2019 International Conference On Business Innovation (ICOBI), 22 November, Colombo, Sri Lanka

3D Walk-through at NSBM

D Manujaya
Faculty of Computing
NSBM Green University Town
Homagama, Sri Lanka
dulan69@gmail.com

D J Jayamanne
Faculty of Computing
NSBM Green University Town
Homagama, Sri Lanka
dileepa@nsbm.lk

Abstract— **In this work, a 3D Virtual Walk-through of NSBM using the Unity Game Engine is presented. The walk-through includes real life and detailed 3D models of buildings and objects. The buildings are modelled using SketchUp including the interiors. All buildings and objects are approximately modelled to scale. Objects are also modelled using SketchUp where possible. All other models which require more details are modelled using 3Ds Max and Blender. As the walk-through is started, a user is dropped in front of a building. Then, the user can control the camera using the mouse. The arrow keys are used to move forward, back, left and right in the virtual environment. Also, the user is able to navigate inside the buildings, walk into rooms and open doors. The light source above the ground that imitates the sun provides lighting and shadows for the buildings and trees. There are separate light sources inside each room to model the actual lighting conditions we observe in real life. All lighting is handled using the Unity Engine. In this work, the user has the freedom to move the camera wherever the user desires in the given area as there are no fixed cameras, predefined paths, and environmental hotspots. However, the user is able access any part of the model if it is physically possible.**

Keywords— 3D Walk-through, Unity Game Engine, SketchUp

I. Introduction

There are countless virtual tour platforms and 3D visualizations to be found on the internet [1], [2] and [3]. None of these programs/projects provide a completely immersive experience. Most of them are video walk-throughs of pre-rendered models with a fixed camera moving on a predefined path. Some lets the user navigate the environment using the mouse. The user points and clicks on the desired destination. The user will then be moved to the hotspot closest to the point where the user clicked.

In this work, we propose a 3D Walk-through at NSBM. The Walk-through will include buildings that are modelled approximately to scale. The user is able to virtually walk at NSBM without having to physically be at the location. This type of an application is beneficial to any stakeholder who has an interest in NSBM to get an idea about the place prior to the actual visit. There are several environments that feature a different time of the day with lighting to match the environment. The included Graphical User Interface (GUI)/Menus provide the user with an intuitive method to select and navigate between different environments and levels.

Literature survey

In the real world, there are many experiments that cannot be run on a system, simply because they are either expensive or impractical to run. The effective way to test these scenarios on a system would be to run simulations. They will provide users with analytical and practical feedback based on the simulation which is simulating a real-life scenario. Architecture Visualizations is the graphical representation of a model [4] used to simulate. It can be represented either physically or digitally. It is mainly used to showcase the attributes and practical application of principal of content of a proposed building. For example, Architectural Visualization includes design sketches and blueprints.

A game engine is the software provides the necessary tools and set of features required by the users to build simulations quickly and efficiently [5]. Game engines provide the necessary framework to bring together several core aspects of game development: Graphics; rendering engine used to generate 3D/2D graphics, Audio; audio engine which contains algorithms related to sound and audio, Networking; Application Programming Interfaces (APIs) which allows players to connect with each other in a network but not used in this work, Physics; engine that lets the game implement the physical laws of nature, Graphical User Interface (GUI); tools to design a 2D interface which the user will interact with, Scripting; scripts that holds all the logic in the game. Out of the 3 main Game Engines available for public namely Unity, Unreal and Game Maker Studio, the Unity Game Engine is used in this work since it is the easiest to work with and has a very simple user interface which is more user friendly for anyone who wants to dabble in the arts of game development.

In this work, a 3D Walk-through of NSBM is proposed. This allows a person to virtually be at a specific location, move around and experience the feeling of being physically present at the location. The proposed Walk-through includes buildings that are modeled close to scale as possible. The 3D models of the buildings are built using SketchUp. The player is able to move and look around the buildings virtually. The player can control a humanoid 3D model called Ellen. Ellen contains all the necessary cameras and audio sources required for the game to properly function. The player is able to control Ellen using the keyboard and the mouse. The model contains openable doors with animations done in real time. Every Game Object obeys the laws of physics in the system, meaning that movement and collisions are realistic. There are several environments for the user to choose and play on. Each environment features a different time of the day with lighting to match the environment. The included GUI/Menus will provide the player with an intuitive method to select and navigate between different environments and levels. The night environments are unique since they have a different lighting scheme than the reset.

