

Cadet Chronicles: March to the Arch - Requirements Analysis Document

1. Project Overview

a) Background/History

Platformer games are a classic genre of game that require players to move around in an environment to achieve some game-specific goal. Dating back to the first platformer released in 1980, Space Panic, which required players to move across platforms, climb ladders to access other platforms, and defeat aliens in a simple stock life and timer system, this genre of game exploded in popularity by the early 2000's. The genre-defining game, and one of the most iconic games of all time, Super Mario Bros, was released in 1985 and serves as the blueprint for Cadet Chronicles: March to the Arch.

In this 2D, side-scrolling platform game, players will have to explore levels which progressively increase in difficulty to defeat insurgents and collect the required items needed to pass the level. Players start with a certain number of lives and may gain additional lives by collecting a certain number of the required tokens. Players will lose a life if they are defeated by insurgents or fall off the map. The game ends when the player runs out of lives, resetting the game and taking them back to the first level. And if time permits, players may be given the option to collect power ups that introduce new mechanics and enhance gameplay throughout the game. For example, if we find a speed item throughout the level, the player could gain a speed increase by a certain percentage.

b) Purpose

The purpose of our project was to emulate the timeless gameplay of Super Mario Bros in a modern, Royal Military College of Canada (RMC) inspired manner. We sought to add unique features that would appeal to RMC cadets who have experienced the challenges of being a cadet that we integrated throughout the game. The level design was inspired by the 4 pillars of RMC with each level requiring the players to collect certain items related to the academic, bilingualism, fitness, and military pillars.

c) Scope

The scope of our project describes the features of our project. Within these features, we have our core features, which are required to create a functional game, and also additional details that would enrich the user's interaction with the system.

Core Features:

- Functional, intuitive GUI
- Key inputs
- Basic movement
- Simple graphics
- A playable level
- Score keeping
- Obstacles

Additional Features

- 4 levels with level selection
- Skins
- Timer for levels to track scores
- Death / Game Over scenes
- Enemies
- Collectible Items

Time Permitting Features

- Appealing graphics
- Tutorial level for new players
- Sounds/Musical elements
- Power-up mechanics
- Multiplayer functionality
- Hidden features/collectables (Easter Eggs)
- Level transition scenes
- Puzzles

d) Target Users

- Individuals who enjoy games like Super Mario Bros, particularly RMC cadets.

e) Objectives and Success Criteria

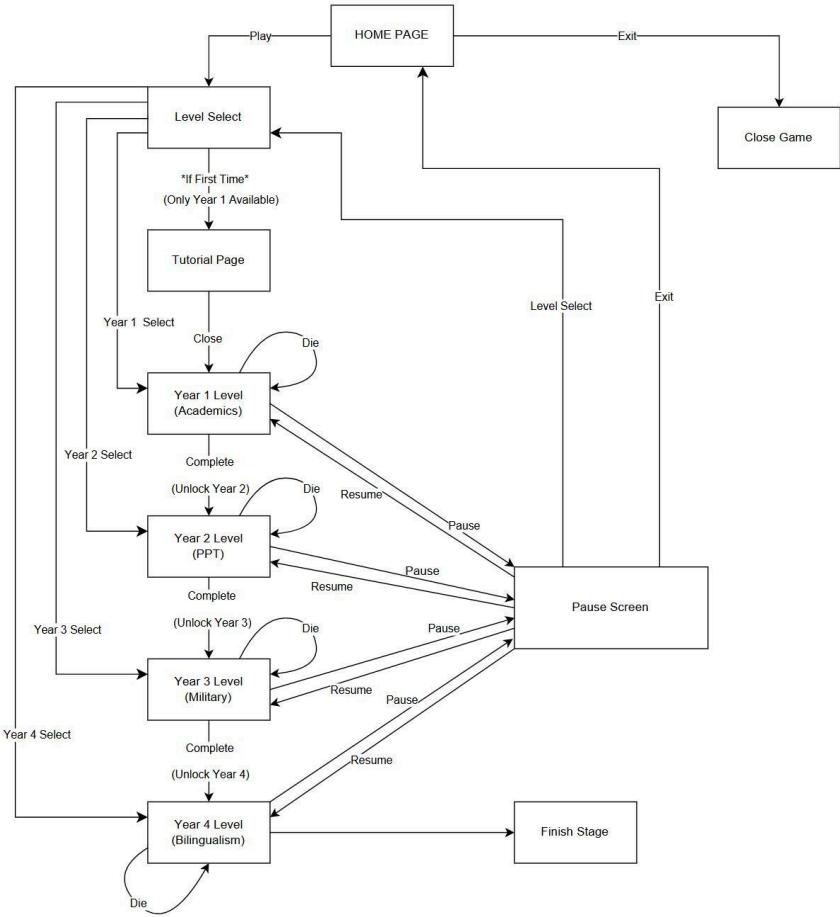
- Implement 4 different levels
- Implement different skins for each level
- Implement a satisfying movement system
- Implement a collectible and power up system
- Develop different enemies and obstacles for each level
- Develop engaging level design
- Develop a working GUI

