Design

Introduction

Program Design

Technologies

Unity

The application that is used to create this application is Unity. It’s website can be accessed from this link: <https://unity.com/>.

Unity is an excellent application for game development that uses C# and offers a graphical interface. It has different plans for students, businesses, and enthusiasts. Most of their plans are royalty-free. Unity focuses more on complex 3D game development. Many successful games like Fall Guys, Rust and Fire Watch were created with Unity.

Unity uses low-level languages like C+, C++ and C# to optimise hardware usage. It provides a sophisticated GUI workspace which makes it easier to create 3D worlds and adjust 3D assets. Users can see the changes as they go along in writing their code.

REPLACE THIS IMAGE WITH THE UNITY GUI OF ANIMAL SIMULATOR

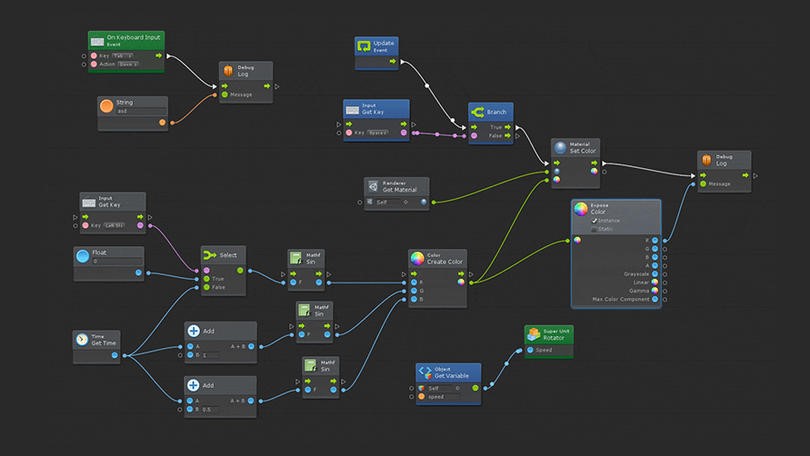


Figure 1 Unity Graphical User Interface

Unity supports a wide range of interfaces, operating systems, and VR hardware. Developers can use Unity in operating systems such as, Windows, OSX and Linux, including mobile operating systems such as iOS, Android, and BlackBerry 10. Unity supports 3 types of application programming interfaces such as DirectX 10, 11 and 12 which allows developers to implement better graphical shaders into their games. It also has VR support for Oculus, HTC Vive, SteamVR, Cardboard, Gear VR and Google Daydream.

Unity has a built-in programming feature called Visual Scripting. The user can click and drag gameplay functions to create games without writing code. However, we won’t be using this for Animal Simulator, the game will be written.

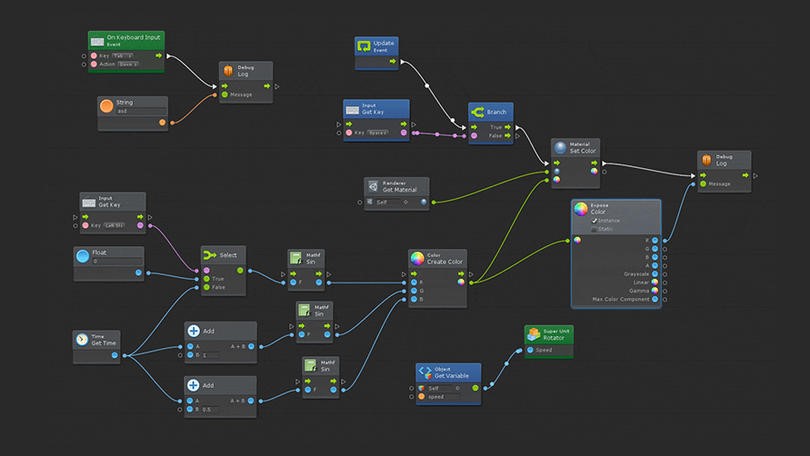


Figure 2 Unity Visual Scripting

Phaser

In 2018 we developed a 2D side-scrolling game called Jungle Ice Golem. It was developed using a game framework for HTML5 called Phaser. The game has a menu UI, 2 levels and 2 different maps, the first level is in the jungle and the second level is in a snowy environment.

The main story starts with a heroic Golem that makes his journey into the unknown fighting Trolls and collecting coins. He needs to collect coins and give them to the Troll King as payment to end the war and save the Golem race from extinction.

Phaser is a free web-framework available from this link: <https://phaser.io/>.

Phaser focuses on less hardware intensive game. It is mainly used for 2D game development for the web. It doesn’t have a GUI workspace or Cinematic view and Visual Scripting, so creating the game world and adjusting assets are more sophisticated.

