INM717 Milestone 3
VR Cultural Experience
Report – Development Focus

Student Name: Li Guo

Student Number: 220060950

Date Submitted: 15 May 2024

Contents

Section 1 – Designing for VR

Design Principles for Virtual Reality – (e.g. Viewing Angles / Depth)

The virtual reality experience, or the immersive experience, is a core concept of how to experience around the player, and the immersion through the scene is to associate the scenes with actions. In this VR cultural experience, the action is to move the player around the scene, grabbing the basketball and shooting the basketball. The viewing angles should make the player feel comfortable while playing, in this game, player can change the view angle with the headset on the head while moving the head, and the movement of the character is smooth enough to make player feel comfortable while moving the character around in the game scene.

Consideration of the Three Illusions of Virtual Reality.

The place illusion is the basketball court in the scene, in the basketball court, there are two basketball stands, few basketballs, few basketball containers to put basketball on, and few NPCs on the chairs in the side court to react while the baskets have been made.

The plausibility illusion is to add proper physic to the basketball and bouncing affects to the basketball to make the bouncing effect more realistic.

The embodiment illusion is the realistic basketball, the realistic basketball court and the basketball net in the scene.

Accessibility \ Avoiding Adverse Health Effects

The usages for avoiding adverse health effects are safety, hygiene, sickness, adaptation, and readaptation. The importance of avoiding sickness is to not present visuals while the user is putting on or taking off the HMD and not stop the application and switch to the desktop without having the screen go blank or closing the eyes while playing.

Embodiment \ Presence.

The player is controlling the NBA player and joining the NBA three-point contest in the all-star contest, player can grab the basketball with both controllers and shooting the basketball only using the right controller. Player can shoot the basketball to the rim to score, after all the basketball are shot, the player with highest score will win the three-point contest champion cup.

Section 2 – Implementation of Technical Requirements

Description of your Prototype

Concept

In this VR cultural experience, player will experience the three-point contest by moving the character around the NBA Huston Rocket basketball court and grab the basketball to shoot it to the rim to score.

Controls

The controls of this VR cultural experience including moving the character around with WASD, grabbing the basketball with G, and shooting the basketball with left button on the mouse. The player running function is not implanted in the game, but it should be the SHIFT button on keyboard. The following table 1 shows the controls of this VR cultural experience.

W	Moving forward
S	Moving backward
A	Moving left
D	Moving right
G	Grab the ball
Left button on mouse (only for the right controller)	Shoot the ball

Table 1 controls of the game

What should players do within the prototype?

After entering the game scene, player can start moving the character around to explore the scene, which is the NBA Huston Rocket basketball court. Then player can grab the basketball by pressing the G button on keyboard and move the basketball around while holding the basketball. By the time that player is holding the basketball, player can use the right controller to shoot the basketball toward the basketball rim with the scoreboard on to score the basket by pressing the top button on the controller or the left button on the mouse. When the basketball has collision with the net, the basketball with be destroyed and the score will be counted. Player will try to shoot all the basketball to the rim to get as higher score as possible.

Technical Requirements Implemented

Movement / Interactivity

What did you implement?

For the movement of the cultural experience, I am implementing the XR original from the XR interaction toolkit into the scene, using the W, A, S, and D button to move around the character in the scene and the G button to grab the objects in the scene. I also add a new function to the object that using left button on the mouse or the top button on the controller to shoot the object while grabbing the object.

Where in the demo did you implement it?

For the moving function and the grabbing function, I implement them in the locomotion system in the XR origin showing in the Figure 1. For the shooting function, I add a new script called shoot to the basketball object showing in the Figure 2.