2. System Description

a) Definitions and Acronyms

- *Level*: Unique map with its own theme, skin, enemies, puzzles and obstacles
- *Skin*: Uniform worn by the player
- *Power up*: Item that gives the player a powerful ability
- *Enemies*: NPCs that can cause damage to the player
- *Obstacles*: Stationary elements that can cause damage to the player or block players from advancing in the environment
- *Puzzles*: Areas in the map that require the player to use the environment in order to proceed
- *Collectibles*: Items scattered throughout the environment that can be collected by the player
- *GUI*: graphical user interface
- *NPC*: Non Player Character
- *Player*: Person who is playing the game

b) Layout



c) Parameters

- Level select: Each of the 4 levels in the game represents a difficulty. Level 1 is the easiest level with simple puzzles, slow enemies, and few obstacles. With each subsequent level the difficulty increases with the last level of the game representing the hardest difficulty.
- Power-ups: Throughout the game, different items will be spread across the level, and finding one will increase chances of completing the level optimally.

d) Input/Outputs

- Mouse Inputs
 - User can start the game
 - User can select a level

- User can close the game at any time
- Keyboard Inputs
 - User can use S and D to move forwards and backward in the level
 - User can use space to jump
 - User can use escape to return to the main menu
- Screen Outputs
 - The program will output a GUI
 - The GUI will contain
 - A home screen with a start and quit button
 - A tutorial screen prior to the first level
 - An overlay of number of collectables, timer, and lives during gameplay
- Speakers Outputs (Extras)
 - Background music
 - Sound effect for collecting an item
 - Sound effect for hitting an enemy
 - Sound effect for failure
 - Sound effect for success

e) Assumptions/Constraints

- Assumptions
 - Team will meet once per week to discuss features
 - Team members will dedicate 4-6 hours of work each week on the project
 - Each member will focus on their specialized role in the project
 - Members will contribute towards the final vision of the game
- Constraints
 - The project will be created using Pygame
 - The project must be completed by April 9th 2024
 - Team members are not expected to work full-time on this project

3. Requirements

a) Functional: features/goals.

Must Haves (these are the top priority requirements)

- As a user, I want to have a GUI so that I can interact with the game using a mouse, keyboard, and trackpad.
- As a user, I want the game to indicate I have completed the level so I know to move on to the next one.
- As a user, I want the game to indicate the game is over so that I know if I have won or lost.
- As a user, I want to be able to exit the game easily so that I can leave whenever I want.
- As a developer, I want players to proceed to the next level only if the previous level has been successfully completed so that the player encounters the right difficulty at each level.

- As a user, I want my character to be able to perform different actions such as jumping, crouching, and switching sides so I can experience a satisfying movement system.

Should Haves (these requirements are not strictly mandatory, but they are highly desirable)

- As a user, I want to see the difference in motion of Mario when I collect coins as opposed to being hit by obstacles so that I can visually tell the difference between the two.
- As a user, I want to feel a sense of progression as I complete a level so that I stay engaged during the entire level.
- As a user, I want to see my total score at the end of each level so that I can see how well I did.

