# Player Controls.

## Movement.

Player can move their plane with the WASD keys

void movementController()

// this method gets user input and translates it into movement velocity

{

// Input.GetAxis gets user input and converts it to a value from 0 to 1

float moveHorizontal = (float)Math.Round(Input.GetAxis("Horizontal"), 2); //rounding it to 2 decimal values

float moveVertical = (float)Math.Round(Input.GetAxis("Vertical"), 2);

//to make animation less smooth

Vector2 movement = new Vector2(moveHorizontal, moveVertical);

GetComponent<Rigidbody2D>().velocity = movement \* speed;

boundaryClamper();

}

## [[1]](#endnote-1)Collision.

If the player collides with an enemy or obstacles they will take damage, player has 3 lives, if lives gets to 0, the player dies and is de-activated until the game is restarted

void OnTriggerEnter2D(Collider2D collision)

{

//if player is hit by an enemy or obstacle and time since last death is greater than 1 second

if (Time.time > deathBuffer && (collision.tag == EnemyShot.tag || collision.tag == Enemy.tag))

{

PlayerTookDamage(collision);

Destroy(collision.gameObject);

}

if (Time.time > deathBuffer && !onUpperLayer && (collision.tag == Land1.tag || collision.tag == Land2.tag))

{

PlayerTookDamage(collision);

}

}

This even is triggered when the player collides with another game objects collider component.

The Playertookdamage method is then called if the player takes damage:

void PlayerTookDamage(Collider2D collision)

//method for when player takes damage from any source.

{

playerLives--; //decrement playerlives

StartCoroutine (FlashDamage(GetComponent<SpriteRenderer>(), 3)); //flashes sprite 3 times

if (PlayerScript.playerLives < 1) //if player lives < 1 the player has died.

PlayerDied();

transform.position = new Vector2(0, -5); //reset player position

deathBuffer = Time.time + 1f; //player cant die in quick succession, addds 1 second to death buffer

}

## Layer.

The player can only collide with obstacles if they are on the upper layer, they can only shoot enemy planes on the upper layer, and can only shoot turrets if on the lower layer.

void layerController()

{

if (Input.GetKeyDown(KeyCode.F) && onUpperLayer)

// if F is pressed then move to lower layer

{

newScale = minScale;

onUpperLayer = false;

// this boolean value tells us whether or not player is on a certain layer

}

else if (Input.GetKeyDown(KeyCode.R) && !onUpperLayer)

// if r is pressed then move to Upper layer

{

newScale = maxScale;

onUpperLayer = true;

}

// Lerp function to smooth the transition-resize animation

GetComponent<Rigidbody2D>().transform.localScale = Vector2.Lerp(transform.localScale, newScale, Time.deltaTime \* layerTransitionSpeed);

boundaryClamper(); //clamp player to boundary after resize

}

## Shoot

Play can shoot two bullets if the fire button is pressed and or held

void shootController()

{

// if time passed is greater than the nextFire wait buffer (0.15 seconds)

// and the fire button is pressed

if (Time.time > nextFire && Input.GetButton("Fire1"))

{

nextFire = Time.time + fireRate;

//shoot by instantiating two bullets

Instantiate(SingleShot, ShotSpawn.position - new Vector3(0.3f, 0, 0), ShotSpawn.rotation);

Instantiate(SingleShot, ShotSpawn.position - new Vector3(-0.3f, 0, 0), ShotSpawn.rotation);

}

}

# Spawn points

Enemies(Planes and turrets) and Obstacles(Islands and Battleships) both spawn from two spawn points located at the top of the screen.

### Enemy spawn points:

### Obstacle spawn points:

# Enemies.

planes shoot 3 bullets that spread, every X seconds

Turrets face player and shoot 1 bullet every X seconds

Move left or right to cross middle depending on spawn location

# Obstacles.

Move straight downwards and carry turrets

# Menu System

Pause Menu

Main Menu

High Scores Menu

Death Screen

# GUI Score and Life display

# Score system

1. [↑](#endnote-ref-1)