



Project One

Technical Specifications

Kathy Stephens Student No: 1023257

Table of Contents

| | |
|-------------------------------------|-----------|
| Class Name: PlayerMovement | 3 |
| Method: Start | 3 |
| Method: Update | 3 |
| Method: OnMove | 3 |
| Method: OnFire | 4 |
| Method: OnJump | 4 |
| Method: Run | 4 |
| Method: FlipSprite | 4 |
| Method: Die | 5 |
| Class Name: EnemyMovement | 6 |
| Method: Start | 6 |
| Method: Update | 6 |
| Method: FlipEnemyFacing | 7 |
| Method: Die | 7 |
| Method: DestroyAfterAnimation | 7 |
| Class Name: EnemyWaveSpawner | 8 |
| Method: SpawnWaves (Coroutine) | 8 |
| Method: SpawnWave | 8 |
| Class Name: Weapon | 10 |
| Method: Start | 10 |
| Method: Update | 10 |
| Method: OnTriggerEnter2D | 10 |
| Method: OnCollisionEnter2D | 11 |
| Class Name: FinishLine | 12 |
| Method: Start | 12 |
| Method: OnTriggerEnter2D | 12 |
| Method: DestroyAllEnemies | 12 |
| Class Name: GameSession | 14 |
| Method: Awake | 14 |
| Method: Start | 14 |
| Method: ProcessPlayerDeath | 14 |
| Method: AddToScore | 15 |
| Method: TakeLife | 15 |
| Method: ResetGameSession | 15 |

Class Name: PlayerMovement

Note: The following are to be made as SerializeField so they can be editable in the Inspector, but protected in the script; runSpeed, jumpSpeed, deathKick, weapon, firePoint. In addition, require: moveInput, isAlive bool, myRigidbody, myBodyCollider and myAnimator.

Method: Start

- What are the inputs (variables/data required for this to work)? (if any)
 - Requires Rigidbody2D, Animator, and CapsuleCollider2D
- What are the outputs(variables/data changed or functions called by this running)? (if any)
 - Initialises myRigidbody, myAnimator, and myBodyCollider
- Describe what is happening, in order, using dot points in plain language (no programming jargon)
 - This is getting these player components and storing a reference so they can be used in the script

Method: Update

- What are the inputs (variables/data required for this to work)? (if any)
 - Requires isAlive bool
- What are the outputs(variables/data changed or functions called by this running)? (if any)
 - Calls the Run, FlipSprite and Die methods.
- Describe what is happening, in order, using dot points in plain language (no programming jargon)
 - Moves the player depending on the input (via the Run method)
 - Changes the sprite direction based on their movement (via the FlipSprite method)
 - Checks if the player should die (via Die method)

Method: OnMove

- What are the inputs (variables/data required for this to work)? (if any)
 - InputValue value
- What are the outputs(variables/data changed or functions called by this running)? (if any)
 - moveInput
- Describe what is happening, in order, using dot points in plain language (no programming jargon)
 - This is getting the player's movement input when they move the joystick/press the arrow keys

Method: OnFire

- What are the inputs (variables/data required for this to work)? (if any)
 - InputValue value

- What are the outputs(variables/data changed or functions called by this running)? (if any)
 - Instantiates a weapon at a specified position point (firePoint variable)
- Describe what is happening, in order, using dot points in plain language (no programming jargon)
 - This creates a weapon at the specified firing point

Method: OnJump

- What are the inputs (variables/data required for this to work)? (if any)
 - InputValue value
- What are the outputs(variables/data changed or functions called by this running)? (if any)
 -
- Describe what is happening, in order, using dot points in plain language (no programming jargon)
 - this adds vertical speed to the player when they press the jump button (makes them jump up!)