It uses JavaScript which is a web-based language. JavaScript’s programming paradigm is based on the procedure, whereas C# is more class-based object-oriented programming. The major downside of Phaser is, it is less supported than Unity. When a new version is released certain codes are changed and there is less documentation.



Figure 3 Jungle Ice Golem Main Menu



Figure 4 Jungle Ice Golem's Gameplay Preview

Structure of Unity

Controls

 Hand Tool – this is used to move around the game world and hot key for this is Q

 Move Tool – this is used to move game objects and hot key for this is W

 Rotate Tool – this is used to rotate game objects and hot key for this is E

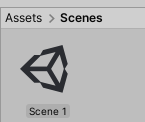
 Scale Tool – this is used to change the scale of game objects and hot key for this is R

 Rect Tool – this is used to change the length of game objects and hot key for this is T

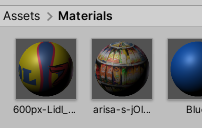
Hierarchy and Inspector

The Hiearchy contains assets of the game that can deleted, grouped into folders, and configured in the Inspector. The Inspector, inspects the game object and allows the user to change it characteristics such as tags, names, size, colour and components.

Scenes Folder

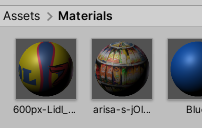
The Scenes Folder contains scenes of the game. A scene is one part of the game e.g. the Main Menu scene, and this scene will contain the assets, fonts, and scripts. Instead of working in the whole entire game, we can create scenes to divide the game into different parts.

Prefabs Folder



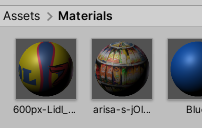
When you add a new game object inside the Hierarchy in Unity and configure it, you can drag and save it inside he Prefabs Folder. These game objects are already configured, and they can be reused or configured again.

Assets Folder



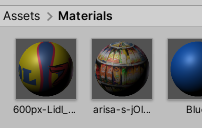
The Assets Folder is the main folder that contains the scenes, prefabs, scripts, and materials. When an asset is imported from the Unity Store a new Assets Folder is created.

Scripts Folder



The Scripts Folder will contain the scripts files of the game code. Scripts can be added as a component.

Materials Folder



The Materials Folder will contain the materials, textures and colours.

Design Patterns

Animal Simulator’s design pattern doesn’t have different Model View Controllers.

Application architecture (1 page)

Include a labelled block diagram of the application.

### Database design

Our game will not require database because it will not be hosted into the web. It’s an offline game and the save files are stored into the computer.

Process design

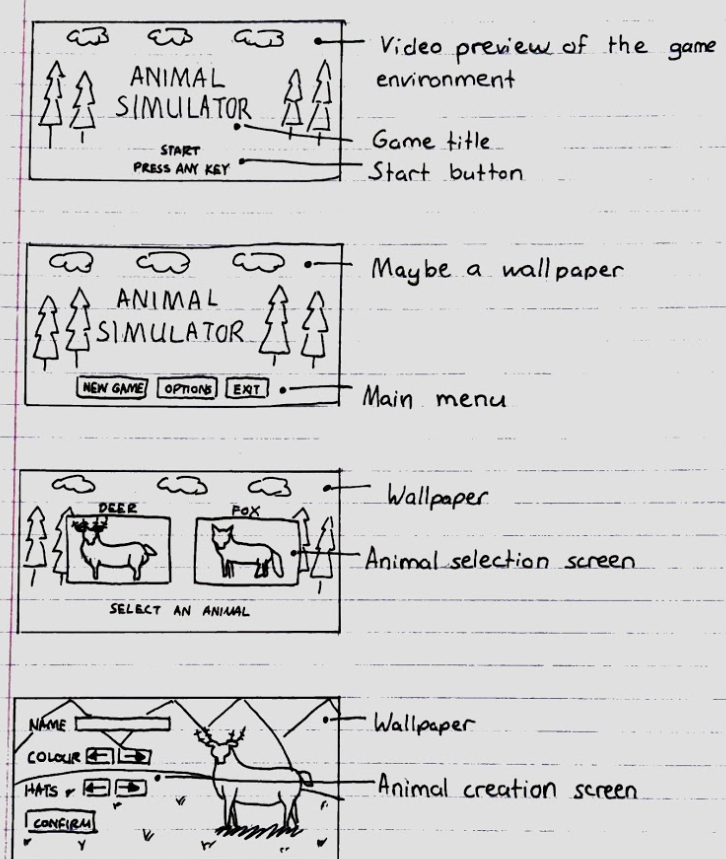
There are a number of techniques which can be used to aid the coding of an application.  The following diagramming techniques are some of the ones which could be useful. Discuss with your supervisor what is appropriate for your project.

