CO3015 Computer Science Project

Interim Report

Games with Smartphones

11/2017 (Interim Report Submission)

Department of Informatics, University of Leicester

sayim khan

Contents

DECLARATION: All sentences or passages quoted in this report, or computer code of any form whatsoever used and/or submitted at any stages, which are taken from other people's work have been specifically acknowledged by clear citation of the source, specifying author, work, date and page(s). Any part of my own written work, or software coding, which is substantially based upon other people's work, is duly accompanied by clear citation of the source, specifying author, work, date and page(s). I understand that failure to do this amounts to plagiarism and will be considered grounds for failure in this module and the degree examination as a whole.

Name: Sayim Khan

Signed: 

Date: Thursday, 07 December 2017

Project Objectives:

My primary objective and challenge is to create and complete 8 – 10 game levels. These levels can be thought of being floors of a building and the overall objective is 7 – 10 levels. Completion of each floor will mean that designing of the level (unique theme, atmosphere model and texture placement, etc), lighting and shadows, sound, story dialogue system, gameplay mechanics are completed and polished, this can be considered a milestone and determining overall progress because each floor can be considered a 10-15% of the project, because when all the floors are 100% completed, the game will be completed.

My second objective is to create a unique story for the game that will be told over the course of the levels with custom characters that have personalities that I create. This will be very challenging as I will have to write the bio and personality traits for each character, which I have never done before. I will also have to learn to design and model each character, so that they have their own unique appearance via 3D Modelling solutions such as Blend or Maya LT, which I have never used or tried 3D Modelling, so learning this and trying to design the characters will be another challenge. I hope to make 4 unique character that are heavily involved in the story.

My third objective is to create several RPG style gameplay features required for this project, these include but are not limited to:

- Dialogue System: This will be the primary way of allowing the game to communicate with the player as well as telling the story. For this system, I will have to figure out and design a way of storing and using story text as well as NPC text. I will also have to figure out the best way of presenting the dialogue to the player, through traditional text boxes or Visual Novel system dialogue with textboxes, options and NPC Portraits.
- Inventory System: This will allow the player to store and use items though out the game, I will have to design the interaction, storage and UI for the inventory system. My challenge is to find a way of making the inventory system intuitive and integral to the gameplay.
- Statistics, Levels and Experience: I will have design and implement a level and experience system, which will allow player to level up and increase character statistics, this will also allow player to take on more powerful enemies, but add an element of progression that will help make the gameplay more fulfilling and rewarding. My challenge is how to calculate how much experience will be required for the next level and how much experience defeated enemies will give the player, as well as trying to predict player level at the start of each floor.
- Combat System: This gameplay feature will allow the player to find and fight hostile NPCs that will attempt to block the player's way and attack them. I will have to design a smart combat system, one where the enemy NPC will try and pick the best move they can again the player, in an attempt to defeat the players character. My challenge is to try and create smart NPC that some how will pick the 'best next move'.

These are my main objective and challenges that I will have to find and make solutions to overcome. By overcoming theses, it will bring me closer to building my ideal game.

Description of Prototype:

My prototype was a basic and early build of the Unity game a few of the things that were implemented were the early version of the dialogue system, player movement throughout the game first level, interaction with game object, basic animation and menu UI with stats tab functional. These systems will be carried over to the final version of the game.

A few updated to the systems will be that the dialogue system will be updated to hold multiple lines of text though Unity Persistence and text or JSON files and will be dynamically loaded in when specific conversations are started. Another update will be that the menu UI will have the other system tabs functional.

Systems that will be added to the final version of the game that were not in this early build will be the combat system, this is where the player walks in range of an AI opponent and will initiate combat and Combat ends when the player or the AI opponent's HP is zero.

Another system that was not implemented in the prototype build was the inventory system. This allows players to find item in the game world and use or equip them. When the player interacts with a chest in the game world, they will be able to obtain the item and then have access to it in their inventory.

The final version will also have a save/load system for storing game progress and allows the user to continue after closing and reopening the game.

The final version will also have the implemented questing system, which will log and follow the player progress on obtainable quests.

Information Sources:

My primary source of information at the moment is from Pluralsight and YouTube, the Pluralsight videos that I have been watching are Tutorial videos on the facilities that I have been learning to using such as Maya LT and Unity Editor.

The Maya videos have been teaching me how to use the suite by showing me around the User Interface and what the different tools are and do. This video is 'Introduction to Maya 2017 by Justin Marshall and Eddie Russell'

The YouTube videos that I have been watching are Unity tutorials on how to use the editor and what the different facilities and tools it has. The first video I watched to help me get started with unity was 'How to Make an Android Game With Unity by Charger Games', this helped introduce me to the Unity Engine and the Editor, helped me understand how to use touch input from a touch screen on an android device. This video also introduced me to C#, as I have never used it before. I also watched a C# Tutorial by Derek Banas to help me get to grips with the syntax of C#. A few other videos that I watched include 'Live Training Mar. 16th, 2015: Adding Music To Your Game by Unity', a tutorial video from the developers of Unity on adding ambient sound to a Unity Project. The last video I watched was 'How to Design a Level in Unity 5 by Sykoo' this helped me understand the level design tools that unity provides, essentially how to build by game world.