Method: Run

- What are the inputs (variables/data required for this to work)? (if any)
 - moveInput, runSpeed
- What are the outputs(variables/data changed or functions called by this running)? (if any)
 - Animator component, Rigidbody2D
- Describe what is happening, in order, using dot points in plain language (no programming jargon)
 - The player's horizontal speed is adjusted
 - The animation changes to running

Method: FlipSprite

- What are the inputs (variables/data required for this to work)? (if any)
 - myRigidbody
- What are the outputs(variables/data changed or functions called by this running)? (if any)
 -
- Describe what is happening, in order, using dot points in plain language (no programming jargon)
 - The sprite direction is changed, depending on the direction they're moving i.e. so if they are moving right, they are facing that way and vice versa.

Method: Die

- What are the inputs (variables/data required for this to work)? (if any)
 - myBodyCollider, deathKick, isAlive bool
- What are the outputs(variables/data changed or functions called by this running)? (if any)
 - isAlive + calls the "ProcessPlayerDeath" method from the "Game Session" script.

- Describe what is happening, in order, using dot points in plain language (no programming jargon)
 - When the player hits an enemy or one of the hazards it:
 - Changes the player state to being dead
 - Shows the 'death' animation
 - Applies the death kick effect (this gives it a little kick up in the air before hitting the ground)
 - Tells the other script called "Game Session" to run another function called 'Process Player Death' which will deduct a life from the score.

Class Name: EnemyMovement

Note: The following are to be made as SerializeField so they can be editable in the Inspector, but protected in the script; moveSpeed, deathAnimationTrigger

Method: Start

- What are the inputs (variables/data required for this to work)? (if any)
 - Need the Rigidbody2D component
- What are the outputs(variables/data changed or functions called by this running)? (if any)
 - myRigidbody
- Describe what is happening, in order, using dot points in plain language (no programming jargon)
 - This is getting the required enemy component and storing a reference so it can be used in the script.

Method: Update

- What are the inputs (variables/data required for this to work)? (if any)
 - moveSpeed, myRigidbody
- What are the outputs(variables/data changed or functions called by this running)? (if any)
 - Calls FlipEnemyFacing method
- Describe what is happening, in order, using dot points in plain language (no programming jargon)
 - This is setting the enemy's movement speed (horizontally)
 - It adjusts the direction the sprite faces (via the FlipEnemyFacing method)

Method: OnTriggerExit2D

- What are the inputs (variables/data required for this to work)? (if any)
 - Collider2D other
- What are the outputs(variables/data changed or functions called by this running)? (if any)
 - moveSpeed and calls FlipEnemyFacing method
- Describe what is happening, in order, using dot points in plain language (no programming jargon)
 - When the enemy hits a trigger (could be an object or an area):
 - It changes the direction the enemy faces (i.e. reverses direction)
 - Adjusts the sprite to match the direction of movement

Method: FlipEnemyFacing

- What are the inputs (variables/data required for this to work)? (if any)
 - myRigidbody
- What are the outputs(variables/data changed or functions called by this running)? (if any)
 - transform.localScale
- Describe what is happening, in order, using dot points in plain language (no programming jargon)
 - This is changing the direction of the sprite, depending on the direction the enemy is moving (flips it).

Method: Die

- What are the inputs (variables/data required for this to work)? (if any)
 - animator, deathAnimationTrigger
- What are the outputs(variables/data changed or functions called by this running)? (if any)
 - The DestroyAfterAnimation coroutine
- Describe what is happening, in order, using dot points in plain language (no programming jargon)
 - This gets the animator to show the 'death' animation
 - Begins the process of destroying the enemy after the animation (tells DestroyAfterAnimation to start)

Method: DestroyAfterAnimation

- What are the inputs (variables/data required for this to work)? (if any)
 -
- What are the outputs(variables/data changed or functions called by this running)? (if any)
 - The enemy game object
- Describe what is happening, in order, using dot points in plain language (no programming jargon)
 - Waits a short time
 - Then destroys the enemy

Class Name: EnemyWaveSpawner

Note: The following are to be made as SerializeField so they can be editable in the Inspector, but protected in the script; enemyPrefab, timeBetweenWaves, enemiesPerWave, spawnPoints.

