Module Name: High-Level Game Development

Module Code: CE318

Student ID: 1704969

Game Title: Space Adventure Game

Project Files: <https://essexuniversity-my.sharepoint.com/:u:/g/personal/tw17086_essex_ac_uk/EYQO3PM7KdtLl2fytrv7UwMBgbKh3nBSAMsQRYBMNnhqfA>

Presentation File: <https://essexuniversity-my.sharepoint.com/:b:/g/personal/tw17086_essex_ac_uk/EV3xm3oH8MRLm-dg2dMhZp0BN-q0h6_oneHwKZ71PBg-fQ>

Presentation Video: <https://essexuniversity-my.sharepoint.com/:v:/g/personal/tw17086_essex_ac_uk/ERVzhLWkuAhLoaErFNyTn58B8MqOgC3TCTWOqmmjW_vwtA>

1. Game Structure :

• Menus: main menu, options, pause and end menu.

Main Menu and options menu – The main menu has two options: ‘start’ and ‘options’. ‘Start’ starts the game, i.e. it loads a level. ‘Options’ opens the options menu (the camera pans to it). The player can change the difficulty setting (toggle that cycles through options), turn music off/on and change volume of audio.

A picture containing indoor, table

Description automatically generated

Pause menu – pausing stops the game completely and shows the following menu. The player can save and quit to the main menu, or choose another level to go to. The player can choose a level from the dropdown and then click ‘Travel to Planet’ to load the selected level.



There is no end menu in my game. This is because there is no ‘end’ – there are no game overs, you always respawn when you die. Also, the player can keep playing forever – they are never forced to quit.

• Splash screen when the game starts.

• 2 playable scenes.

The first playable scene is on a grass plain with a small hill and mountains on all sides (to keep the player in). There are large shaped stones all around that might have placed by ancient dwellers (in-game lore).

The second scene is on a floating set of platforms – such that the player can fall to their ‘death’. The platforms are metallic – perhaps they make up an enemy base.

A picture containing floor, outdoor

Description automatically generated

• There must be fade in/out transitions between screens, menus and scenes.

There is a fade in and out transition when travelling to another level.

//function to set up level change

public void toOtherLevel()

{

PlayerPrefs.SetInt("LastLevel", drop.value + 1);

StartCoroutine(ChangeLevel(drop.value + 1));

FindObjectOfType<LevelState>().Save();

Time.timeScale = 1;

}

//function to call fade and load new level

IEnumerator ChangeLevel(int scene\_id)

{

FaderScript fs = GetComponent<FaderScript>();

fs.BeginFade(-1);

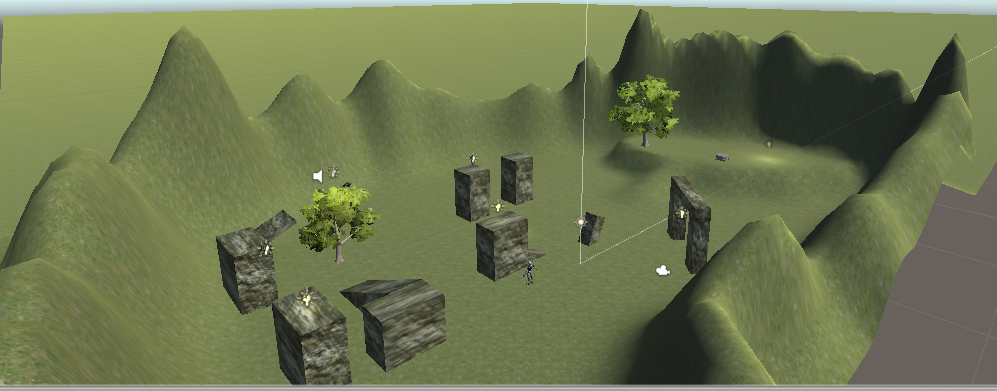
yield return new WaitForSeconds(1);

SceneManager.LoadScene(scene\_id);

}

• Terrain usage

The first level is played on a terrain. The terrain is used to stop the player from leaving the main area, to make some raised ground, and to have trees in the level.