II.METHODOLOGY

Modern game engines can help create games with stunning graphics. They also help import models/assets from other programs and platforms thus eliminating or drastically reducing the waiting time for each import. The game engine used in this project is the Unity Game Engine. Unity is a real-time cross-platform engine and is one of the most popular game engines available in the market.

The 3D modelling tool used in the project was SketchUp, formerly known as Google SketchUp. Sketchup is a 3D modelling program that is being used in a wide range of applications. A tape-measure was used to get the dimensions of the desired building. There was no practical way to measure the heights of the walls accurately since the necessary instruments/equipment was not readily available. These height dimensions were approximated using the average human height and then these were marked on some on-site sketches. The Way-finding map was added to SketchUp and a line was drawn along the wall of the measured section. The length of this line was the same as the dimensions recorded from the tape measure. The photo was then adjusted/scaled so that the measured wall matched the length of the line. The end result was that now there was a base which is almost exactly to scale for the model to be built.

The main character model used in the project is called Ellen. This rigged character model is available in the Asset store. Ellen is the most important Game Object in the entire project. It is the parent object for the two cameras and multiple essential scripts and audio files. Ellen is rendered using the mesh renderer component with the materials and provided with the asset when downloaded from the asset store. To make sure Ellen obeys the laws of physics of the system, the RigidBody component is used. Ellen is controlled using keyboard and mouse and is capable of opening and closing doors with animations done in real time. The Use Gravity option is used for an attached object to move/fall realistically in the scene. Rotation on all axes were locked so that it doesn’t fall off balance when moving or in collisions. Ellen uses a capsule collider since the capsule shape approximates the shape of Ellen’s body better than other collider shapes.

Every scene contains the default main camera and a secondary camera which is used to change view mode. The main camera is used to render the game in first-person view and the other camera is used for third-person view. The cameras are attached to centre of Ellen. Each camera contains an Audio Listener by default. This component is used to listen to sounds in the environment and play it back to the user through speakers or headphones. Ellen has two Audio Sources attached: Walk Sound and Jump Sound. They are played when the specific keys are pressed. Walk Sound is played when W/A/S/D keys are pressed. Jump Sound is played when Space is pressed. This logic is handled by scripts. An Audio Source object with the relevant script attached is made child of Ellen. Once the positions are reset, audio source should move everywhere with the player just like the cameras.

There are 6 different scenes implemented in the project: Morning, Midnight, Daytime, Afternoon, Cloudy and Dusk. Each of these scenes have the same model building and characters.

- Morning: Contains the Morning skybox. Directional light used in the scene has a light yellowish color to match the mood and the sky/environment. Its direction is also set to look like it's just risen so that the object shadows corresponds accordingly.

• Midnight: Contains the Midnight skybox. Directional light is set to dark gray so that it simulates moonlight. The light provided by the skybox is also extremely dim. This scene also contains several other light sources. These were used to replicate light bulbs placed in rooms, stairs and gardens. Together, these light sources provide a realistic night atmosphere that matches the mood and the overall environment.

• Daytime: Contains the Daytime skybox. Directional light is brighter than all the other scenes and the color Is set to pure white as well to brighten up the environment. There are no artificial light sources since it would be unrealistic for light bulbs to be turned on in the middle of the day.

• Afternoon: Contains the Sunset skybox. Directional light used in this scene has an orange color since the color of light in sunset is usually orange in real-life. The direction of the direction light is also rotated so that it looks like light is shining from the horizon at the objects. All of this helps in providing a realistic atmosphere for the scene.

• Cloudy: Contains the Cloudy skybox. This scene is extremely grim and moody. The sky and the dim lighting provide an atmosphere which makes the player feel like a storm is on its way. The color of the directional light is set to gray.