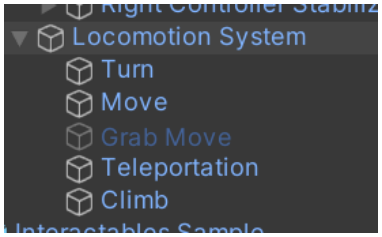

Figure 1 movement and grab function implementing position

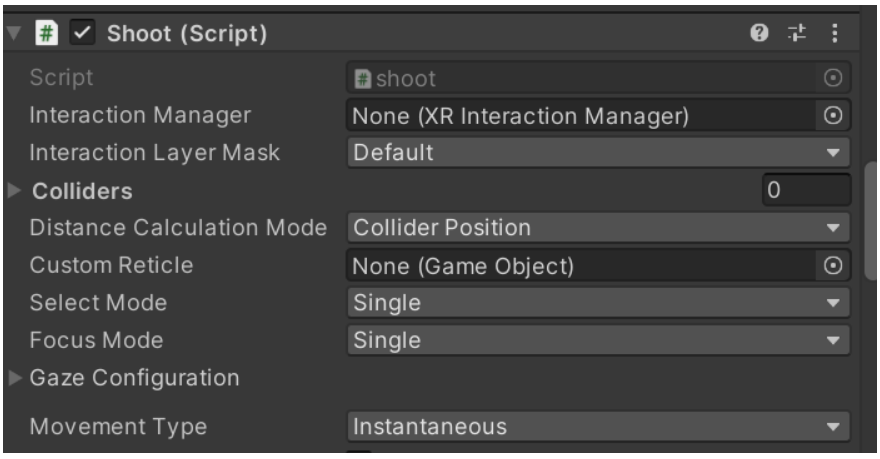

Figure 2 grabbing function implementing position

How did you implement it?

For the moving function and the grabbing function, I implement the XR interaction toolkit from the project settings and then add the XR interaction setup to the game scene, showing in the Figure 3.

For the shooting function, I add the shoot script to the objects with shooting function activated. As shown in the figure 4, in the shoot script, I first give the interactor and the objects that which object needs to interact with and give the force to the object so the object can have the initial force to start with. In the ShootBall

function in the script, I first set the object which is grabbed to the status of not able to be interact with to force dropping the object while grabbing, after dropping the object, the status of interaction will set to true again to set the status of the object to original status. Then the rigid body of object is get to detect if this object has the rigid body, if this object has rigid body, then using the AddForce function to add direction and BallShootForce parameter defined before to give the initial force to the ball shooting movement. And finally in the interactable events, add the ShootBall function to the Activated function, so when the left mouse button is pressed, this function can be activated to shoot the grabbed object.

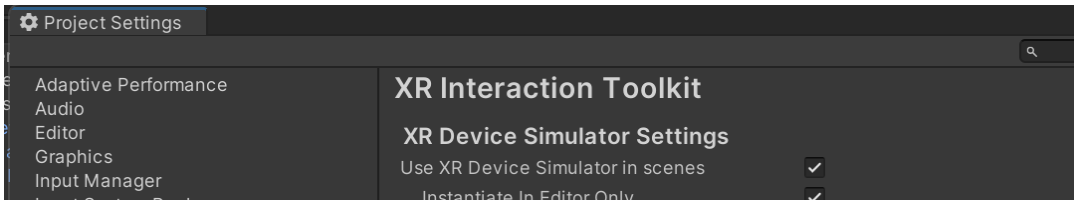

Figure 3 way to implement interaction toolkit

```
Unity 脚本(5 个资产引用)|1 个引用
public class Shoot : XRGrabInteractable
{

    public XRBaseInteractable grabbedObject;
    // The Direct Interactor
    public XRBaseInteractor thisInteractor;

    [SerializeField]
    public float BallShootForce;



    0 个引用
    public void ShootBall()
    {
        // force to drop the ball from interactor
        grabbedObject = GetComponent<XRGrabInteractable>();
        if (grabbedObject != null && grabbedObject.isSelected)
        {
            grabbedObject.enabled = false;
        }

        grabbedObject.enabled = true;

        Rigidbody rb = grabbedObject.GetComponent<Rigidbody>();
        if (rb != null)
        {
            // Calculate shoot direction (in this example, it's the forward direction of
            Vector3 shootDirection = thisInteractor.transform.forward;

            // Apply force to shoot the object
            rb.AddForce(shootDirection * BallShootForce);
        }

    }
```

Figure 4 code for shoot script

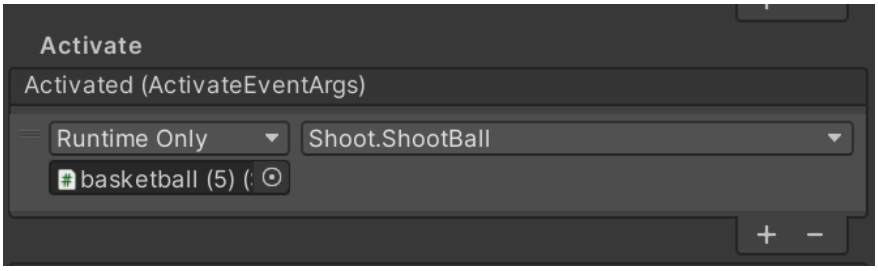

Figure 5 interactable events

Environment

What did you implement?

For the environment in this game scene, I implement the NBA Huston Rocket basketball court model from sketchfab, the lighting from the original sunlight, the text from the TextMeshPro, and the environmental storytelling from the scoreboard script added to the UI and the score script added to the basketball objects.

Where in the demo did you implement it?

