

Front page of the submission reports

Module Code: CS3VR16

Assignment report Title: VR Group World (Individual)

Date (when the work completed): 06/12/2022

Actual hrs spent for the assignment: 28

Assignment evaluation (3 key points):

- Learn more about Unity during this project.
- Communication and sharing resources were essential for development.
- In terms of coding, there was no class resources provided to help with the project.

Introduction

During this project I first delegated myself and my team members their own specific area to work on build from the ground up, using any modelling software or assets needed. To get to this point I had created a mind map of all topics the group wished to pursue and create for the VR project, and a vote was placed for the result, which was an Animal based educational game. By splitting up the general idea into smaller manageable tasks for the group starting with modelling then working our way from there, each member was given a specific habitat to build and design, all tasks were reported with a Kanban board which was used to clearly show progress and completion of a task with evidence. The area in which I had mainly worked on was the Middle central area – the spawn – and the Jungle area also known as “Jungle Rumble”.

Design and Implementation

Modelling

First, I had gathered information onto the habitat I was assigned which was the Jungle, which included animals such as monkeys and gorillas, the habitat was designed based on realistic attributes such as the surroundings being cluttered with trees and grass, as monkeys reside in areas encapsulated in greenery and trees. All models for the sections I had designed and modelled were constructed with Blender – a 3D modelling software. Blender was easy to use to create both simple and complex objects using various techniques and in-built functions to achieve the models I have created. Such as the trees used various points which are all connected to one another to create a branch like structure, blender has a “Skin Modifier” which layers a cube like structure to the points and the connected edges. There are trees scattered across the Jungle habitat to replicate a monkey’s natural aesthetic. The grass was created in a special way by using blenders “Particle System”, this allowed me to create objects looking like grass and procedurally replicate thousands of copies across a plane – a flat surface. Blender’s functionality improves efficiency as it would be tedious to duplicate an object thousands of times repeatedly, this function procedurally duplicates and places objects at random across the plane creates a grass like texture.

As well as the jungle habitat and the animals, the middle area which is the spawn was created in Blender, each “Arch way” leading to the different habitats and distinguishing them by a dedicated colour. Centre of this structure is where the player spawns from.



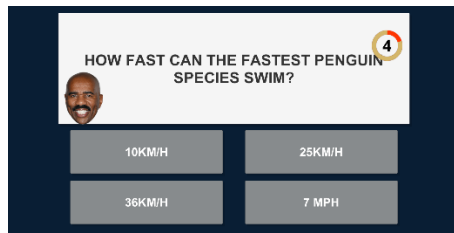
Modelling Animals



Each member was issued with the task of creating or obtaining one 3D animal, I had chosen to create a Penguin. This model was created using Blender. Each body part is a separate object merged into one whole object to create a believable Penguin model. The model shares the same unique assets as a realistic counterpart. This model clearly illustrates a Penguin as the beak, fins, webbed feet along with the colours show that the model is a Penguin.

Functionality

Functionalities which I had delegated to myself where the implementation of; a quiz, collision detection with sounds, destroyable objects upon collision, as well as helping group members with tackling errors and bugs.



Quiz

Quiz

The quiz idea was based off game shows such as “Who wants to be a millionaire?”, as the quiz is a basic game asking the user Questions related to the information illustrated in the game, where each habitat expressing information about the different kinds of animals and their unique traits. The quiz is loaded and interactable upon the completion of the checklist, when the user can play the quiz, they will be presented with a series of questions and must answer them within a time limit, this tests the user’s knowledge gained from the experience gathered from the game. The more correct answers the user gets the better the overall score, the higher the score correlates to a greater understanding in the animal genre. The quiz plays a key role in the education aspect of the game, as it makes the player focus and retrace all the information and details shown in the game, thus checking if the user was truly focused on the topic at hand.

The quiz has a rather simple design however this was intended, as the questionnaire UI has a universal understanding of where the question is presented and where the answers are located. As well as the timer clearing illustrating both the seconds remaining and the ticking counter down with numbers and an image.

Interactable Objects

The objects were modelled in blender and exported into Unity, the object was then created into a prefab which allowed me to create multiple duplicates of an object with the same Unity modifiers such as scripts, rigid body, mesh collider, etc. The interactable object has a script component embedded, whenever the player collided with the object, the object will behave in a specific way, in this case the object would be destroyed, therefore removed from the game. This gives the illusion/believability with a simple function of picking up a banana.



Interactable Objects



Collision and Sounds

Collision and Sounds

I had implemented an easy-to-use prefab which was able to be parented upon any object, in this case the objects would be the animals. As upon colliding with the collider the object would play a specific sound related to the animal. In the image above the illustration shows the two models showing sounds viewed from the objects which shows that there is a sound source connected to the object, when the player interacts/collides with these objects an animal sound associated will play. This function creates a more natural feeling to the atmosphere as animals make sounds therefore improving quality of life features and believability.

