Run N Jump

CE301 Capstone Project Final Report

By Bartosz Markiewicz

Supervised by Dr. Doctor, Faiyaz

1704256

Acknowledgements

Throughout the writing of this report I have received support and assistance from numerous people whom I’d like to thank. Firstly, I would like to thank my project supervisor Dr. Doctor, Faiyaz who was invaluable in providing me advice and challenging me to make the project as best as I could. Allowing me to ask many questions and providing me clear answers when I was feeling uncertain on how to proceed.

Secondly, I’d like to thank my CE301 course supervisors; Dr Vishanathan Mohan and Professor Anthony Vickers who provided a great deal of insight and information into the creation of a final year project, their support throughout the year was vital. Their lectures were very useful in explaining everything.

Furthermore, I would also like to thank my parents who gave me a sympathetic ear and being there for me in times of struggle. Them answering my calls in the morning after an all-nighter of work was always reassuring. Lastly, I’d like to thank my fellow Computer Science students at the University of Essex, school of Computer Science and Electronic Engineering department who gave me encouragement and a time of relaxation in-between the work on this project. A distraction was vital to rest my mind between times of work.

Abstract

The Arcade has been called the grandfather of modern video-gaming. The influence of arcade games can be seen even on modern games. The video-game market has been growing rapidly and is projected to continue growing. This had encouraged me to recreate the classic side-scrolling Super Mario Bros experience with my own twist on the formula.

My project goals have been to create a Mario game with light RPG-elements such as friendly NPCs, a text-based story narrative and a basic inventory system. My focus has been on having a working game with collisions, enemy AI agents and other dynamic world features such as unlockable doors. I aim to have it be playable on Windows and Android platforms. For Android I intend to have the game published on the Android Google app store in the future. To create this game, I had utilised mainly the LibGDX Java game development framework based on OpenGL using the Android Studio IDE. I had used TMX tiled maps for the creation of my game levels. Furthermore, I intend to add an online SQL based high-score system for the game to allow the players to compare their score to others. Additionally, I had added a wide range of animations and SFX into the game to give the game a more professional feel to it.

Table of Contents

aCKNOWLEDGEMENTS1

aBSTRACT2

**TABLE OF CONTENTS3**

iNTRODUCTION4

Chapter 1: Literature Review4

1.1 Context5

1.2 Problem Definition5

1.3 Serious Games5

1.4 Design5

Type chapter title (level 3)6

Chapter 2: Project description4

1.1 Aims and objectives5

1.1.1 SUMMARY OF OBJECTIVES6

2.2 Scope5

Introduction

The objective of my product is to have a fun and engaging 2D side-scrolling platforming game for people who enjoy videogames. A 2D side-scrolling platformer videogame is a game where the user sees the world through an orthographic projection, this can be a side-view, the front, or the top of the imagined 3D world. In my case the projection is from the side – the user goes from left to right with gravity pulling the player down on the x, y coordinate system, down being the ground, up being the sky, this is typical of a 2D platformer. I chose to re-create Super Mario because I realized the classic experience was lacking RPG elements which could make it better.

Chapter 1 Literature Review

1.1 Context

In the early 1980s the video game industry was growing steadily. There was a ‘golden age of arcade video games’ from 1978 to 1983. With popular releases such as Space Invaders(1978), Pac-Man(1980) and Donkey Kong (1981)[6], . following this economic boom the video games industry crashed in 1983 with many game developers going bust. [1] Many saw it as a failure of the industry, but following the bust the industry began to recover globally with classic hits such as Tetris(1984), Super Mario Bros(1985) and Blasteroids(1987). Later in the 90s there was a boom of the fighting genre of arcade games.[7] The release of the Mortal Kombat and Street Fighter games shaped the overall arcade game market for years to come. Those big Arcade games of the 80s and 90s have inspired countless modern re-iterations or similar games. Even though currently the video game industry is the fastest growing software industry segment. [3. p25]