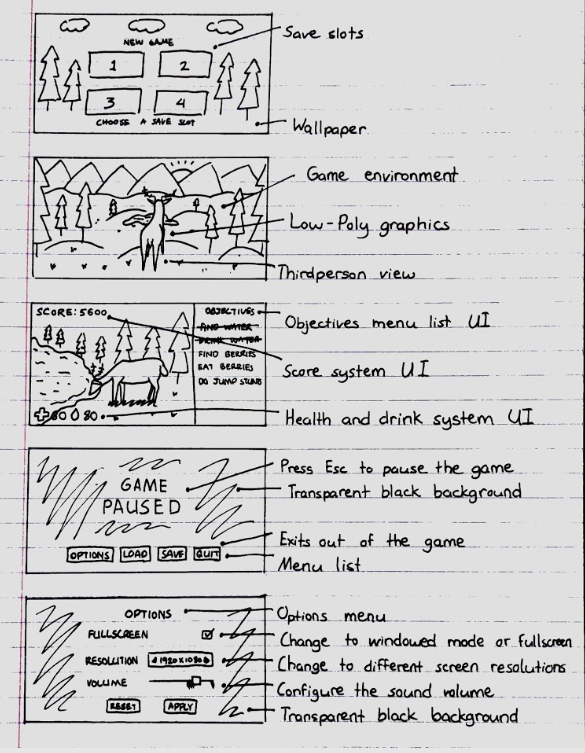
* Class diagrams
* Sequence diagrams
* Flow charts
* Pseudocode

User interface design

This section describes how the interface is designed.  The section will differ depending on whether an app or a game is being developed.

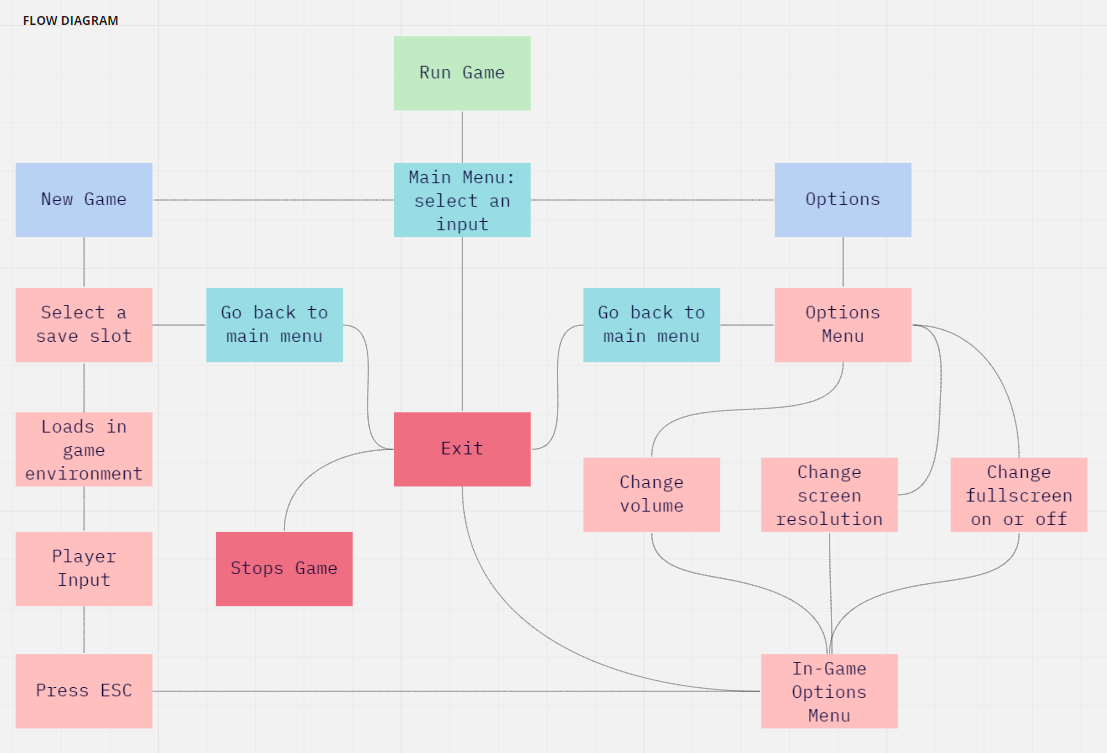
Wireframe





User Flow Diagram

This shows how the user will navigate from one page to another page within the application.



Style guide

This shows the colours, typography and layout for a single page.  Often the theme for this page will be used for all pages in the app.  Within this section, explain which colour scheme is being used and why that colour scheme has been chosen and also which fonts are being used and why they have been chosen.  This section also covers grids and spacing.