Contribution to Group Members Scripts

I had the authority to help my fellow group members with fixing and debugging their scripts as issues would arise and I would lend my help to efficiently fix errors and bugs. Such as the Sign script, the animation being called was not correctly being called as the method of retrieving the animator and its child was wrong, this was not working well with other aspects in the code such as the use of "SubCoroutine's" which help when dealing with animations and timers. I had also helped with the UI system for the checklist, by cleaning the code and making the code more efficient and readable to other members, as well as myself.

Testing and Results

Importing/Exporting

There were common issues when importing and exporting objects to and from Blender and Unity, there were issues with orientation, issues with textures and colours, and issues with collisions. When facing these problems, I would slowly recreate situations to test why the object behaved in an illogical manner, such as not having mesh colliders in Unity therefore the player could phase through the object. Or missing textures when importing from Blender to Unity as the blender texture system doesn't export appropriately to Unity, therefore workarounds were found such as, "Baking" the textures from the Blender node system and creating image files with height maps, occlusion levels etc. These files were necessary to recreate the texture in Blender to Unity. Another workaround was to create an image texture with colour swatches, and by UV unwrapping the objects in Blender and linking these flat object faces to the corresponding colours the colours would be exported to Unity without any issues, as Unity would automatically know which texture file is linked to that texture, as without this some colours would look bland and dark, which didn't mix well with Unity lighting.

Debugging

Debugging was necessary when handling both Unity issues as well as scripting issues, the console was vital in locating syntactical errors, which led to quick fixes in the code due to the debugging providing the location to the errors and the IDE applying some form of handle tips on solving the said issue. However logical errors were fixed with another method by printing "Debug.Log()" at various parts of code which was used to check if the corresponding functions were called correctly and work appropriately, the "Debug.Log()" would present text in the Unity console which helped majorly when it came to fixing logical errors, as it clearly showed me if interactions were working correctly.

Reflection and Further work

Animation

Due to lack of understanding how to animate models in Blender or Unity, there weren't any animated objects in the game, as all objects in the game were created from scratch by each team member. Animations would bring the game more to life and create a realistic environment, thus giving life to the animal's objects and improving the believability, as the player would have the impression of walking with the animals and getting a close first-person glimpse of how animals behave and move. The animations would also make the world more exciting as the world currently has little to no animations this making the world a bit dull due having many stationary objects being suspended in the scene.

Particle Effects

Unity has an expansive and brilliant particle system which was not taken advantage off in this project, the particle effects could be used in many areas of the world, such as water bubbles for the fishes in the fish tank, or snow effects falling from the ice habitat. I regret not being able to test and use the Unity particle system as it would add more unique aspects to the world and bring the world to a more eye-catching experience. Like the animations the particle effect would let the player visually see movement and updates in real time, as there are not many objects which show any form of moveability.

Contributions

I had shown many contributions to this project from 3D modelling, team management, scripting as well as information gathering. The unnecessarily hurdles our team had to go through were very stressful as each member had their own case of troubles, however as a team and individuals we had to stick it through to the end and delegate more tasks to each member as well as show results at a faster pace. Certain group member was a liability due to there constant inactivity and lack of work shown throughout the course of the project, as a collective the group had no choice but to share the tasks between each other and complete the project with minimal to no help from our member. My contributions to the team were numerous as I was more confident in my ability to understand Unity and Blender I had shown how to implement and code functionalities of the code which made understanding the implementation much easier for the other members, this also sped up development time.

Research Material for Jungle & Questionnaire

Monkey Information

Monkey Habitat

- Tend to live in forest regions, with lots of trees around them.
- Monkeys tend to move around continuously, therefore their nest isn't sustainable for long periods of time.

Monkey Reproduction

- The reproduction process can be full of rituals for the Monkey. Those rituals will depend on the species though. They generally will become mature between 4 and 5 years of age. The females are attracted to the biggest and strongest males

Monkey Predators

- Monkeys are very small in size and stature therefore they are prone to many predators. Common predators are large birds, big cats, hyenas.

Monkey Social Setting

- A group of monkeys is referred to as a troop, or mission. A troop consists of individual monkeys that live and travel together in search of food, also providing and taking care of young monkeys.

Monkey Facts

- Rhesus monkey was the first primate in space. Rhesus monkey named Albert, was sent into space before humans, brought back unharmed.
- Monkeys sleep while sitting on trees.
- Myanmar species sneezes when it rains. Due to the shape of their long nose and rain falling into them causing them to sneeze.
- Monkeys with the loudest calls are the Howler monkeys. They do this to protect their territory and alert them to stay away.
- Largest monkey in the world is the Mandrill monkey.