1.2 Problem Definition

Yet there isn’t really a major notable modern alternative to the Super Mario franchise which has been dominating the 2D platforming genre since its conception. Super Mario Bros U for the Wii U released in 2012 sold 5.8 million copies despite the console itself not being very successful, showing that the Super Mario franchise is still alive and well after 40 years.[8] Furthermore Super Mario games are still being made, some with innovative 3D graphics and gameplay such as Super Mario 3D World but others with the classic 2D side-scrolling formula intact such as Super Mario Maker 2. There hasn’t been much innovation in the franchise since the original Super Mario Bros, aside from the move towards 3-dimensional gameplay. The 2D Super Mario Maker 2 plays the same as Super Mario Bros. There are not many new interesting features, increased complexity or fundamental changes to the formula. Aside from more levels (allowing user creation of levels), most of the enemies, world features and gameplay elements haven’t changed as much as one would expect in over 40 years, aside from being modernized. Some people would say, ‘if it isn’t broken don’t fix it’, but I believe there is room for innovation in the genre. I personally found things missing and I am aiming to add these things into my own Mario-inspired game. The rest of this report will specify what these things are.

1.3 Serious Games

Arcade games are not just used for fun. The development of games has often been used by new software engineers to teach themselves computer science principles they can use in all spheres of software development. But it is not just the process of creation of a video game that can have an educational benefit.

Playing a video game can be very educational and useful, most arcade games can be in some way adapted for an educational purpose. Even a simple game like Tetris can solve an educational purpose, at MIT researchers created a web-based vocabulary-drill version of Tetris. It works using voice recognition in the form of Web Accessible Multimodal Interface software [4. p4]. The game begins like classic Tetris with a piece appearing at the top of the game board. The twist is that also an image appears every time, if its the first time, the image appears with the corresponding word. If the image is appearing for a subsequent time it appears without the word and this is when the player has to say the word associated with the image out loud to unlock block rotation. [4. p3] Through this the player learns new words and their meaning.

There are limitations to the ability of games to teach, and certainly adding learning to a game can often lead to lesser overall enjoyment of the game. [4. p5] This necessitates the developer to often choose between either high learning effectiveness or fun when choosing how to implement features.

Games with a ‘higher’ purpose besides entertainment are often called serious games. This can be a game with an educational/learning aspect or a game promoting health/therapy. Currently the biggest challenge in developing serious games is the difficulty in assessing whether the game possesses actual learning value/benefit, which is why there is a lot of research going into game related learning analytics. [5. p23]

1.4 Design

When designing the product, I had conducted research into game design and have learnt that games are meant to arouse meaningful immersive experiences, furthermore a game consists of various elements such as mechanics, story, aesthetics and technology - these are equally important. Each player’s experience is totally unique. [1. p1] There is a subset of serious games which can be educational, educational games have to be designed properly to incorporate engagement that integrates with educational effectiveness - the challenge for the game designer is in finding a balance between game-play and learning objectives. The goal of a designer is to balance the five elements - flow, immersion, presence, arousal and engagement. [1. p2] If a designer wants to teach something, the learning should impose a cognitive load. If the learning objectives are disconnected from the gameplay the game may fail to produce educationally effective experiences. [1. p4] Additionally the difficulty of the game should be increased when the player is bored. [1. p7] Simply put player engagement is important, the experience of the player needs to be rewarding, non-player characters need to resemble real players as much as possible - since gameplay alongside real player had much higher user engagement. Lastly the game should give feedback to the user to show how he is performing, especially if the game has a learning component. [1]

Games can be fun and educational due to tapping into the motivational drivers of human behaviour - positive/negative reinforcement and emotions. The principles central to gamification and games in general are the MDE framework - mechanics, dynamics and emotions. The types of people involved are - players, designers, spectators and observers. Mechanics are the decisions that designers make to specify the goals and the boundaries of the situation to be gamified. There are three different types of mechanics, setup, rule and progression mechanics. Which are tremendously important for games, setup mechanics shape the environment of the experience. Rule mechanics shape the concept or the goal of the experience to be pursued. Whereas progression mechanics dictate the reinforcements present in the experience - this increases the likelihood that certain behaviours will be repeated in the future. [2]

1.4.1 Tools Considered

The small touch screen controls of Android and iOS favours the simpler arcade kind of games [9] whilst windows computer games with keyboard, mouse and possible periphery control gadgets led gamers owning those devices to be more inclined towards more complicated and competitive games.