The tutorial on Persistence from the Unity tutorial website (https://unity3d.com/learn/tutorials/topics/scripting/persistence-saving-and-loading-data) will show me that persistence in Unity is and how it can be used to save and load data files from within the Unity App on a smartphone.

I have finally also been using the online Unity Manual (https://docs.unity3d.com/Manual/) for information and reference on how to use and do certain things such as hardware optimisation via Quality Settings (https://docs.unity3d.com/Manual/class-QualitySettings.html)

At the moment I have not got much in terms of text references because I haven't started using any well-known algorithms such as Min-Max as I haven't started the Combat System which will require research and reference. Most of the things I have been doing are specific only to my game and the way that I implement things in the game, such as the inventory system, which is custom to my game and is based on my programming knowledge. I will have more sources as I make progress on my game and will start needing more complex algorithms or coding techniques.

Software Architecture, Algorithms and Data Structures:

I will now underline a few of the main systems that the game will use that I have a rough idea on how to implement.

Inventory System: The Inventory System is the how the player will be to use and hold multiple items. The Inventory System will be accessible through the 'phone' Menu. Here they will have a grid or list layout that will display a thumbnail of the Item as well as item information, there will also be the option to use or equip the item dependent on what the item is. All of this information will be obtained from an Item subclass based on what the item is, so if the item is a weapon, it will have the WeaponItem class which will hold information such as name, the stat boost for the players character as well as the specifics of how the item will interact with the inventory System. These classes and the Item Interface that they inherit from will be kept on a created GameObject Prefab, which is an object that can be directly placed in the game world and has several components such as a Model (3D Model of the weapon), Animator Controller, C# scripts, etc. The Prefab will be stored in a Resources folder for storing game assets. When adding a new item from an item chest which will hold a string array of item name which already exist in the resources folder, the Inventory class will copy the item name and store it in an internal array. When the item information or the GameObject its self is needed, the Inventory System will fetch the correct C# script for information when displaying item information in the Inventory Menu or instantiate the GameObject itself into the game world when the item is to be equipped onto the player where it can fetch the script for adding the stat bonuses to the player's character.

Save/load: this is a feature that allows the user to save their progress as well as things like what items they have in their inventory, all of their current quests, what level they are on and what their current position on the game world is, what weapon they currently have equipped is, their current level, their skills, the current stats, current experience points amongst other things. When thinking of how to design and implement this, I will have to search for specific data from different locations such as the inventory system, Character class, etc and save it in a type of file in some form of data structure to organise the data, I have yet to decide on what data structure for storing the data in the file, because I need to look into what kinds of data structures can be used as well as what file extensions Unity can use for Persistence on Android. Another part of this is that I will then have to initialise this data on game start up there it will have to check if there is a save file and then fetch the data and change or set the data in the correct locations, to try and replicate the previous state. This is how the game will store the users game state and re-load it next time the user starts the application up.

Dialogue System: The basic dialogue parsing system is in place on the Prototype demo, where a character class can access the static addDialogueText method of the Dialogue system on interacting with the Player and send a text paragraph and the method will parse the text to the dialogue window and will allow the player to progress through the text and close on finish reading the text. I will however be expanding this a fair amount, as it needs an additional layer between the NPC invoking a method and the Dialogue System, this layer is the Dialogue Store or file containing all of the dialogue in a large organised file which will hold all of the conversation dialogue in the game in the file and when an NPC initiates a conversation the dialogue store will look at what their name is, what game level they are on and what their in world coordinates are and then fetch the correct dialogue for them then parse that onto the screen for the player to read. I also will like to add a in-game notification system which will help notify the player of certain things such as the names of the items obtained, a completed quest or other things that the player will need to be aware of. This notification system I will be adding onto the Dialogue system as another function, so all text related things are unified under the Dialogue System. I will do it so that the System has another method called setNotification or along those lines and it will require some information like

the message to be displayed from the class calling the method, it will then handle parsing the message and the UI elements as well as the window interactions. The final piece of the Dialogue system will hopefully be a choice System where the user during a dialogue can select a response to something said by the NPC and this will branch the Dialogue off. The system would interact view closely with the Dialogue Store and search and fetch dialogue based on the response.

