Animation And Scene Management Lab

Friday Deadline 11th November 1pm

**Main Menu**

Create a main menu scene with 2 buttons “Play” and “Game Settings “.

Play loads the Game Scene.

Game Settings loads a Game Setting Scene

**Game Setting Scene**

4 Button “Easy “, “Hard”, “Impossible”;, “Back”

This set the difficulty level.

“back” returns to mainmenu.

When the user hits play on the main menu the current difficulty level will be used.

The game is more difficult with smaller time periods between the crates appearing.

Note : You must add a new scene to the build settings.

Use the below singleton class in the MainMenu scene. Create an empty gameobject and add it to the scene. This is only in MainMenu Scene.

public class ExampleGameManager : MonoBehaviour

{

public static ExampleGameManager instance;

//example

public int difficultyLevel.;

private void Awake()

{

numtimes = 0;

if (instance == null)

{

instance = this;

DontDestroyOnLoad(gameObject);

}

else if (instance != this)

{

Destroy(gameObject);

}

}

}

Game Scene

Use the cat animation from this folder.

Create the cat animation

Add the animations listed below using the animator , add a state integer as a parameter.

Add a trigger called dead.

Add a transition from the any state to the dead state and use the dead trigger as the condition.

Define the transitions between animations based on the state variable and what value it equals.

Cat Controller starter Script.

public enum States

{

Idle =0,

Running=1,

Jumping =2,

Falling=3,

Sliding=4,

Dead=5

}

public class catController : MonoBehaviour

{

public Rigidbody2D rig;

public Animator anim;

public BoxCollider2D boxCollider;

public States state;

public LayerMask groundLayer;

float horizontalInput;

float jumpVerticalPushOff=?;

Vector2 savedlocalScale;

float horizonatlSpeed = ?;

void Start()

{

rig = GetComponent<Rigidbody2D>();

anim = GetComponent<Animator>();

boxCollider = GetComponent<BoxCollider2D>();

savedlocalScale = transform.localScale;

}

// Update is called once per frame

void Update()

{

horizontalInput = Input.GetAxisRaw("Horizontal");

// what do the below lines do ?

if (horizontalInput > 0.001f)

{

transform.localScale = new Vector2(savedlocalScale.x,savedlocalScale.y);

}

else if (horizontalInput < -0.001f)

{

transform.localScale = new Vector2(-savedlocalScale.x,savedlocalScale.y);

}

if (state == States.Idle )

{

if (IsGrounded())

{

if (Input.GetKey(KeyCode.Space))

{

rig.velocity = new Vector2(?, ?);

state = ?;

}

}

}

if ( state == States.Running )

{

}

else if (state == States.Jumping)

{

//test if the cat is moving down the screen. Change state to falling

//If it hits the ground then state is idle

}

else if (state == States.Sliding)

{

//If moving and left shift is not pressed then run

//it no input idle

}

else if (state == States.Falling)

{

//If moving and left shift is not pressed then run

//it no input idle

}

else if (state == States.Dead)

{

}

rig.velocity = new Vector2(horizontalInput \* horizonatlSpeed , rig.velocity.y) ;

anim.SetInteger("State", (int)state);

}

private bool IsGrounded()

{

RaycastHit2D raycastHit = Physics2D.BoxCast(boxCollider.bounds.center, boxCollider.bounds.size, 0, Vector2.down, 0.1f, groundLayer);

return raycastHit.collider != null;

}

}

Create a new layer ( it’s not a sorting layer) called ground.

Add a ground sprite , change the width to be the same width as the screen.

Set the layer to be ‘ground’

We could also use the

private void onCollisionEnter2D ( Collision2D collision)

{

If (collision.gameObject.tag == “ground”)

{

//use a grounded bool.

grounded=true;

}

}

We also add a grounded variable. Set it to false when the cat jumps.

I introduced the raycast method here because it can be used in many situations in Unity .

**Crates**

Spawned Crates arrive from the left or right of the screen.

Create 2 empty gameobjects in the scene and place them to the left and the right of the screen, just off the screen. Add a script that periodically instantiates a crate and moves it towards the player.

If the player is sliding in the opposite direction the crate is moving, then add points to the player and destroy the crate. Otherwise, the player loses a life.

Marks

Cat Movement 10

Cat Animation States in Animator 20

Cat animated correctly 20

Crates 5

Scoring and lives 5

Scene Management 10

Extras : 30 ( only when completed first 80 marks, discuss with me)