

# Project 1: Pseudocode

## **Player Movement:**

### Start:

- Initialise variables

### Update:

- If player is not alive, exit
- Call Run()
- Call FlipSprite()
- Call Die()

### OnMove:

- If the player is not alive, exit
- Get the input from the player (joystick/keys) and update moveInput

### OnFire:

- If player is not alive, exit
- Instantiate the weapon prefab at the firePoint

### OnJump:

- If player is not alive, exit
- If player is not touching the platform, exit
- If the jump button /key is pressed:
  - Apply jump speed to the player's Rigidbody

### Run:

- Calculate the player velocity
- Set player's Rigidbody velocity to playerVelocity
- Update the Animator's 'isRunning' parameter

### FlipSprite:

- Check if the player is moving
- If yes, access the local scale and flip the player's sprite direction

### Die:

- If player touches enemy or hazard:
  - Set isAlive bool to false
  - Trigger dying animation
  - Apply deathKick to player's Rigidbody
  - Disable the player controls
  - Call ProcessPlayerDeath() from GameSession

ProcessPlayerDeath:

If playerLives > 1:

TakeLife()

Else:

ResetGameSession()

TakeLife:

Decrease playerLives by 1

Reload current scene

Update UI with remaining lives

ResetGameSession:

Load the first scene

Destroy GameSession object

### **Enemy Movement:**

Start:

Initialise variables

Set initial velocity

Update:

Set velocity to move the enemy

Call FlipEnemyFacing

OnTriggerExit2D:

Change the movement direction of the enemy

Call FlipEnemyFacing method

FlipEnemyFacing:

Flip enemy's sprite direction

Die:

Trigger death animation

Start DestroyAfterAnimation coroutine

DestroyAfterAnimation:

Wait for 0.1 seconds

Destroy the enemy object

## **Enemy Wave Spawner**

Start:

- Start the SpawnWaves coroutine

SpawnWaves:

While true:

- Wait for a set time before spawning the next wave

- Call SpawnWave

SpawnWave:

For each enemy to be spawned:

- Choose a random spawn point

- Instantiate the enemyPrefab at the spawn point

StopSpawning:

- Stop all spawning coroutines

## **Weapon**

Start:

- Initialise variables

- Calculate the xSpeed (speed the weapon's ammo fires horizontally)

Update:

- Set weapon's Rigidbody velocity in x-direction

OnTriggerEnter2D:

If the weapon collides with an enemy:

- Add points for weapon damage to the score

- Call Die() of the enemy

- Destroy enemy after 1 second

OnCollisionEnter2D(Collision2D other):

- Destroy weapon object

## Finish Line

### Start:

- Check if enemyWaveSpawner is assigned
- Check if audioSource is assigned

### OnTriggerEnter2D:

- If the player collides with the finish line
  - Play the finish visual effect (VFX)
  - Stop enemy spawning from the enemyWaveSpawner
  - Destroy all enemies with DestroyAllEnemies()

### DestroyAllEnemies:

- Find all enemies
- For each enemy:
  - Instantiate the VFX - enemy explosion effect at the enemy's position
  - Play the sound effect (SFX) explosionSound
  - Destroy the enemy object

## Game Session

### Awake:

- Implement a singleton pattern to ensure there is only one GameSession object

### Start:

- Update UI with initial lives and score

### ProcessPlayerDeath:

- If player has more than 1 life:
  - TakeLife()
- Else:
  - ResetGameSession()

### AddToScore:

- Add points to the score
- Update UI with the new score

### TakeLife:

- Decrease playerLives by 1
- Reload current scene
- Update UI with remaining lives

### ResetGameSession:

Load the first scene  
Destroy GameSession object