For the NBA Huston Rocket basketball court model, I add the model into middle of the game scene. For the lighting of the scene, I use the original sunlight and put it at the top of the basketball court. And finally for the scoreboard text UI, I add it in the middle of the basketball board, shown in Figure 7.

Figure 6 text UI for scoreboard

How did you implement it?

For the basketball court model, I just drag and drop the model from the file and put it in the middle of the scene, and same for the camera, I just drag the camera to the proper position in the basketball court.

For the scoreboard UI text, I implement the Score script to all the basketballs and the scoreboard script to the text UI shown in Figure 9 and 10. The logic for the score script is that every time the basketball colliding with the net, the score can be added to the scoreboard. So the tag called Score is added to the net, and when the basketball collides with net, the Basket parameter in scoreboard component will add 1, and then destroy this basketball after collision. In the scoreboard script, it simply print out the Buskets parameters to the score can be seen on text UI.

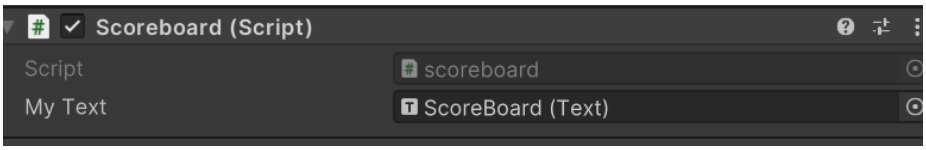

Figure 7 scoreboard script component in text UI

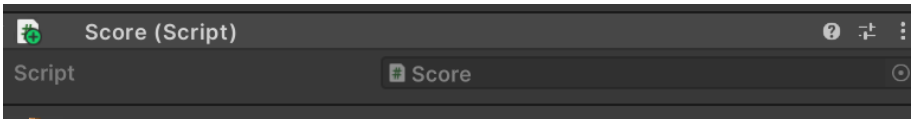

Figure 8 Score script component in basketball objects

```
public class Score : MonoBehaviour
{
    Unity 消息 | 0 个引用
    private void OnCollisionEnter(Collision collision)
    {
        if(collision.gameObject.tag == "Score")
        {
            print("fucking score");
            scoreboard.Baskets += 1;
            Destroy(gameObject);
        }
    }

}
```

Figure 9 score script

```
Unity 脚本(1 个资产引用) | 1 个引用
public class scoreboard : MonoBehaviour
{

    public static int Baskets = 0;
    // Start is called before the first frame

    public Text myText;

    // Use this for initialization
    Unity 消息 | 0 个引用
    void Start()
    {

        myText.text = "";

    }


    // Update is called once per frame
    Unity 消息 | 0 个引用
    void Update()
    {

        myText.text = "" + Baskets;

    }
}
```

Figure 10 scoreboard script

Immersion

What did you implement?

For the Immersion, I have implemented the sound of basketball bouncing from https://pixabay.com/sound-effects/search/basketball/, the player hands function, which the further function implementation is not successful based on the implementation of player hands, so I implement the player hands in the prefab scene not in the prototype scene to make sure the current functions can work properly in the prototype scene.

Where in the demo did you implement it?

For the bouncing sound of the basketball, I add the sound source into the audio source in the basketball objects, shown in Figure 6. For the player hands, I implement the XR hands in the XR original under the Camera Offset.

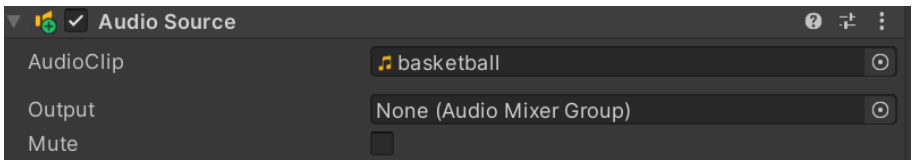

Figure 11 audio source for basketball objects

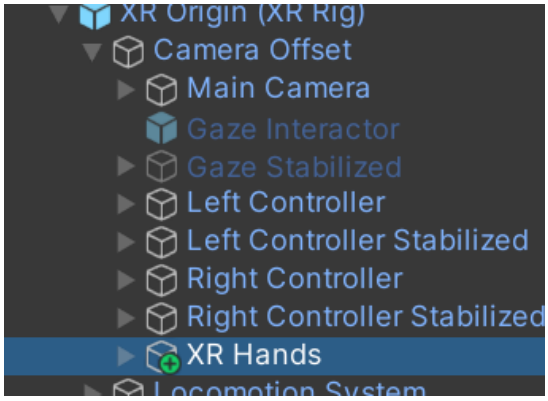

Figure 12 XR hands implementation position

How did you implement it?