Combat System: The Combat System will be how the player attacks and fights enemy NPCs who attack the Player and Combat will end after one side loses all their Health Points. The player will start to initiate combat when they tap on the enemy NPC, when they do 2 UI health bars will appear on screen that display the player and enemies HP. The player character will start to auto-attack the enemy at regular intervals, these attack movement and attack timing will be handled by the Unity Animator, in which I can animated the motion of attacking and this will allow me to do certain things during the attack motion such as turn on and off collision detection on the players weapon to detect when the weapon hits the enemy, when it does, damaged will be registered onto the enemy. The Animator will also handle the timing, so when an attack motion has ended the animator will wait a second or so and start another attack, this will happen until the player clicks away or the enemy has been defeated. While the character auto-attacks, the player will have certain 'Skill' buttons on the UI available to the them to use, these button to special attacks, such as heavy hit, multi-hit skill or quick attack. Heavy will interrupt the current auto-attack and do a more damaging attack with an action duration the same as the normal attack, Multi-hit is a sequence of uninterruptable attacks that last a little longer than a standard attack, but do 4 or 5 hit in that time span, finally quick attack is a very fast attack that does less damage than the others. These Skills will have a cooldown period, where they cannot be used until the cooldown has finished, usually a few seconds. These attacks will have special attack motions and will invoke different damage methods. The Damage dealt to the enemy will have to be a formula based on both their stats, such as agility, defence and attack, meaning an attack is able to miss. The enemy AI, will using the help of Min-Max algorithm will have to think what their next best move is while auto-attacking the player because enemies will have access to counter-skills, a way of countering the different skill attacks that the player has. For example, if the player uses a heavy attack and the enemy uses a heavy counter, the enemy will take much less damage and be able to inflict additional damage onto the players character.

Questing System: The questing system is a function where on each game floor there is a specific set of tasks or quests that are obtainable via NPC conversations and have rewards for completion such as weapons or potions. When a specific Quest NPCs conversation has started, they will start to talk about their dilemma and their task that they want the player to complete. After the conversation has ended the Quest System will copy a Quest information script from the NPC, this script will have information such as task summary, a location or area that the task will take place, a reward and the NPCs name. This will then be stored in the Quest System class on the Player Object and can be viewed in the Quest Tab of the 'phone' menu, which will display all the information about the quest on selection from the list of active quests that the player has. The Quest System will then log what the player does and track the quest requirements actively by either checking NPC conversation names or checking Enemy types or names. On Completion the Player should get a notification from the Notification system that the quest has been completed and they should return to the NPC to collect the rewards. At this point the quest will be moved into a completed section of the questing menu, making space for the other active quests.

Planning and Timescale:

The timescale has had to be shuffled around because of the way that Unity handles Scripts and GameObject, I decided to develop and flesh out most of the systems such as the Combat, Dialogue, Menu, Inventory and Questing Systems before the creation of the 2nd floor of the game as it means that as I develop the other floors most of the games systems will have been set up and ready and I just apply the ready systems to those floors meaning that I can get on will adding the content such as adding additional dialogue, quests, etc for the other floors while making minor modifications along the way. I hope to start 2nd floor creation before the end of Christmas. Then the plan is to create the other floors, character models and fine-tune the systems during the 2nd semester

Career Plan

Where do I want to go after Graduation?

I would like to have a job either in 2 positions, first where I can use the years of programming I have learned so that I know that these years spend learning all the different things in university such as Java, SQL, etc. In this job, I really hope to learn even more and build on my current skills such as learning other languages, new techniques or different way of doing things using languages that I already know, like a Software Developer for a software development company or a company that builds software solutions for clients.

The other type of job is in Game Development, I find that Game Development, while very long, is fun because its not all code, there's game level design, modelling, animation, hardware optimisation and other things. I would like to be part of a Game Development company and build video games, as a avid gamer as well I think it would be very interesting.

What will I do this academic year to get there?

Firstly, I will have to make sure that my CV is up to par and ready. I will ask the Career Development service help with this part. I will have to also look at all the different types of vacancies and graduate schemes there are for these types of jobs and apply for them. I will have to practice isometric tests as I tend not to be very got at them, in preparation for the application process.

How does my project contribute to my career?

In a fairly large way actually, firstly it was the push I needed to get me out of my comfort zone and try something completely out of the blue and get a taste of game development. It forced be to learn about Unity, how to use it as well as gain experience with it. This is helpful as unity is one of the top Game Development Engines in the Game Development world, along the way I can try different one like Unreal Engine 4. Next it made me try another high-level language such as C#, which is the 2nd most popular game development language other than C++ (which I am also learning). I also got a taste and experience with other things such as 3D Modelling and Game World Design, which before the project, would never have touched. It also will improve my programming skills as there are some fairly complex things in the project that I will have to tackle. Finally, the project is my first game and can go towards my portfolio as something that I created.