Arctic Adventure

Team:

* Pablo Dávila
* Roberto Rivera
* Catalina Domínguez

Sowftware Engineering Document

Table of Contents

GAME MECHANICS

Game Scenes

Game Flow

CLASSES AND SCRIPTS

GAME MECHANICS

Game Scenes

* The game has 4 basic scenes:
* Main Menu. Allows the user to Start the game or quit.
* Game scene. Shows the platforms where the player will jump to gain points.
* Game Over. Shown if the player falls from the platforms. Allows the player to submit their score to the leaderboard.
* Leaderboard scene allows the player to see the leaderboard, return to the main menu or go to the game scene directly.

Game Flow

* Start scene leads to the Game scene.
* Game scene can lead to:
  + Game over scene
* Game over allows the player to submit a score and goes to leaderboard.
* Leaderboard scene can lead to main menu scene or game scene to restart the game.

CLASSES AND SCRIPTS

For this game we need several objects.

* **Level Manager** – Manages the scenes.

Methods:

void OnTriggerEnter2D(Collider2D other)

* **Camera Control** – Gives the camera the position to follow.

Attributes:

public GameObject player;

private Vector3 offset;

Methods:

void Start ()

void Update ()

* **Camera Follow** – Gives the camera the instructions to follow the player to where it moves.

Attributes:

public float xMargin = 1f;

public float yMargin = 1f;

public float xSmooth = 8f;

public float ySmooth = 8f;

public Vector2 maxXAndY;

public Vector2 minXAndY;

private Transform player;

Methods:

void Awake ()

bool CheckXMargin()

bool CheckYMargin()

void FixedUpdate ()

void TrackPlayer ()

* **Coin** – Determines the value of the object the player is collecting.

Methods:

void Start()

void Update()

void OnTriggerEnter2D(Collider2D other)

* **Destroyer** – Allows to destroy game objects.

Attributes:

public bool destroyOnAwake;

public float awakeDestroyDelay;

public bool findChild = false;

public string namedChild;

Methods:

void Awake ()

void DestroyChildGameObject ()

void DisableChildGameObject ()

void DestroyGameObject ()

* **Enemy** – Gives the enemy attributes like speed and the actions to make when the

Attributes:

player interacts with it.

public float moveSpeed = 2f;

public Transform groundCheck;

private bool grounded = false;

private SpriteRenderer ren;

private bool dead = false;

Methods:

void Awake()

void FixedUpdate()

public void Flip()

* **Follow Player** – Finds the player in order for the camera to follow it.

Attributes:

public Vector3 offset;

private Transform player;

Methods:

void Awake()

void Update ()

* **Leaderboard** – Compares all the scores of the game every time it has been played and displays a list.

Attributes:

private InputField inputField;

private GameObject leaderBoard;

Methods:

void Start()

public static List()

public void showLeaderboard()

public void Submit()

void Update()

* **Lives** – Manages the lives of the player.

Attributes:

public static int lives = 2;

Methods:

void Update ()

public static void Reset()

* **Pauser** – Allows the player to pause and continue the game at any time.

Attributes:

public bool paused = false;

Methods:

void Update ()

* **Platform Fall** – Gives each platform a delay and after that time, it falls.

Attributes:

public float fallDelay = 1f;

private Rigidbody2D rb2d;

Methods:

void Awake()

void OnCollisionEnter2D(Collision2D other)

void Fall()

* **Player Control** – This object is out player itself. It has its attributes and the actions to follow each time.

Attributes:

public bool facingRight = true;

public bool jump = false;

public float moveForce = 365f;

public float maxSpeed = 5f;

public AudioClip[] jumpClips;

public float jumpForce = 1000f;

public AudioClip[] taunts;

public float tauntProbability = 50f;

public float tauntDelay = 1f;

private int tauntIndex;

public Transform groundCheck;

private bool grounded = false;

private Animator anim;

Methods:

void Awake()

void Update()

void FixedUpdate ()

void Flip ()

public IEnumerator Taunt()

int TauntRandom()

* **Player Health** – Manages the player’s health.