2. Gameplay :

• Different collectable item.

The first of the collectables are ‘batteries’. They are the most common collectable. They restore one point of health when picked up.

A picture containing outdoor, grass, sky, bench

Description automatically generated

The next type of collectable, which I refer to as ‘Data points’ in the game represent short term goals for the player. They mark points of interest and single challenges, that organically guide the player through the level.



The other type of collectable which I refer to as ‘Big Data points’ in the game represent long term goals for the player. They mark the end of multiple platforming challenges. In a full game they would be needed to progress in the game.

A picture containing tree, outdoor, grass, sport

Description automatically generated

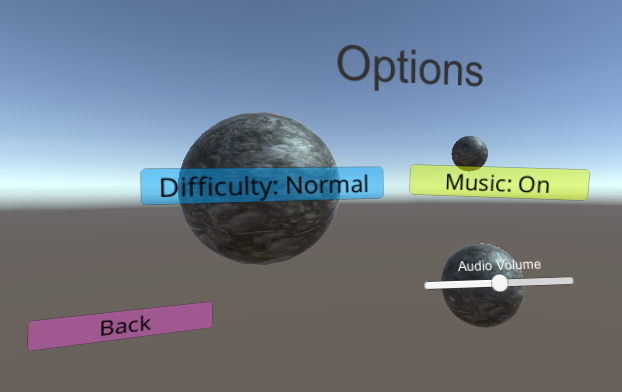
• Difficulty levels.

3 difficulty levels: normal, hard, daredevil. ‘Normal’ is the intended difficulty and the others are harder. The difficulty setting changes the maximum HP that the player has; Normal has 3, while daredevil is called such because the player has only 1 hit.



Though, this is all that the difficulty changes in the game. This is appropriate for my game, since the challenge comes from the level design in conjunction with the player’s performance. It doesn’t make sense to change anything else to make the game harder as that would be more likely to break the level design than provide more difficulty with the same fun.

• Game parameters.



The player, in the options menu, can set the difficulty setting, turn music on/off, and adjust audio volume (applies to all audio). The preferences of the player are stored in PlayerPrefs.

//enum for referencing difficulty levels

enum difficulty

{

Normal,

Harder,

Daredevil

}

difficulty diff\_setting;

//initialises difficulty, using PlayerPrefs if possible

private void Start()

{

if (PlayerPrefs.HasKey("Difficulty")) diff\_setting = (difficulty)PlayerPrefs.GetInt("Difficulty");

else diff\_setting = difficulty.Normal;

diff\_text.text = "Difficulty: " + diff\_setting.ToString();

PlayerPrefs.SetInt("Difficulty", (int)diff\_setting);

}

//change difficulty setting

public void changeDifficulty()

{

if (diff\_setting++ == difficulty.Daredevil) diff\_setting = difficulty.Normal;

diff\_text.text = "Difficulty: " + diff\_setting.ToString();

PlayerPrefs.SetInt("Difficulty", (int)diff\_setting);

}

• It must be possible to save and load games.

Saving and loading is done automatically when moving between levels.

//save function of LevelState script

//save the state of each level – mark collectables that have been collected

public void Save()

{

string filename = Application.persistentDataPath + "/level" + SceneManager.GetActiveScene() + ".dat";

GameObject[] data = GameObject.FindGameObjectsWithTag("SmallData");

GameObject[] bigdata = GameObject.FindGameObjectsWithTag("BigData");

CollectionState cs = new CollectionState

{

data = new bool[data.Length],

bigdata = new bool[bigdata.Length]

};

for (int i = 0; i < data.Length; i++)

{

cs.data[i] = data[i].activeSelf;

}

for (int i = 0; i < bigdata.Length; i++)

{

cs.bigdata[i] = bigdata[i].activeSelf;

}

BinaryFormatter bf = new BinaryFormatter();

FileStream file = File.Open(filename, FileMode.OpenOrCreate);

bf.Serialize(file, cs);

file.Close();

}

//save function in GameplayController

public void Save()

{

string filename = Application.persistentDataPath + "/savegame.dat";

SaveGame sg = new SaveGame(dataPointsCount, batteryCount, bigDataPointsCount, health);

BinaryFormatter bf = new BinaryFormatter();

FileStream file = File.Open(filename, FileMode.OpenOrCreate);

bf.Serialize(file, sg);

file.Close();

}

The GameplayController script also uses a singleton system to keep the right score between levels.