* LibGdx - A flexible cross-platform 2D video game development framework based on OpenGL 2.0 and the Java programming language with some C and C++ components for performance intensive code It allows a project to target Windows, Linux, Mac, Android and iOS. Supports many 3rd party tools. Has low level OpenGL helper features such as meshes, textures, framebuffer objects etc. Additionally, contains high-level 2D APIs relevant to my project such as Othographic camera, sprite batching and caching, 2D particle system and a TMX tile map support.[10]
* SFML - Simple and Fast Multimedia Library, it is a simple interface for the development of games and multimedia for multiple operating systems, Windows, Linux and macOS. It is a C++ API but can be utilised from C#, Java and Python among many others. It is an exclusively 2D graphics-based library. It is more of a cross platform layer abstraction API rather than a straight game development framework. [12]
* Pygame - Is a Python game programming library used for the development of 2D games. It is based on SDL. It allows for easy multi-threading, utilises optimized C and assembly code for core functions, which is much faster than Python code. It prides itself on being easy to use due to requiring a small amount of code while giving the user a lot of control over their game. Many indie games have been made with it. It is very modular so its possible to utilise different libraries for specific aspects of the game. [13]
* Java- Java has Swing and Awt which can be used for 2D graphic work, this makes the language able to be utilised for making arcade games without any external libraries. It is great for Windows development and doesn’t require any installation porting the game to Android is trivial. Porting to other platforms is also possible due to the JVM but harder.
* rise.global - Allows for the creation of dynamic real-time leaderboards from a CSV file (Comma Separated Values) or excel spreadsheets. Useful for adding a player versus player (in terms of score) competitive element or a goal to an arcade game. It has an extensive list of features such as email updates, score card, scheduling of score collection, etc. [11]

Chapter 2 Project Description

2.1 Aims and Objectives

My main goal was to create a 2D side-scrolling game for Windows and Android platforms which gives the user a fun and engaging experience. I had chosen to name the game ‘Run n Jump’. My project statement was to re-create a classic arcade game. After careful research I set out to recreate the Super Mario Bro. I wanted to try my own hand at re-creating the classic experience with my own twist on the formula – while keeping what made the classic experience fun and engaging intact.

I chose the Android platform due to my research showing that the Android market would be most appropriate for this sort of game alongside the iOS market. If given enough time and resources I would port the game to iOS but time constraints only allow me to focus on two platforms at maximum. I had chosen to make the game for Windows too due to the ease of testing and development on Windows.

In my game the player would play a character like in Super Mario, he would move this character using the Android on-screen touch controls or the keyboard if on PC. The player will be able to set his own player character’s name when starting the game for the first time on the device. This will be an arcade game at its heart it will compete on the Android and Windows markets with other arcade games. I aim to have it released on itch.io – an indie/arcade game distribution site for Windows and on the Google play store for Android In the future.

My twist on the formula is the addition of RPG-elements such as quests, NPC dialogue, non-linear level design and more player choice in the way he has a choice when to use a power-up, which the player can store in his inventory and which route to take through the level. The survival game mode also adds unlimited replay-ability to the game which is lacking in Super Mario Bros.