For the sound of basketball bouncing, I create the BounceSound script and attach it to the basketball, also add the audio source component to the basketball and drag and drop the sound source to the audio clip in audio source component, which is shown in Figure 6. As shown in Figure 7, in the BounceSound script, the audio clip will play while the object colliding with the other object with the tag called “BounceSound”. And I add the “BounceSound” tag to all the other objects in the game scene to make sure the bounce sound will play when the basketball collides with them.

For the player hands, I implement the XR hands from the package manager, and add the XR hands to the camera offset under the XR origin. In the game scene, by pressing H button on keyboard, the player controller will change to player hand. Figure 14 shows the player hands in the game scene.

```
public class BounceSound : MonoBehaviour
{
    public AudioSource BounceSource;


    Unity 消息|0 个引用
    public void OnCollisionEnter(Collision collision)
    {
        if(collision.gameObject.tag == "BounceSound")
        {
            print("fucking bouncing");
            BounceSource.Play();

        }
    }
```

Figure 13 BounceSound script

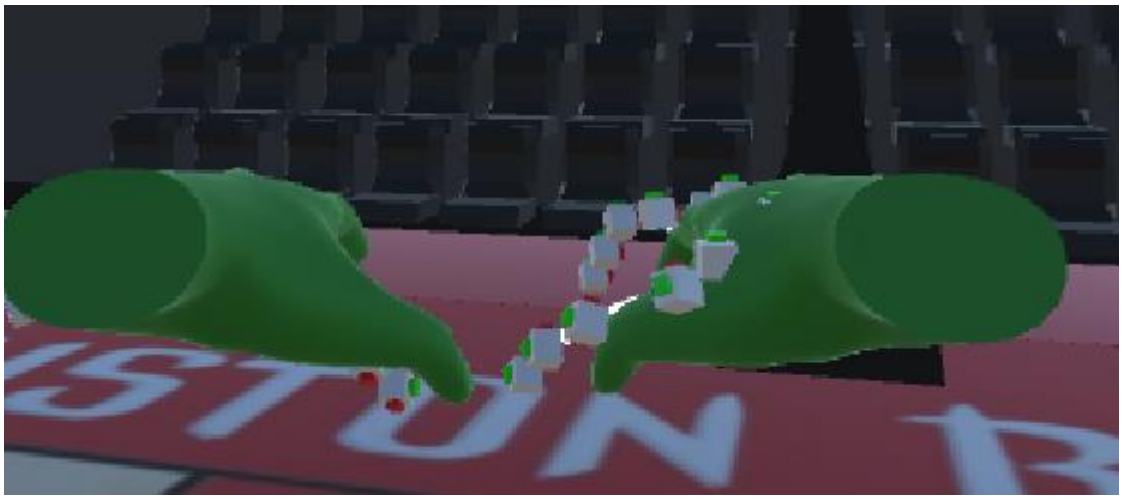

Figure 14 player hands

Gameplay

The gameplay in this game is to grab the basketball on the container and shoot the basketball to the basketball rim and make the score. For the grabbing function and shooting function mentioned above, the following figures show the grabbing ball, shooting ball and scoring situations.

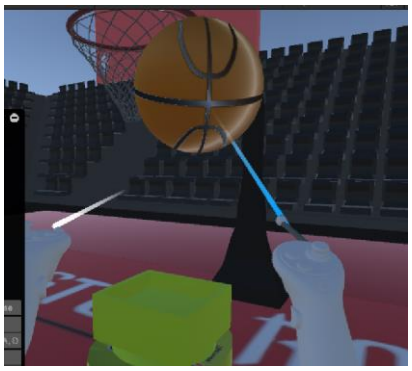

Figure 15 grabbing ball

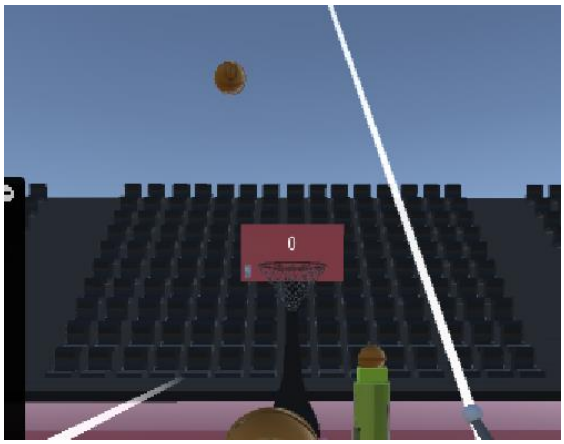

Figure 16 shooting ball

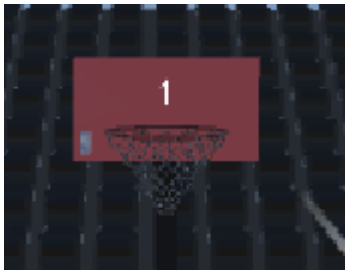

Figure 17 scoring

Optimisation

The assets in this cultural experience are the basketballs, the basketball court, and the containers to put basketballs on.