Could Haves (if time allows, these requirements will be realized as well)

- As a user, I want the game to keep track of my best time of completing each level so that I can try doing better.
- As a user, I want a wide range of mobility options when traversing the map so that I can get around obstacles faster.
- As a user, I want to be able to access a tutorial whenever so that I can learn the game mechanics.
- As a user, I want to collect as many power-ups so that I can increase my chances of completing the game optimally.

Won't Haves (these requirements will not be realized in the present version. They are recorded though.)

- As a user, I want many different levels to play, including difficult Boss levels so that I can test my skills.
- As a user, I want to attack/defend against enemies so that I have other options rather than just avoiding them.

b) Non-functional: how well the system does what it does.

Must Haves

- As a user, I want the game to be intuitive and easy to pick up so that I can enjoy playing right from the start.

Should Haves

- As a user, I want the game to start within 5 seconds of selecting a level so that it minimises annoying delays.
- As a user, I want the game to have appealing visuals so that it is interesting to look at.
- As a user, I want the level to restart 2 seconds after I die so that I can get right back to playing.

Could Haves

- As a user, I want the game to run on Windows and Mac OS so I can play it on most laptops or computers.

Won't Haves

- As a developer, I want the game to be available to download off the Internet so that it is accessible to a wider audience.

4. Prototypes of GUI Interface



Figure 1. Home Page

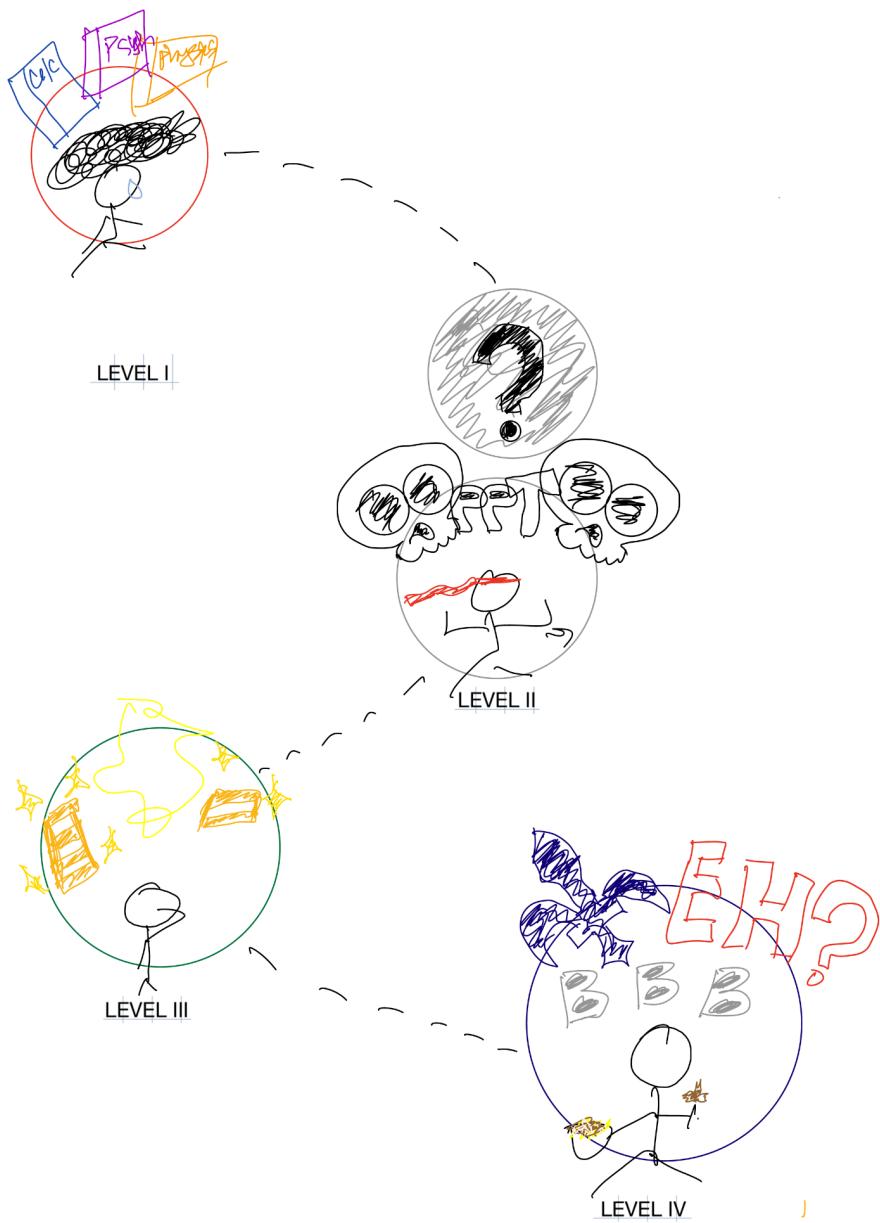


Figure 2. Level Select

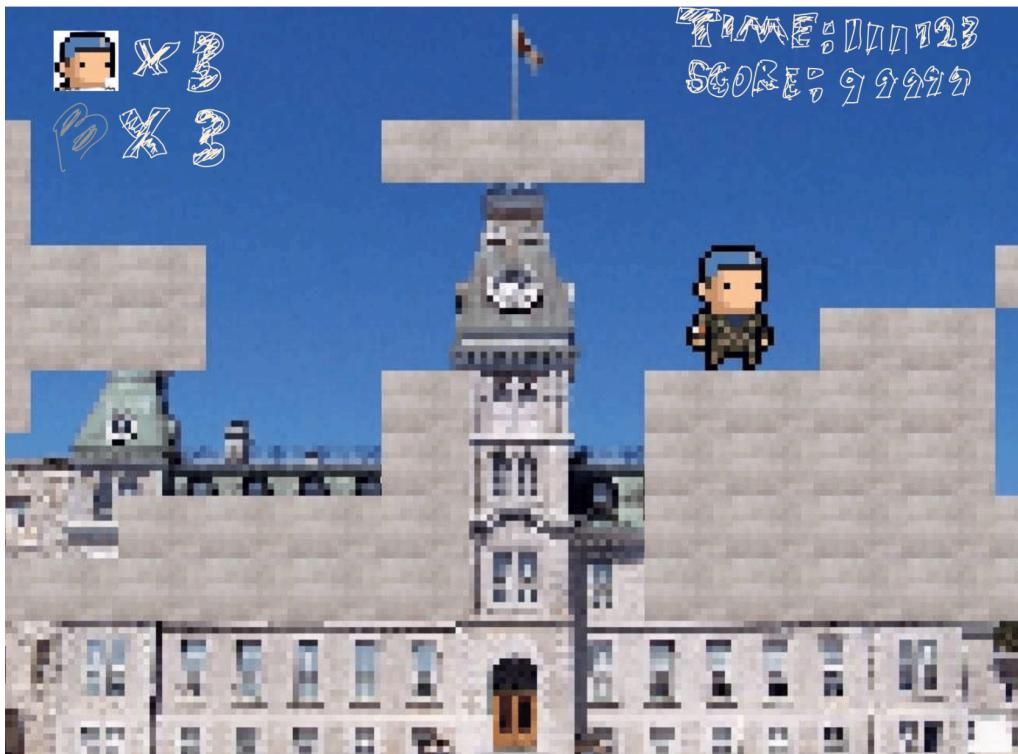


Figure 3. General Game Overlay



Figure 4. Game Result Page

5. Team Name and Roles

- Team **RJS**: RMC Junk Software

Team Member	Role
Rieon Ji	Project Manager, UI/Graphics Designer, Programmer, Product Tester
Rishi Toranagal	System Architect, Programmer, Product Tester
Justin Norman	Chief of Programming, Sound Designer, Github Manager, Product Tester
Joshua Kalhous	Scrum Master, Level Designer, Programmer, Product Tester
Seth Gillingham	Github Manager, UI/Graphics Designer, Programmer, Product Tester
Esther Kim	Note-Taker, UI/Graphics Designer, Programmer, Product Tester