Method: Start

- What are the inputs (variables/data required for this to work)? (if any)
 -
- What are the outputs(variables/data changed or functions called by this running)? (if any)
 - Calls the SpawnWave coroutine
- Describe what is happening, in order, using dot points in plain language (no programming jargon)
 - This begins spawning waves of enemies

Method: SpawnWaves (Coroutine)

- What are the inputs (variables/data required for this to work)? (if any)
 - timeBetweenWaves, SpawnWave method
- What are the outputs(variables/data changed or functions called by this running)? (if any)
 - Calls the SpawnWave method
- Describe what is happening, in order, using dot points in plain language (no programming jargon)
 - This keeps repeating endlessly on a loop
 - It calls on a method to create waves of enemies (the Spawn Wave method)
 - It waits a specified amount of time between each wave of enemies

Method: SpawnWave

- What are the inputs (variables/data required for this to work)? (if any)
 - enemyPrefab, enemiesPerWave, spawnPoints
- What are the outputs(variables/data changed or functions called by this running)? (if any)
 - spawnPoints
- Describe what is happening, in order, using dot points in plain language (no programming jargon)
 - It loops through the number of enemies needed to spawn
 - Then selects a random position to spawn the enemy
 - It creates the enemy in that position

Method: StopSpawning

- What are the inputs (variables/data required for this to work)? (if any)
 -
- What are the outputs(variables/data changed or functions called by this running)? (if any)
 - Stops the coroutine

- Describe what is happening, in order, using dot points in plain language (no programming jargon)
 - Stops enemies from spawning

Class Name: Weapon

Note: The following are to be made as SerializeField so they can be editable in the Inspector, but protected in the script; weaponSpeed, pointsForWeaponDamage. Also require; xSpeed

Method: Start

- What are the inputs (variables/data required for this to work)? (if any)
 - Rigidbody2D, PlayerMovement
- What are the outputs(variables/data changed or functions called by this running)? (if any)
 - myRigidbody, player, xSpeed
- Describe what is happening, in order, using dot points in plain language (no programming jargon)
 - This is getting the required weapon component and storing a reference so it can be used in the script.
 - It's finding the player in the scene and storing a reference to it (from the PlayerMovement script)
 - Calculating and storing the horizontal weapon (ammunition) speed, based on the direction of the player and the weapon speed.

Method: Update

- What are the inputs (variables/data required for this to work)? (if any)
 - xSpeed, myRigidbody
- What are the outputs(variables/data changed or functions called by this running)? (if any)
 -
- Describe what is happening, in order, using dot points in plain language (no programming jargon)
 - This sets the weapon's horizontal speed (i.e. speed of the bullet or laser)
 - Makes it fly across the scene

Method: OnTriggerEnter2D

- What are the inputs (variables/data required for this to work)? (if any)
 - Collider2D other
- What are the outputs(variables/data changed or functions called by this running)? (if any)
 - Calls EnemyMovement "Die" method, Adds points to the score via GameSession "AddToScore" method
- Describe what is happening, in order, using dot points in plain language (no programming jargon)
 - When the weapon collides with an object that is tagged as an "Enemy":
 - Points are added to the score
 - It calls the death method from another script (EnemyMovement)
 - It destroys the enemy after a small delay

Method: OnCollisionEnter2D

- What are the inputs (variables/data required for this to work)? (if any)
 - Collision2D other
- What are the outputs(variables/data changed or functions called by this running)? (if any)
 - The weapon game object
- Describe what is happening, in order, using dot points in plain language (no programming jargon)
 - Destroys the weapon (i.e. the ammunition: bullet or laser) when it collides with any object

Class Name: FinishLine

Note: The following are to be made as SerializeField so they can be editable in the Inspector, but protected in the script; finishEffect, enemyWaveSpawner, enemyExplosionEffect, audioSource, explosionSound

Method: Start

- What are the inputs (variables/data required for this to work)? (if any)
 - Need the enemyWaveSpawner and audioSource
- What are the outputs(variables/data changed or functions called by this running)? (if any)
 - audioSource
- Describe what is happening, in order, using dot points in plain language (no programming jargon)
 - This is checking if the enemyWaveSpawner has been assigned. It logs an error if it hasn't
 - It checks if the audioSource is assigned and tries to get the component if not.