Attributes:

public float health = 100f;

public float repeatDamagePeriod = 2f;

public AudioClip[] ouchClips;

public float hurtForce = 10f;

public float damageAmount = 10f;

private SpriteRenderer healthBar;

private float lastHitTime;

private Vector3 healthScale;

private PlayerControl playerControl;

private Animator anim;

Methods:

void Awake ()

void OnCollisionEnter2D (Collision2D col)

void TakeDamage (Transform enemy)

public void UpdateHealthBar ()

* **Power Up** – Manages the questions asked in the game and the rewards or penalties the player gets depending on the answer given.

Attributes:

public static Canvas pantalla;

public TextAsset textFile;

private static List<string> preguntas;

private static List<bool> respuestas;

private static int index;

public static bool isAsking;

static Power\_Up instance;

public Sprite[] sprites;

public float jpd;

public static float JPduration;

public static int type;

private static Text qText;

Methods:

void Start()

void OnTriggerEnter2D(Collider2D other)

void LoadQuestion()

public void callSelectedAnswer(bool button)

static public void selectedAnswer(bool button)

static IEnumerator Wait(float duration, bool powerup)

void processTextFile()

static void PowerUP()

static IEnumerator JetPack()

* **Remover** – Used when the player dies. Stops the object related to the player.

Attributes:

public GameObject splash;

Methods:

void OnTriggerEnter2D(Collider2D col)

IEnumerator ReloadGame()

* **Reset** – When the player’s lives reach zero and they answer the question correctly, it allows them to start the game over from base 1, if the answer is incorrect, they lose.

Attributes:

private Score score;

Methods:

void Start()

void Update()

void OnTriggerEnter2D(Collider2D other)

* **Score** – Calculates the score the player is making by collecting the objects. Allows to reset the score to zero when the game starts over.

Attributes:

private PlayerControl playerControl;

private int previousScore = 0;

public static int score = 0;

Methods:

void Awake()

void Update()

public static void Reset()

* **Score Shadow** – Gives the score a shadow when displayed on screen.

Attributes:

public GameObject guiCopy;

Methods:

void Awake()

void Update()

* **Set Particle Sorting Layer** – Organizes the particles in layers.

Attributes:

public string sortingLayerName;

Methods:

void Start ()

* **Simple Controller** – Allows the player to run, jump, flip and get lives throughout the game.

Attributes:

public bool facingRight = true;

public bool jump = false;

public float moveForce = 365f;

public float maxSpeed = 5f;

public float jumpForce = 1000f;

public Transform groundCheck;

public bool jetpackActive;

public float jetpackForce;

private bool grounded = false;

private bool jetpack = false;

private Animator anim;

private Rigidbody2D rb2d;

Methods:

void Awake()

void Update()

void FixedUpdate()

void Flip()

public void oneUP()

* **Spawn Coins** – Clones and sets the coins in random positions.

Attributes:

public Transform[] coinSpawns;

public GameObject coin;

Methods:

void Start()

void Spawn()

* **Spawn Manager** – Clones and sets the platforms in random positions.

Attributes:

public int maxPlatforms = 18;

public GameObject platform;

public float horizontalMin = 6f;

public float horizontalMax = 15f;

public float verticalMin = -6f;

public float verticalMax = 6f;

private Vector2 originPosition;

Methods:

void Start()

void Spawn()

* **Spawner** – Clones and sets the enemy in random positions.

Attributes:

public float spawnTime = 5f;

public float spawnDelay = 3f;

public GameObject[] enemies;

Methods:

void Start()

void Spawn()

* **Submit Score** – Saves the score gained by the player in the leaderboard.

Attributes:

private InputField inputField;

Methods:

void Start ()

public void Submit()

* **Score Board** – Saves the score with the player’s name.

Attributes:

public int score;

public string name;

Methods:

public scoreboard(string name)

* **Submit Button** – Allows to submit the score to the leaderboard when clicked on.

Attributes:

Leaderboard leaderboard;

Methods:

void Start ()