Mini Project Progress Report - IS 31230

Sabra Explorer- Video game for Windows

R.M.D.A.P. Samarasinghe (EP-2057)

B.I. Batuwanthudawa(EP-2031)

Supervisor - Miss Samadara Dhanapala

Department of Computing & Information Systems

Faculty of Applied Sciences

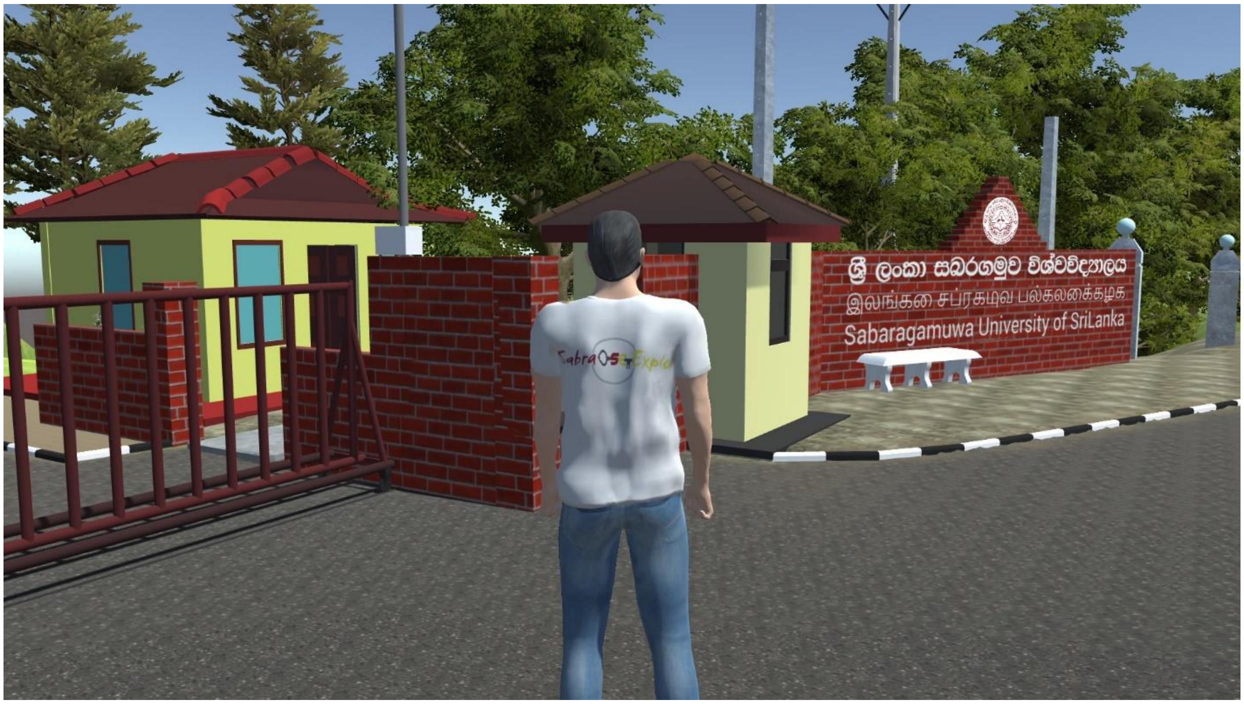
Sabaragamuwa University of Sri Lanka.

Belihuloya

Current Progress

Levels

When game starts level 1 is loaded.This belongs to a scene around main gate of the Sabaragamuwa University. All the models are created mostly look like real environment. The first objective is to find coins around the map nearby gate. When player reaches the expected amount of coins forward to other objectives, Upcoming objectives uses these coins so this objective flows the game story too.


ශ්‍රී ලංකා සබරගමුව විශ්වවිද්‍යාලය
இலங்கை சப்ரகமுவ பல்கலைக்கழக
Sabaragamuwa University of SriLanka

Perspective view of first chapter

3D Modelling

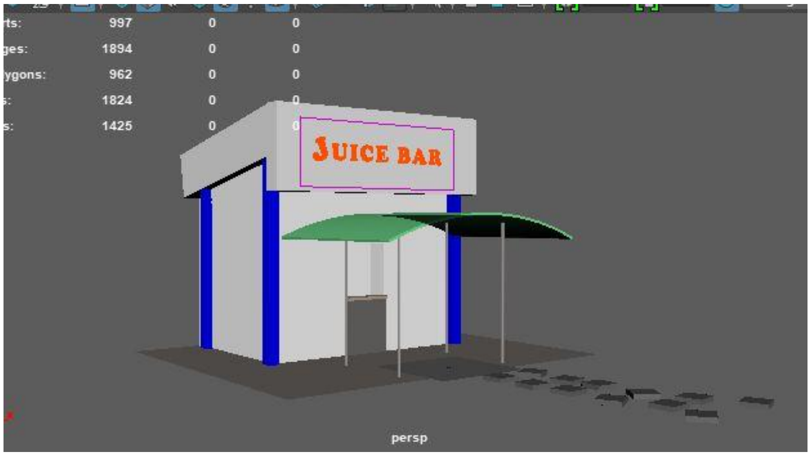
Up to now around 40 assets has been designed using Autodesk Maya. Some assets like trees, plants, flowers are downloaded by unity asset store. All assets 60+ up to now.