Monkey Questions

- Monkeys with the loudest call?
 - Howler Monkey (X)
 - Rhesus Monkey
 - Myanmar Monkey
 - Mandrill Monkey
- What is the largest monkey in the world?
 - Howler Monkey
 - Rhesus Monkey
 - Myanmar Monkey
 - Mandrill Monkey (X)
- First primate to reach space?
 - Howler Monkey
 - Rhesus Monkey (X)
 - Myanmar Monkey
 - Mandrill Monkey
- What is a group of monkeys called?
 - School
 - Mission (X)
 - Clowder
 - Waddle
- Which is not a common monkey predator?
 - Hawk
 - Jaguar
 - Hyena
 - Python (X)
- What ages does a monkey become fully mature?
 - 3-4 years old
 - 4-5 years old (X)
 - 5-6 years old
 - 6-7 years old

Appendix

Quiz

QnA.cs

[System.Serializable]

```
public class Questions_Answers{
    public string Question;
    public string[] Answers;
    public int CorrectAnswer;
}
```

AnswerScript.cs

```
public class AnswerScript : MonoBehaviour
{
    public bool isCorrect = false;
    public QuizManager quizManager;
    public void Answer()
    {
        if (isCorrect)
        {
            Debug.Log("Correct Answer");
            quizManager.correct();
        }
        else
        {
            Debug.Log("Wrong Answer");
            quizManager.wrong();
        }
    }
}
```

QuizManager.cs

```
public class QuizManager : MonoBehaviour
{
    // Quiz
    public List<Questions_Answers> QnA;
    public GameObject[] options; // Buttons
    public int currentQuestion;

    public Text QuestionText;

    // Panels
    public GameObject QuizPanel;
    public GameObject GameOverPanel;

    // Game Over
    public Text scoreText;
    private int totalQuestions = 0;
    public int score;

    // Timer
    [SerializeField] private Image uiFill;
    [SerializeField] private Text uiText;

    public int Duration;
    private int remainingDuration;
```



```

private void Start()
{
    totalQuestions = QnA.Count;
    GameOverPanel.SetActive(false);
    generateQuestion();
}

private void generateQuestion()
{
    if (QnA.Count > 0)
    {
        remainingDuration = Duration;
        StopAllCoroutines();
        Being(Duration);

        currentQuestion = UnityEngine.Random.Range(0, QnA.Count);

        QuestionText.text = QnA[currentQuestion].Question;

        setAnswer();
    }
    else
    {
        Debug.Log("OUT OF QUESTIONS!");
        Duration = 0;
        gameOver();
    }
}

private void setAnswer()
{
    // Loops through each button and assigns the text.
    for (int i = 0; i < options.Length; i++)
    {
        // All buttons will be assigned with a false bool value.
        options[i].GetComponent<AnswerScript>().isCorrect = false;
        options[i].transform.GetChild(0).GetComponent<Text>().text =
QnA[currentQuestion].Answers[i];

        if (QnA[currentQuestion].CorrectAnswer == i+1)
        {
            // Assigning a true bool to the correct answer.
            options[i].GetComponent<AnswerScript>().isCorrect = true;
        }
    }
}

public void correct()
{
    score ++;
    // Removes the previous/current question.
    QnA.RemoveAt(currentQuestion);
    generateQuestion();
}

public void wrong()
{
    QnA.RemoveAt(currentQuestion);
    generateQuestion();
}

```

```

    }

    void gameOver()
    {
        // Disables quiz panel & enables game over panel.
        QuizPanel.SetActive(false);
        GameOverPanel.SetActive(true);
        scoreText.text = score + " / " + totalQuestions;
    }

    public void retry()
    {
        SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex);
    }

    public void goBack()
    {
        SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex - 1);
    }

    // Timer
    public void Being(int Second)
    {
        remainingDuration = Second;
        StartCoroutine(UpdateTimer());
    }

    private IEnumerator UpdateTimer()
    {
        while (remainingDuration >= 0)
        {
            uiText.text = $"{remainingDuration}";
            uiFill.fillAmount = Mathf.InverseLerp(0, Duration,
remainingDuration);
            remainingDuration--;
            yield return new WaitForSeconds(1f);
        }
        onEnd();
    }

    public void onEnd()
    {
        Debug.Log("TIMER ENDED!");
        wrong();
    }
}

```

Collision with Sound

Collision.cs

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Collision : MonoBehaviour
{
    public AudioSource audioSource;

    private void OnTriggerEnter(UnityEngine.Collider other)
    {

```

```

    {
        if (other.gameObject.tag == "Player" && this.gameObject.tag ==
"Pickup")
        {
            Destroy(this.gameObject);
        }
        else if (other.gameObject.tag == "Player")
        {
            audioSource.Play();
            Debug.Log("Entered");
        }
    }

    private void OnTriggerExit(UnityEngine.Collider other)
    {
        if (other.gameObject.tag == "Player")
        {
            audioSource.Stop();
            Debug.Log("Exit");
        }
    }

    private void OnCollisionEnter(UnityEngine.Collision other)
    {
        if (other.gameObject.tag == "Player")
        {
            audioSource.Play();
            Debug.Log("Entered");
            //Destroy(collision.gameObject);
        }
    }
}

```

Interactable Object

Pickup.cs

```

public class Pickup : MonoBehaviour
{
    private void OnCollisionEnter(UnityEngine.Collision other)
    {
        if (other.gameObject.tag == "Player"){
            Destroy(gameObject);
            Debug.Log("Banana");
        }
    }
}

```