemy1 and Enemy2

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

1. Drag Bullet prefab into Hierarchy and rename it “EnemyBullet”. Right click and go under prefab and select unpack completely. Then drag the “TopGunBullet” and place into game object sprite. Then remove BulletController.cs and replace it with EnemyBulletController.cs. Lastly, save it as a prefab.Graphical user interface

   Description automatically generated
2. Open EnemyController.cs. We don’t want the enemy to attack too often so we need to set a cool down time. Also, give it the game logic for how it can attack.

Text

Description automatically generated

Graphical user interface, application

Description automatically generated

1. Set the attack Low and Highs for both Enemy1 and Enemy2. Initialize Bullet with prefab for both.





1. Initialize the Bullet from prefab to inspector for Enemy 1 and 2.

Text

Description automatically generated

Text

Description automatically generated

1. We go into EnemyBulletController.cs and copy and paste over what we had for the BulletController.cs. However, you must add a minus for Vector2.up in OnEnable because it is going in a negative direction in the y-axis.

Text

Description automatically generated

1. Set the speed for the Enemy Bullet in the prefab.
2. Create game object named GameController, then create a scriptfile GameController.cs and attach it to GameController game object.

Graphical user interface, text, application

Description automatically generated

1. In GameController.cs we make the rules for Spawning enemies. The time between spawns, the cool down for spawning, and where they will spawn. The location of the spawning is down outside the view of the camera.

Text

Description automatically generated

Graphical user interface, text, application

Description automatically generated with medium confidenceText

Description automatically generated

1. We go into the GameController game object and under the Game Controller (Script) we initialize Enemies as 2 in order to create a 2 element array, and we insert the Enemy1 and Enemy2 prefrabs in that order. A screenshot of a computer

   Description automatically generated with medium confidence
2. Go back into PlayerController.cs in order to write the rules for player heath and how it takes damage.

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

1. Go into the EnemyBulletController.cs so that if the enemy bullet collides with the player, the player will then take damage. Call disable to destroy the bullet when it makes contact with the player.



Text

Description automatically generated

1. Tag the Player game object as “Player”.



1. Mark the Enemy Bullet prefab as a TriggerCollider.

Graphical user interface, application

Description automatically generated

1. Go into EnemyController.cs and define enemy health.

Text

Description automatically generated

A picture containing text

Description automatically generated





1. Then go into the BulletController.cs and create the function OnTriggerEnter2D. for enemy to take damage.



Graphical user interface, text

Description automatically generated

1. Tag Enemy1 and Enemy2 as “Enemy”



1. In the Player Bullet set the Trigger Collider.

Graphical user interface, text, application, email

Description automatically generated

1. Set the maximum health for both Enemy1 and Enemy2.

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

1. To create the healthbar we insert a Canvas game object and insert, and empty game object named HealthParent.



1. Attach Rect Transform to HealthParent. Then change its position to top left. Then attach Horizontal Layout Group, make spacing 5.

A screenshot of a computer

Description automatically generated with medium confidence

1. The attach an Image object under HealthParent.

Graphical user interface, application

Description automatically generated

1. In the Image object attach the healthHeart sprite into Source Image. The we set the size of the image.

Graphical user interface, application

Description automatically generated

1. We will use script file to create multiple heart to represent health of player. We will do this in PlayerController.cs. We will also create function “Remove heart” to take away hearts in the event damage is taken.



Text

Description automatically generated with medium confidence

Text

Description automatically generated

Text

Description automatically generated

1. Go into the Player game object and go to the Player Controller (Script) and drag the HealthParent and Image game objects into Health Image and Health Parent.



1. Next we want to display the score of the player. A player will gain points when destroying an enemy, but they will lose points if they allow any enemy to pass them. Go under UI and select text. Rename text to ScoreText and place it under Canvas.

Graphical user interface, text

Description automatically generated with medium confidence

1. In ScoreText under Rect Transform we pick top center. Then we change the position of X and Y. Then we change width and height. Then we set Font size, and bold it. In Text we put “Score: 0” and for both Vertical and Hortizontla Overflow we pick Overflow.