All the assets are the prefab models (Figure 18) are modified in the prefab scene that can simply drag and drop into the game scene, and put them in the proper position, the following figures show the example for the assets in the game scene.

figure 18 prefabs for the assets

Figure 19 round container with basketball

Figure 20 big container with 5 basketballs

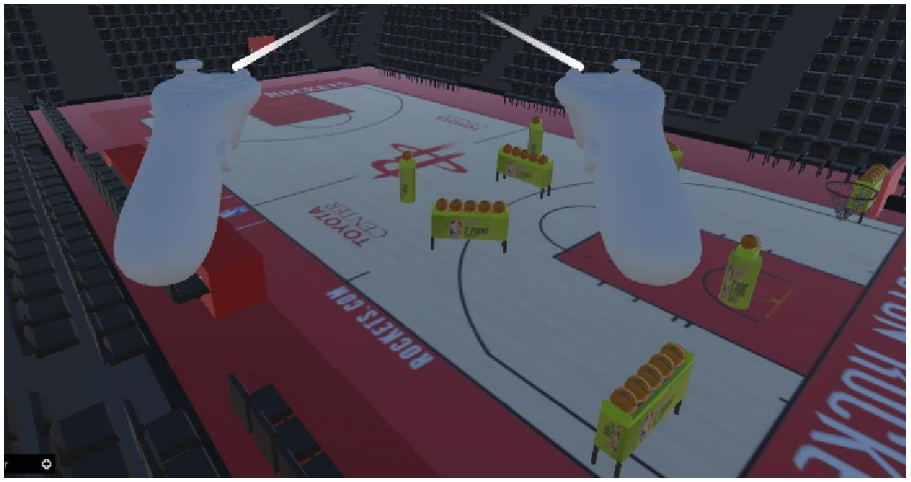

Figure 21 full basketball court

Section 3 – Reflection and Further Work

How was the process of designing and building your VR Cultural Experience?

In the scene, the most challenging function is to force the basketball to drop and add force to the basketball. At the beginning, the function to drop the basketball is not successful by adding a ForceToDrop function using OnSelectExit funciton in the XRBaseInteractable file but based on the version of the XR interaction toolkit, this function is not available to use. Then I find the way to disable the object status so the object can be dropped.

There would be more functions in the scene to make the experience more interesting, for example, make the player can run in the scene. I am being over ambitious by trying to make the shooting ball function by trying to use the hand to shoot the basketball by detecting the hand motion from real hand to grab and throw the basketball.

What would I do to improve my design?

If I have more time, the game experience can be improved with more NPCs in the game scene that can join the three-point contest that make the game experience more thrilling. Also adding more storytelling to the scene, for example, adding more rounds for the players to contest with other players. Another improvement is to add a function that while the player pressing the shooting button, the force to the initial ball shooting force will increase with certain force increase speed, so the ball shooting motion will be more realistic.

What would I do differently on future VR projects?

In the future VR projects, there should be a full court NBA basketball contest, which there are 10 players on the court, with the main character and 9 NPCs, player can choose which basketball tactic to use while in different offending or defending rounds, call the timeout while playing and listen to the coach and teammates talking while in or off the court . Player can pump into other NPCs and the NPCs will reflect to the position of the pumping happens. The ball shooting motion will be more complicated that ball shooting force depends on the speed of the controller is rotated by the player.

Links to Downloads

Link to Download your Unity Project

https://www.dropbox.com/scl/fi/93vdwcv08xq3bp1emmkp2/INM717-Milestone-3-Li-Guo-Unity-Project.zip?rlkey=lcu85w4rcbvmrvxgy5kba9pls&st=xilfvs9q&dl=0

Link to Download a Build of your Unity Project

https://www.dropbox.com/scl/fi/rys0u9p7kghs5kpxk7ue9/Three-Pont-Contest.zip?rlkey=63of9w0lw7gdoytj9izg9fpd3&st=fzczrkkd&dl=0

References:

Jason, J. *The VR Book : Human-Centered Design for Virtual Reality*, Association for Computing Machinery, 2015. *ProQuest Ebook Central*, Available at: http://ebookcentral.proquest.com/lib/city/detail.action?docID=6952709.