Method: OnTriggerEnter2D

- What are the inputs (variables/data required for this to work)? (if any)
 - Collider2D other
- What are the outputs(variables/data changed or functions called by this running)? (if any)
 - Calls the DestroyAllEnemies method and StopSpawning method (from the EnemyWaveSpawner script)
- Describe what is happening, in order, using dot points in plain language (no programming jargon)
 - When the player collides with the finish line:
 - The finish VFX is played
 - It stops any further enemies from spawning
 - It calls another method (destroy all enemies) so any remaining enemies can be destroyed

Method: DestroyAllEnemies

- What are the inputs (variables/data required for this to work)? (if any)
 - Need the audioSource, enemyExplosionEffect, and explosionSound
- What are the outputs(variables/data changed or functions called by this running)? (if any)
 - All enemies tagged as "Enemy"
- Describe what is happening, in order, using dot points in plain language (no programming jargon)
 - This method finds all the enemies in the scene

- It plays a explosion VFX
- Plays an explosion SFX
- Destroys the remaining enemies

Class Name: GameSession

Note: The following are to be made as SerializeField so they can be editable in the Inspector, but protected in the script; playerLives, score

Method: Awake

- What are the inputs (variables/data required for this to work)? (if any)
 - Needs the GameSession objects to be present in the scene
- What are the outputs(variables/data changed or functions called by this running)? (if any)
 - ** My initial plan was to have multiple levels for this game, so this would ensure the game session persisted across each scene load. I ran out of time and only finished the one level, but kept the code in as it's something I'm probably going to come back to and expand.
- Describe what is happening, in order, using dot points in plain language (no programming jargon)
 - This checks how many game sessions there are and ensures that there is only one scene running

Method: Start

- What are the inputs (variables/data required for this to work)? (if any)
 - playerLives, score
- What are the outputs(variables/data changed or functions called by this running)? (if any)
 - Updates the UI display. Converts the playerLives and score integers into strings.
- Describe what is happening, in order, using dot points in plain language (no programming jargon)
 - This is setting UI display to show the number of lives and the score.

Method: ProcessPlayerDeath

- What are the inputs (variables/data required for this to work)? (if any)
 - playerLives
- What are the outputs(variables/data changed or functions called by this running)? (if any)
 - Calls TakeLife method (if more than one life remaining) and ResetGameSession method (if no lives left).
- Describe what is happening, in order, using dot points in plain language (no programming jargon)
 - Checks how many lives the player has remaining.
 - If they have more than one, it calls another method (TakeLife) to decrease the lives
 - If they have no more lives, it calls another method (ResetGameSession) to reload the game

Method: AddToScore

- What are the inputs (variables/data required for this to work)? (if any)
 - int pointsToAdd
- What are the outputs(variables/data changed or functions called by this running)? (if any)
 - Updates the score and UI
- Describe what is happening, in order, using dot points in plain language (no programming jargon)
 - Adds points to the current score
 - Updates the score displayed on the UI

Method: TakeLife

- What are the inputs (variables/data required for this to work)? (if any)
 - playerLives
- What are the outputs(variables/data changed or functions called by this running)? (if any)
 - Decreases playerLives, updates UI and reloads scene
- Describe what is happening, in order, using dot points in plain language (no programming jargon)
 - This decreases the players lives by one
 - It reloads the scene
 - It updates the UI with the new amount of lives

Method: ResetGameSession

- What are the inputs (variables/data required for this to work)? (if any)
 -
- What are the outputs(variables/data changed or functions called by this running)? (if any)
 - Loads the first scene (index 0) and destroys the current session
- Describe what is happening, in order, using dot points in plain language (no programming jargon)
 - When the player has run out of lives the current game session is destroyed and reset.