The game upon launching will show the player the main menu screen, this is where the player can pick the game-mode, after choosing the campaign game mode the player will pick the level to play. Initially only the first level will be unlocked. When the player starts the level the gameplay loop begins. The game world is loaded, and the player character is placed with the camera following the player character. The player must avoid any dangerous hazards – spikes, enemies, etc. The main objective of a campaign mode level is to reach the end of the level victory flag, secondary objective is to reach highest score you can. The game will have an aspect of exploration with the idea that the player must find the end flag, each level will have multiple non-linear routes the player can take and optional content. On the way the play can interact with collectibles, use power-ups, talk to NPCs, reach checkpoints to respawn at in case of player character dying. The player can die and respawn at a checkpoint if he has lives left. The player can receive quests or tasks from NPCs he encounters which can give him rewards and things to do. The quests can be something like finding another NPC by interacting with him and returning for a reward, killing certain number of enemies or finding something.

The game will have a storyline aspect in the campaign game mode, the story will be conveyed to the player through the NPC dialogue system. The NPCs will talk to the player and tell him things. There will be a trading system where the player can purchase power-ups, other items such as keys (for unlocking passageways) or quest items from vendor NPCs.

The second game mode, the survival game mode features an infinite procedurally generated world with a more arcade-like gameplay. The goal in this game mode will be for the player to survive and run as far as he can to the right, gathering as much score as possible in the process. This game mode is called survival because the player is chased by a wall of fire of an erupting volcano, the idea is that the player needs to keep moving to the right of the screen otherwise he will die, any obstacles that delay the player can result in the wall of fire catching up and killing him. As the player runs the level is generated in front of him to provide a potentially endless experience.

2.1.1 Summary of Objectives

* Working on most devices of the proposed user-base
  + Android version 18
  + Windows 10
  + Device with at least 1gb of RAM memory
* Two Game-modes
  + ‘story’ based campaign mode of 3 levels which need to be done sequentially with saving of progress
  + ‘survival’ mode which consists of an infinite randomly generated level ‘endless run’, with the goal being to acquire as much score as possible before inevitably losing.
* Power Ups which can be stored in a player inventory and used whenever needed, with indication of how many of each the player has in his possession.
  + Gravity
  + Super-speed
  + Invincibility
  + Shooting
* NPCs placed on the campaign levels which can
  + Engage in dialogue, the player can approach an NPC and interact with him.
    - Dialogue window appears and the player can read it in the style of a Visual Novel type game.
  + Give the player scripted quests and rewards for their completion
  + Trade with the player, the player can buy quest items, a key, power up, a life etc.
* Tile-based levels
* Enemy Monsters with differing AI
  + Hedgehog – patrols from one point to another, kills the player on contact
  + Bandit – stays in one place, upon spotting the player he chase, jumping when necessary to reach the player, if the player is in range, he strikes him with his sword. He kills the player only using his sword attack, which has a short cooldown. Can be killed by the player jumping on his head.
  + FlyingMonster
* Collectibles
  + Power Ups
  + Score
  + Coins & Stars
  + Lives
  + Quest items
* High score saving and leader board system
  + The player will be able to see his highest scores.
  + If time constraints allow this will be done through an online database system.
* Animations
  + All the sprites for moving elements will be animated.
* Music and SFX
  + The game will have background music and SFX where appropriate.

2.2 Scope

2.2.1 Deliverables

Below I will write the deliverables for the release of the game, with the estimated workhours required to implement that feature. This doesn’t include the deliverables achieved by week 11.

* A fully playable game on Android and Windows. – 20 hours
* Campaign mode – 50 hours
  + 3 fully playable levels
* Survival game mode – 20 hours
  + Infinite procedurally generated level.
* Scoring and leader-board system – 10 hours
* At least 3 power up types – 10 hours
* At least 3 type of enemies with differing AI – 20 hours
* Inventory system – 5 hours
* NPC system – 25 hours
  + Dialogue system – 5 hours
  + Trading/Shop system – 10 hours
  + Quests – 10 hours
* Basic storyline – 5 hours

Estimate time required to implement the product in the weeks 11-30 – 190 hours. All in all, essentially, I am aiming to have a complete albeit short videogame by week 30. The intended playtime of the campaign mode should be 1 to 2 hours for a player who plays the game for the first time with no prior knowledge aside from reading a basic guide. The way the game will be built is that a player who already played it will be able to complete it much quicker due to no need for exploration.

