

ICS 168-Single Player Game Architecture

Team Members:

Jasmine Hunter
Arielle Chongco
Matthew Ruiz
Luke Lohden

Project Details:

Game Chosen: Bombeman
Platform Chosen: Unity (C#)

Source Code Architecture:

Animations:

- BombAnimation - Bomb fusing down animation
- BombermanAnimation - Player walking animation
- ExplosionAnimation - Bomb explosion animation
- WormAnimaton - Worm Movement animation

Prefabs (Game Objects):

- Bomb
- Explosion

Scripts:

- Bomb.cs
 - Description: Defines Bomb object behavior
 - Functionality:
 - OnDestroy(): Instantiates an Explosion object at the Bomb's location
- BombDrop.cs
 - Description: Handles when to drop a Bomb
 - Functionality:
 - Update(): If the player presses space, instantiate a Bomb
- DestroyAfter.cs
 - Description: Defines Bomb to Explosion transition
 - Functionality:
 - Start(): Calls Destroy()) on bomb With 3 second delay.
- Explosion.cs
 - Description: Defines Explosion behavior
 - Functionality:
 - OnTriggerEnter():
 - If collided with player: Player takes damage
 - If collided with an enemy: Player scores, monster dies

- GameController.cs
 - Description: Handles gamestate transitions
 - Functionality:
 - Fixed Update(): If all enemies killed, display winning interface
 - Reset(): Loads a fresh gamestate
 - PlayerDied(): Displays game over interface
 - PlayerScored(): Increments player Score
- Move.cs
 - Description: Defines player movement and animates for movement
 - Functionality:
 - FixedUpdate(): Define movement variables (position and velocity) and set animation parameters.
- PlayerController.cs
 - Description: Handles updates to character state
 - Functionality:
 - Awake(): Calls respawn and binds the GameController
 - FixedUpdate(): Update player movement and movement animations
 - Respawn(): Move to respawn location
 - Damage(): Calls respond and decrements lives. Kills player is lives == 0
- Worm.cs
 - Description: Defines Worm object behavior
 - Functionality:
 - Start(): Changes direction every .5 seconds
 - randDir(): Returns a random number representing a direction.
 - isValidDir(): Checks to see if the worm can move in the direction given.
 - changeDir(): Sets worm movement and animation to reflect new direction.