# Pseudocode:

**AudioPlay Class:**

On Start

Get AudioSource Component

On AudioSource Enable

If Animator State is Player\_Move Animation

Play Walking Audio

Call RandomizeSfx method

RandomizeSfx method

Set audio to a random pitch

Play Audio

**CameraController Class:**

Update method

Find Player object as target

Follow Player object

**Dialogue Class:**

TextArea (minimum lines 1, maximum lines 2)

String array sentences

**DialogueController Class:**

On Start

Add prepopulated sentences to a queue

StartDialogue method

Clear any previous sentnces

Foreach sentence

Enqueue sentence to the sentences queue

Call DisplayNextSentence method

DisplayNextSentence method

If sentences count is zero

Call EndDialogue method

Return

Dequeue the current sentence from the queue

Call StopAllCoroutines method

Call the TypeSentence Coroutine and display the current sentence

IEnumerator TypeSentence

dialogueText = “”

Foreach letter in a sentence

Display letter

Yield return new WaitForSeconds(some delay)

Yield return new WaitForSeconds(some delay)

Call DisplayNextSentence method

EndDialogue method

End Convesation

**DialogueTrigger Class:**

On Start

Find DialogueController and call StartDialogue Method

**DisplayText Class:**

On Start

Set the text object active to false

Set changeText attribute to Text object

ChangeText method

Set Text object text to something else

OnTriggerEnter2D method

If collider collides with Player object

Set Text object active to true

OnTriggerExit2D method

Set Text object active back to false

**Door Class:**

CheckForOpen Method

If door is openable

If door is locked

If the keycard is in the player inventory

Unlock door

Change door text

Else door was not unlocked

Else the door is unlocked

**Enemy Class:**

On Start

Find player object with PlayerController

Update

Move the enemy to the right based on speed value over time

Raycasting for ground detection

If enemy collides with a wall or there is no floor

Call FlipEnemy method

Set Animator speed to enemy speed

If speed is greater than zero

Set animator to moving

Set animator to not idle

Else

Set animator to not moving

Set animator to idle

TakeDamage method with int damage parameter

Health -= damage

If health <= 0

Call Die method

FlipEnemy method

If moving right

Flip enemy 180 on x axis

Moving right = false

Facing right = false

Else

Flip enemy 180 on x axis

Moving right = true

Facing right = true

Die method

Set animator to dead

If not facing right

Flip enemy 180 on x axis

Instantiate death effect animation at current position

Destroy game object

OnTriggerEnter2D method

If collided with player object

Player takes damage to value of enemy damage given

Call PlayerController SetInvincible method

Set player knockback count to player knockback length

If player hit on right side

PlayerController knock from right is true

Else

PlayerController knock from right is false

If enemy collides with another enemy

Call FlipEnemy method

**EnemyPatrol Class:**

On Start

Find player object with PlayerController

Update

Move the enemy to the right based on speed value over time

Raycasting for ground detection

If enemy collides with a wall or there is no floor

Call Turn method

Die method

Health - 1

If Health less than or equal to zero

Dead attribute equals true

Instantiate death effect animation at current position

Destroy game object

Turn Method

If moving right

Flip enemy 180 on x axis

Moving right = false

Facing right = false

Else

Flip enemy 180 on x axis

Moving right = true

Facing right = true

TakeDamage method with int damage parameter

Health -= damage

If health <= 0

Call Die method

OnTriggerEnter2D method

If enemy collides with another enemy

Call Turn method

If collided with player object

Player takes damage to value of enemy damage given

Call PlayerController SetInvincible method

Set player knockback count to player knockback length

If player hit on right side

PlayerController knock from right is true

Else

PlayerController knock from right is false

**HUD Class:**

On Start

Find player object with PlayerController

Update

Health UI sprite = HealthSprites array with player currentHealth as a parameter

**InteractionObject Class:**

DoInteraction method

Set interacted game object active to false

Open Method

Set door collider to a trigger collider

Run door opening animation

**IntroCutscene Class:**

On Start

Create instance of the Player object’s RigidBody2D

StartCoroutine JumpToScene Method

FixedUpdate

Speed = (transform.position – lastPosition).magnitude

lastPosition = transform.position

IEnumerator JumpToScene method

Wait 30 seconds

Call LoadByIndex Method

LoadByIndex Method

Load next scene from game build index

**Inventory Class:**

Create new array instance of inventory to a size of 1

AddItem method

Set itemAdded to false

Search through the array for the 1st available item slot

If an empty slot is found

Add item to this slot

itemAdded = true

SendMessage(“DoInteraction”) to InteractionObject Script

break

if itemAdded is false

Inventory is full

FindItem method

Search through the array for the item required

If item is found in array

Return true

Return false

**Laser Class:**

On Start

Set RigidBody2D Velocity to transform.right multiplied by speed

Create instance of the AudioSource

Set play audio to true

On Update

If the laser sound play is set to true

Play the sound

OnTriggerEnter2D

Get Enemy Object

If Enemy is not null

Enemy takes damage

Destroy laser object

**LoadScene Class:**

LoadByIndex method

Use the SceneManager to load the scene with the appropriate index reference

**PlayerController Class:**

On Start

Set amount of extra jumps to extra jump value

Get Rigidbody2D Component

Set “Idle” animation to true in animator

Set Player current health equal to max health

FixedUpdate

isGrounded attribute equals current position of the groundCheck object attached to the player