2.2.2 Risk identification

Every software development project has risk involved. Software development projects can fail or be delayed due to unforeseen factors such as unexpected difficulties in implementation, badly defined system requirements, wrong estimates of project resource requirements etc. As the project gets bigger more time will need to be spent towards maintenance and code clean up to avoid failure.

To mitigate these risks when planning my deliverables my focus was on under-promising and over-delivering. Up to week 11 I had focused on the Android version, but after getting my MVP out I had realised that focusing on Windows would be better due to the Windows version being better at debugging and testing. The Android emulator or device simply doesn’t run as quickly as the Windows android studio debugger. So the change of focus is necessary in speeding up the testing and development process. New features would be added and tested initially on the Windows version and only later integrated and tested on Android. I will use Jira to plan my weekly work. I also will be kept focused on the tasks by meeting weekly with my project supervisor and deciding with him what aspect of the project I should focus on each week.

Many project managers overestimate their team’s capability and downplay the possible risks that is why my plan will be realistic in its estimation of what I am capable in eleven weeks. It is better to be conservative in planning and be pleasantly surprised rather than be over-optimistic and then be disappointed in the result.

Project Requirements

Design and Implementation

Summary of Technical Documentation

(link to gitlab detailed technical documentation)

Project Management

Conclusion

References

[1] S. A. Arnab. (2012) “The Design Principles for Flow Experience in Educational”. *Procedia Computer Science* 15 ( 2012 ) pg78 – 91. Available:

<https://www.sciencedirect.com/science/article/pii/S1877050912008228?via%3Dihub>

[2] K.R, Robson, “Is it all a game? Understanding the principles of gamification”. *Business Horizons.* Vol 58, issue 4, pg 411-420, Available:

<https://www.sciencedirect.com/science/article/pii/S000768131500035X>

1. K.B, Becker, “Teaching with games: the minesweeper and asteroids experience”. *Journal of Computing Sciences in Colleges.* Vol. 17, issue 2, pg 23-33, Available: <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.473.5072&rep=rep1&type=pdf>
2. C.J.C, Cai “Adapting Arcade Games for Learning”. *Extended Abstracts on Human Factors in Computing Systems*. Pg 2665-2670, Available: <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.294.5335&rep=rep1&type=pdf>

[5] L.S, Laila Shoukry, “Learning Analytics and Serious Games: Trends and Considerations” *Proceedings of the 2014 ACM International Workshop on Serious Games.* Pg 21-26 Available: <https://dl.acm.org/doi/pdf/10.1145/2656719.2656729>

[6] USgamer (2016, Jan. 1) by Jaz Rignall, Available: <https://www.usgamer.net/articles/top-10-biggest-grossing-arcade-games-of-all-time>

[7] GAMERANT (2020, Sep. 11) by John Rinyu, Available:

<https://gamerant.com/best-90s-arcade-games/>

[8] Nintendo Available: <https://www.nintendo.co.jp/ir/en/finance/software/ds.html>

[9] H.B-L.D “Playing Different Games on Different Phones: An Empirical Study on Mobile Gaming“. *Proceedings of the 10th international conference on Human computer interaction with mobile devices and services.* Pg 391-394 Available: <https://dl.acm.org/doi/pdf/10.1145/1409240.1409296>

[10] BadLogicGames Available: <https://libgdx.badlogicgames.com/features.html>

[11] Rise.Global Available: <https://www.rise.global/>

[12] SFML frequently asked questions Available: <https://www.sfml-dev.org/faq.php>

[13] Pygame Available: <https://www.pygame.org/wiki/about>