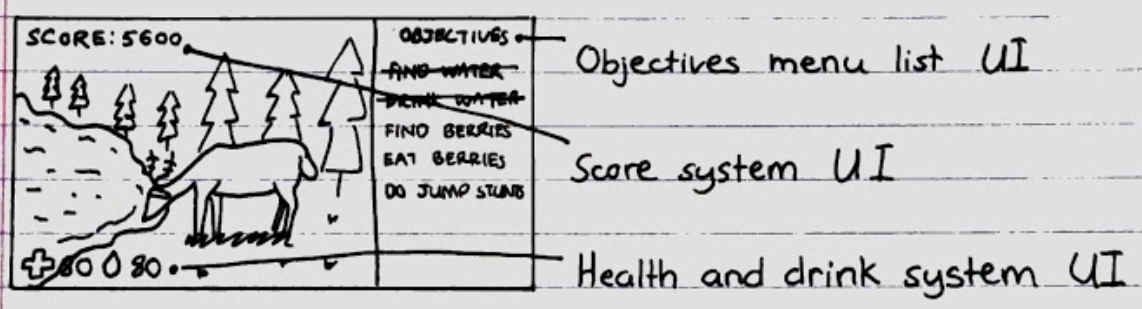
Storyboard

This will be required for any games being developed.

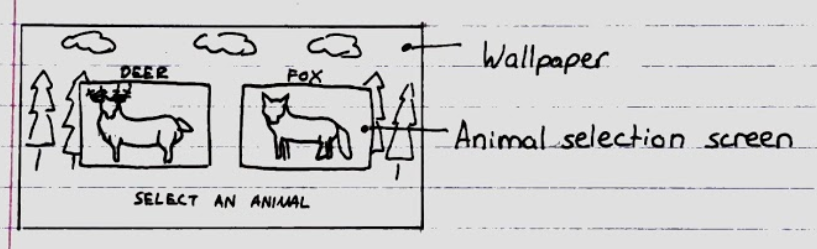
Level Design

Animal Simulator will not have a linear level design where the player must reach a certain level to equip a weapon or access a place. Not many sandbox games have this type of level design.

In Lego Worlds, when a player levels up they can unlock more maps and skins and we may implement this feature into the game. This feature will work well with a score system. If the player reaches a particular score and completed all the objectives, they can unlock a new map or a new animal. The player of course can continue in the same world.



The player can select 2 different animals, but by default, only 1 animal is available when they start a new game. When a player unlocks a new animal in this case a fox, if they go back to the main menu and start a new game, they can select the fox and play in a different world environment.

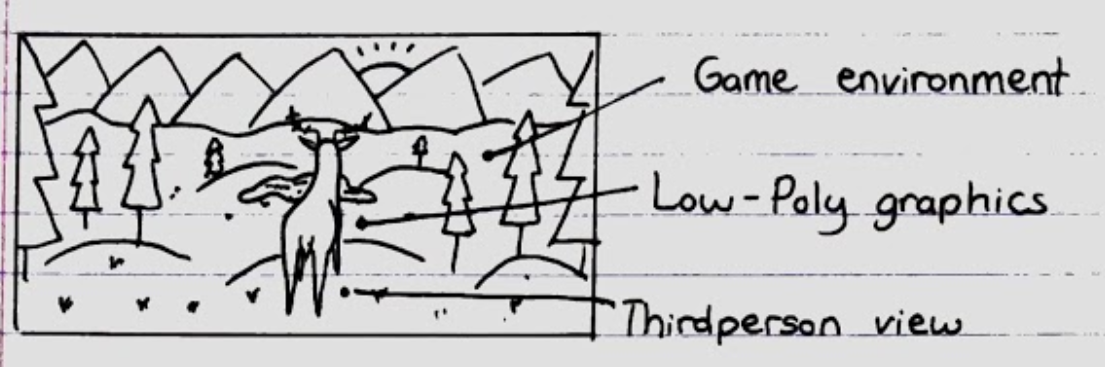


Environment

The game environment is set in a medium-sized open world forest. The forest floor is covered with flat grass and the edge of the world environment will be surrounded with mountains that has collision detection so the player cannot go out of bounds.

Inside the forest environment, there are evergreen trees, small lakes, grass, mushrooms, and rocks. All these assets will also have collision detection.

The player can eat the mushrooms or grass which will increase the score and decrease hunger. The player can also drink water from the lakes which will decrease thirst.



Conclusion

Write a couple of paragraphs summing up the chapter.  Explain what area your project is about.  Describe what the chapter has discussed.

References

16/11/2020

<https://www.visartech.com/blog/unity3d-2020-what-is-new-and-how-business-benefits/>

<https://store.unity.com/#plans-individual>

<https://unity.com/how-to/make-games-without-programming>