• Dusk: This scene is very similar to a night scene. The dark starry sky brings out the cold, silent atmosphere. The directional light and the light from the skybox are like the lighting in the midnight scene but here it's a bit brighter. The same artificial lights are also used in the exact same positions as the midnight scene.

In this work, the direction and point lights are used. Directional light is a light that represents sunlight. The source position of directional light doesn't matter since the distance of the light is not defined whereas Point light emits light rays in all directions in the specified range. We used Point lights as outdoor lights.

Scripts are an essential aspect of game development. These files contain the core logic of the components and even the game itself. Scripts in Unity are written in C# using Microsoft Visual Studio. Following scripts are used in ths work.

• CameraCycle: Used to change the view-mode from 1st person to 3rd person. Assigns 2 cameras and uses a simple bool variable to switch between the two cameras when a key is pressed.

- CamMouseLook: Enables the player to control the attached camera with the mouse.
- CharacterController: Enables the player to control the attached object using the keyboard. The script is also used to hide and lock the cursor at the start of a scene.
- Headbobber: Simple script which bounces a camera along a fixed midpoint to give realistic headbobbing effect when moving.
- Jump and Walk Sounds: Plays a sound when a specific key is pressed. While these are two different scripts attached to 2 different audio sources, the logic in the scripts are more less the same.
- OpenableDoor: Holds all the logic for opening/closing of doors when a button is pressed. Can be attached to any set of doors to add the openable door functionality.
- JumpScript: Allows the user to jump up when a key is pressed. Provides an upward force to the attached RigidBody and pushes it to up few units.
- PauseMenuScript: Contains all the functions to restart and load environments. It's also the script that hides/shows the attached Panel. This script is attached to an empty GameObject, and then its referenced to from multiple buttons and menus.
- RestartScripts: Restarts the current scene on key press.
- StartMenuScript: Very similar to the PauseMenuScript but this script refers to start menu instead.

III.Results and Analysis

In this work, the proposed 3D Walk-through of include buildings that are close to scale as possible. Fig. 1 and Fig. 2 depict the 3D sketches that are used to build 3D models through SketchUp. The user moves and looks around the buildings as if a player in a simulator. The user controls a humanoid 3D model called Ellen. Ellen contains all the necessary cameras and audio sources required for the game to properly function. The user can control Ellen using the keyboard and the mouse. The model contains openable doors with animations done in real time. There are several environments for the user to choose and play on. Each environment will feature a different time of the day with lighting to match the environment. The included GUI/Menus will provide the player with an intuitive method to select and navigate between different environments and levels. The night environments are unique since they have a different lighting scheme than the reset. This work is built using the Unity Game Engine and all scripts are written C#. Scripts are an essential aspect of game development. These files contain the core logic of the components and even the game itself. Scripts in Unity are written in C# using Microsoft Visual Studio that define the logic in the game. Figures 3 and 4 below show few screen shots of the developed 3D game.

IV.Conclusion

The project will allow any interested stakeholder to have a virtual walk in NSBM without having to physically be at the location. A person will be able to feel the premises and experience the real sensation of actual walk-through. This can also be used as a marketing material to attract more foreign and local potential students who are seeking higher education in Sri Lanka. Furthermore, this platform will continue to promote education at NSBM in foreign communities.

As for future work, we are planning to model the buildings using the official/genuine floor plans owned by NSBM. The floor plans would provide all the necessary details for the 3D model to be completely modeled to scale. Also extending this work to complexly model the outdoor environment of NSBM will complete this project. We are also planning to model the interior of the building with the accurate dimensions and to further allow a user to walk inside the buildings as well.