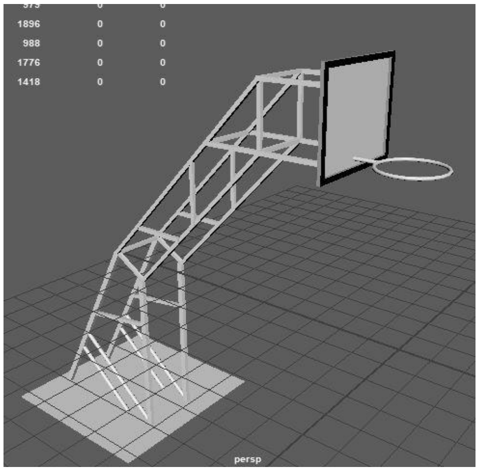
Main character has been created using Makehuman software which released as a github project. Character model, geometries, Material, pose/animate and rendering already given by software itself. Some materials changed according to game like shirt logo.

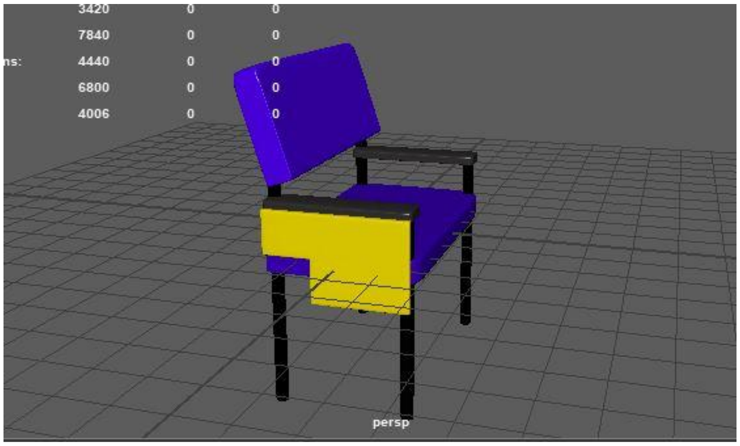
Front view

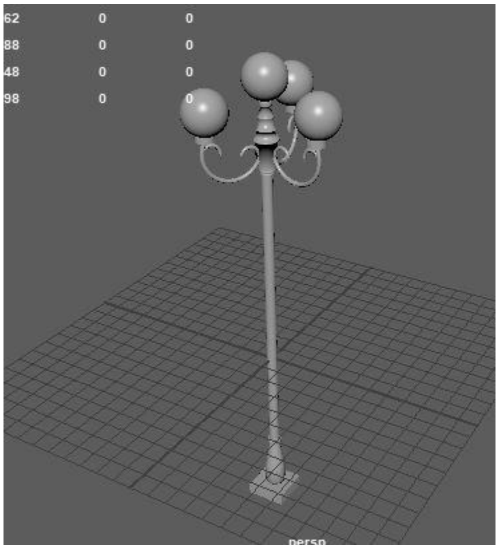
Back view

Most of the assets are designed by Autodesk maya referencing to real world assets , and modeling and texturing has been done considering them. Assets exported as .fbx format which unity can identify with textures. But when Texting board added we have to use unity material part. Then materialize separately added textures to unity project.









Scripting

Scripting part is the most valuable part of the game. Unity use C# object oriented language with fully controlled game asset, animation, player controller and camera controller also.

Player controller script

When player animation bind to key controllers, player controller script is used. In here W,A,S,D keys used to control player forward, left, Backward & right. All these controlling part set in Update method of script because every frame needs to be updated with controllers get touch when keys pressed.

Further player animations like jump, grab, interact with objects manipulated by player script. (Ex:- When "space" key press down jump animation implemented.) Player object tag with "player" tag sometimes object identify player by its tag.

Camera controller script

Camera is scripted to give a third person perspective of the player character. Camera follows behind the character as the player moves. And we scripted the ability to orbit the camera using the mouse movement. Camera has a default offset to view the character behind his back. We considered mouse sensitivity in scripting camera controller.

Main menu

First we created a wireframe in Adobe Photoshop. Then we developed User Interface for the main menu in Unity Editor. We included three buttons to navigate.

Start button – To start the game

Options – To navigate to the options menu

Exit – Exit the application

Using Unity's Scene Management library we bind each button with specific methods using on click event handlers.

In further development we have to create Options menu, giving the user ability to change volume, graphics quality and resolution of the game.

Non-Playable Characters

We implemented an Artificial Intelligence to a non-playable enemy character to follow the player if the player gets too close to it. For this we baked an area in the terrain which decides the area the enemy can move. And we set up a sphere around the enemy. So if the player intersects with the sphere the enemy starts to follow.

In further development, we have to implement the player and enemy interaction. And we have to implement enemy patrol to random areas to give unpredictability to the game.

Challenges

A Challenge we faced when we are creating the game environment is memory overload. It caused the game to lag. The reason for this problem to happen is all the assets we included loads when we start the game. So the memory usage is very high.

The solution we implemented for this problem is dynamic scene generation. From this implementation only the environment closer to the player will be generated. Rest of the environment loads to the memory when the player moves on the map.

Instead of one whole scene. We create multiple scenes to form the environment. When player moves forward in a specific direction. Scenes load according to the direction that the player moves. And unnecessary scenes will be unloaded.

Further development

In further development, we are going to implement other levels based on the first level. 3D modelling will be done by Autodesk Maya as previous.

Lighting of the environment will be done to add a realistic look to the game.

Scripting needs to be done to track game progress, add more non playable character features and in other game levels.

Dynamic scene generation should be created.