3. Art :

• Audio:

There is background music, and a sound effect attached to collectables.

• Lights:

A picture containing tree, outdoor, grass, sport

Description automatically generated

As well as the main direction light in each level, pick ups give off light to help the player identify them.

• Cameras:

When a level is loaded, the camera begins in a higher up top down view. The camera then swiftly moves into the right position to play the game.



• Particle Effects:

A picture containing tree, outdoor, grass, sport

Description automatically generated

The ‘Big Data’ collectables has particles that swirl around it as they fall to the ground. An enemy puffs out smoke as they go.

• Materials:

This is an example of a use of materials in the game. This material has metallic property, so it reflects light more (as opposed to being diffused). A normal map is used to augment the lighting on the objects to make them not look flat and look used. This creates more visual interest as well.



4. AI artefacts :

A state machine, with a steering behaviour and involving use of NavMesh is present.

• Player Animation:

A person with a football ball on a field

Description automatically generated

Other Features:

Tutorial level added that tells player what to do to traverse the whole level.

A picture containing sky, ground, man, skating

Description automatically generated

I drew some assets myself:



How To Play:

Move using WASD (recommended, arrow keys also work). Input is relative to camera, and camera moves to face in direction of the player, so press down, for example, makes the player move towards the camera, so you run in a small circle. Space is jump. You get more height out of a jump while running (height relative to speed). Press c to crouch. Press jump while crouching and stationary to do a high jump (more height than any normal jump). Press crouch and immediately after jump – or alternatively, you can crouch, hold a direction, then jump – and you will do a long jump (more horizontal distance than normal jump). Press jump while facing and touching a wall and do a wall jump (get more height than other jumps). – you get sent in the opposite direction to the way you were facing i.e. away from the wall. Enemies may reduce your health if you bump into them. Re-gain health by collection batteries. Explore to find ‘data’. Press P to pause. Select a ‘Planet’ from the dropdown and press the button to go there. Press ‘Quit to Main Menu’ to go back to main/options menu. **Do this to save the game**.

The Game scenes are in Assets/Scenes and are the following scenes only: MainMenu, Grass, Metal, Tutorial.

Please note: The whole of Standard Assets is in the project files, so there is a lot of stuff that is entirely unused, but I didn’t get rid of in case it broke something.

Assets Used:

Unity Asset Store:

“Standard Assets”, by Unity Technologies (<https://assetstore.unity.com/packages/essentials/asset-packs/standard-assets-32351>)

[Scripts used from Standard Assets: ThirdPersonCharacter – modified for my specific use, ThirdPersonUserControl, AutoCam, ProtectCameraFromWallClip]

“Space Robot Kyle”, by Unity Technologies (<https://assetstore.unity.com/packages/3d/characters/robots/space-robot-kyle-4696>)

“Sound FX - Retro Pack”, by Zero Rare (Levi Moore)

(https://assetstore.unity.com/packages/audio/sound-fx/sound-fx-retro-pack-121743)

Creative commons:

“Seamless Metal Texture” by hhh316 (<https://www.deviantart.com/hhh316/art/Seamless-metal-texture-182943398>)

“(Tsander) Large Impact Crater, Lunar Surface“ by Earth Observatory/Nasa (Image from: <https://upload.wikimedia.org/wikipedia/commons/b/bc/%28Tsander%29_Large_Impact_Crater%2C_Lunar_Surface.jpg>, Source: <https://earthobservatory.nasa.gov/images/39769/fresh-craters-on-the-moon-and-earth>)

“Wood Texture”, by JasonLab (<https://jasonlab.itch.io/seamless-real-texture-pack>)

Music:

"Deliberate Thought" by Kevin MacLeod (https://incompetech.com)