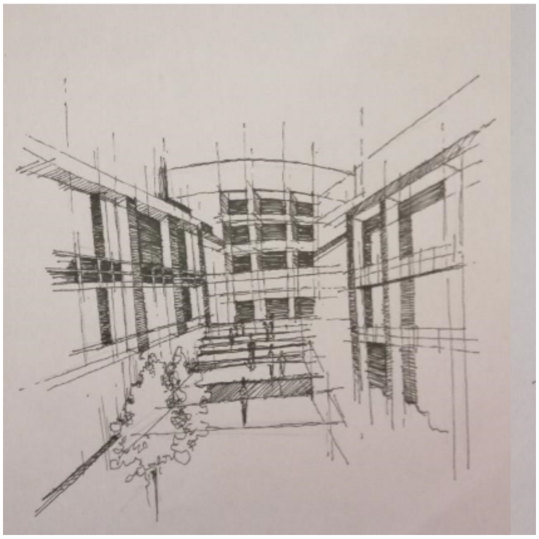

Fig. 1. A 3D sketch of a building

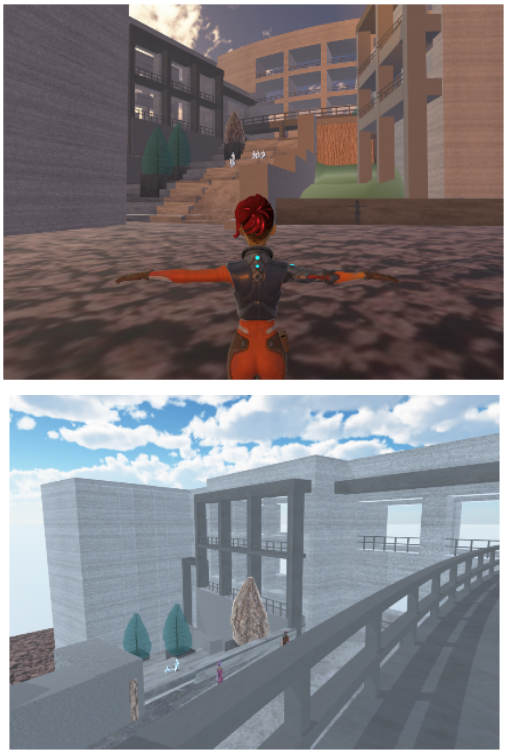

Fig. 2. Screenshots covering different views and environments.

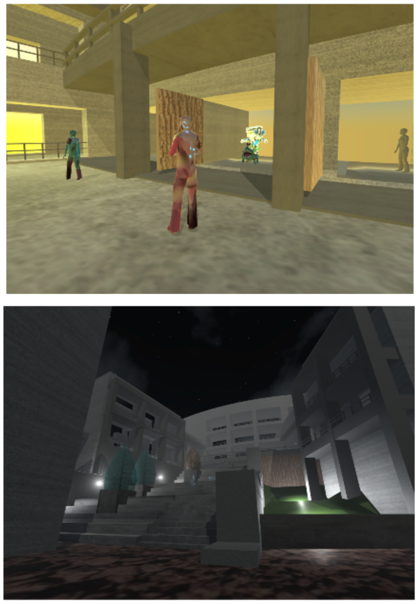

Fig. 3. Screenshots covering dusk and night environments

References

[1] Salon. (2019). A brief history of the “walking simulator,” gaming's most detested genre. [online] Available at: https://www.salon.com/2017/11/11/a-brief-historyof-the-walking-simulator-gamings-most-detested-genre/ [Accessed 8 Apr. 2019].

[2] Talumis. (2019). What is simulation | Simulation Software Explained. [online] Available at: https://www.talumis.com/what-is-simulation/ [Accessed 8 Apr. 2019].

[3] Technologies, U. (2019). Unity - Manual: Unity User Manual (2018.3). [online] Docs.unity3d.com. Available at: https://docs.unity3d.com/Manual/UnityManual.html [Accessed 8 Apr. 2019].

[4] Paauwe, M. (2019). Architecture Visualization Definition - Dragon1. [online] Dragon1.com. Available at: https://www.dragon1.com/terms/architecturevisualization-definition [Accessed 8 Apr. 2019].

[5] The Knights of Unity. (2019). Myths and Facts of the Unity Game Engine. [online] Available at: https://blog.theknightsofunity.com/myths-and-facts-of-unity-gameengine/ [Accessed 8 Apr. 2019].