A screenshot of a computer

Description automatically generated with medium confidence

1. Then we go into GameController.cs to allow the score text to change in game.





Graphical user interface, text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

1. We then go to EnemyController.cs to initialize the game controller. We do this in order to be able to use AddScore().



Graphical user interface, text, application

Description automatically generated

Graphical user interface, text

Description automatically generated

1. Then in the GameController game object we drag the ScoreText into the Score Text variable under Game Controller (Script).

Graphical user interface, application

Description automatically generated

1. Then in the enemy prefabs we set the damage “Amount[s]” they take. As well as the score the player will receive.

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

1. If the player moves outside the bounds of the camera the player will lose score.



Text

Description automatically generated

Text

Description automatically generated

1. If the player lives is less than or equal to zero the game pauses and the game over screen is visible. This will pause the game. Ad UI text to Canvas and name it GameOverUI. Under the GameOverUI add Text under it.

Graphical user interface, text, application

Description automatically generated

1. Go into GameOverUI game object change size and set the color of the game over screen.

Graphical user interface

Description automatically generated

1. Then go under Text and change size, position, font size. Set Horizontal/Veritcal Overflow to Overflow.

Graphical user interface

Description automatically generated

1. Go into GameController.cs we set the GameOverUI. When the player presses any key it will restart the game. This can be seen in Restart()





Graphical user interface, text

Description automatically generated

Graphical user interface, text, application

Description automatically generatedText

Description automatically generated

1. We then go into PlayerController.cs and create the GameOver(). This will set the parameters for the GameOver to be true.

Graphical user interface, text

Description automatically generated

1. We disable to GameOverUI game object so that we can only see it when it is activated.



1. Next take the explosion sprite and change sprite mode to multiple

Graphical user interface, application

Description automatically generated

1. Slice the sprite manually and make sure each sprite is the same size. Then press apply. Select all the slices created and drag it to the hierarchy panel. Then we save it into the animation folder. Then we change the size in the game object.

Graphical user interface, application

Description automatically generatedGraphical user interface

Description automatically generated

1. Create ExplosionController.cs and attach it to the Explosion game object.

Graphical user interface, text, application

Description automatically generated

1. Go into the ExplosionController.cs to control the explosion animation.

Text

Description automatically generated

1. Go to the animation folder and drag the explosion animation to the clip under Explosion Controller (Script) in the Explosion game object.

A picture containing graphical user interface

Description automatically generated

Graphical user interface, text, application

Description automatically generated

1. Go into the EnemyController.cs and go to Die() and call the explosion when an enemy dies.

A picture containing graphical user interface

Description automatically generated

1. Save the explosion as a prefab and drag it into the hierarchy panel. Then attach it to Enemy1 and Enemy2 prefab.

Graphical user interface

Description automatically generated

Graphical user interface

Description automatically generated

1. Then we are going to create the flash animation when the player shoots the bullet. Insert the sprites from the flash folder into the Sprites folder.

A screenshot of a computer

Description automatically generated with medium confidence

1. Select them all and put them into the hierarchy and save them into the animation folder. Make it into a prefab.

A picture containing text

Description automatically generated

A screenshot of a cell phone

Description automatically generated with medium confidence

1. Lock the inspector and attach the animation to the clip.

Graphical user interface, text, application

Description automatically generated

1. Go the PlayerController.cs and go to Shoot(). This will allow us to see the flash animation when the player shoots.



Graphical user interface, text, application

Description automatically generated

1. Then drag the flash prefab into the Player game object.

Graphical user interface, application

Description automatically generated

1. In PlayerController.cs we will have a cool down time for when a player can take damage.

Graphical user interface, text

Description automatically generated

Graphical user interface, text

Description automatically generated  
Graphical user interface, text, application, chat or text message

Description automatically generated

1. Then we implement that if the enemy collides with the player, the enemy will get destroyed and the player will take damage. Go into the PlayerController.cs and create the function OnCollisionEnter2D().

Text

Description automatically generated