moveInput attribute equals movement on the x axis

rigidbody velocity equals moveInput \* speed

if the player is not facing right and the player moves in a positive direction on the x axis

Call Flip method

if the player is facing right and the player moves in a negative direction on the x axis

Call Flip method

On Update

Set animator speed parameter to the value of moveInput

If rigidbody y axis value is less than zero

Play falling animation

Else

Stop playing falling animation

If isGrounded is true

Set extra jump to extra jump value

Deactivate any jumping animation playing

If up arrow is pressed and extra jump attribute is greater than zero

Rigidbody velocity equals Vector2.up multiplied by jumpForce attribute value

Extra jump – 1

Set “Jump” animation to true

Else If up arrow is pressed and extra jump attribute is greater than zero

Rigidbody velocity equals Vector2.up multiplied by jumpForce attribute value

Extra jump – 1

Set “Jump” animation to true

If current animation is “Idle” and shoot button is pressed

Play “Shooting” animation

Else

Disable “Shooting” animation

Play “Idle” animation

If current animation is “Jumping” and shoot button is pressed

Disable “Jumping” animation

Disable “Falling” animation

Play “JumpShooting” animation

Else

Disable “JumpShooting” animation

If current animation is “Player\_Move” and shoot button is pressed

Play “MoveShooting” animation

Else

Disable “MoveShooting” animation

If current health is greater than max health value

current health equals max health value

If current health less than zero

Call Die method

If knockbackCount attribute value is less than or equal to zero

Rigidbody velocity = moveVelocity attribute value [x axis]

Else

If knockFromRight is true

Rigidbody velocity = -knockback attribute value, knockback attribute value

If knockFromRight is false

Rigidbody velocity = knockback attribute value, knockback attribute value

knockbackCount -= Time.deltaTime

OnLanding method

Set “Jumping” animation to false

Set “Falling” animation to false

Set “Damaged” animation to false

Set “Idle” animation to true

Flip method

facingRight attribute equals not facingRight

Flip sprite 180 degrees on x axis

SetInvincible method

If not invincible

Set invincible attribute to true

Invoke SetDamagable method after invincibleTime has passed

SetDamagable method

Invincible attribute equals false

TakeDamage method (passes damage variable from enemy)

If not invincible

Set invincible attribute to true

Invoke SetDamagable method after invincibleTime has passed

Current player health -= damage value

Play “Player\_Damage” animation in full

Else

Damage done equals zero

AddHealth method (passes health variable from health pickup object)

Current player health += health

If current health is greater than max health allowed

current health equals max health

Die method

Load current scene

**PlayerInteraction Class:**

On Update

If the current object is openable

If it is locked

If the inventory contains the item required to open the object

Object.locked set to false

Object was unlocked

Else

Object was not unlocked

Else

Open the object

Object was opened

OnTriggerEnter2D method

If collided with an interact able object

Current interact able object equals this object

Create instance of CurrentInteractableObject script

If this interact able object can be added to the inventory

Add this to the player inventory

Else if this interact able object is a health pick-up

Destroy game object

Update player health

OnTriggerExit2D method

Set interact able object to null

Set CurrentInteractableObject script to null

**QuitOnClick Class:**

Quit method

If unity editor isPlaying is true

Set this to false

Else

Application Quit

**SceneTransition Class:**

On Start

Call instance of the player inventory

OnTriggerEnter2D method

If collided with player object

StartCoroutine(LoadScene) method

IEnumerator LoadScene method

Set transition animation to “End” animation

Wait 1 second

Load required scene by name

Clear player inventory

**SelectOnInput Class:**

On Update

If up or down arrow key is detected and a button is not selected

Set selected game object (button)

Button selected is true

OnDisable method

Set button selected to false

**SoundManager Class:**

On Awake

Load instances of the AudioClips

DontDestroyOnLoad(AudioClip)

Update

If the Scene is the Main Menu

Play Main Menu audio

If the Scene is the Cutscene

Play Cutscene audio

If the Scene is Level1

Play Level1 audio

If the Scene is Level2

Play Level2 audio

If the Scene is Level3

Play Level3 audio

**Weapon Class:**

On Update

If the shoot button is pressed

Call Shoot method

Shoot method

Instantiate Laser Prefab at current fire point position