Miles Ferro

A game following a software engineer through his past to find out how he ended up with his dream career

---

This is our software requirement specifications for Miles Ferro

---

Written and Prepared by
Issac Geer
Robert VanScoik
Joe H
Maxwell Maitland

For
CS360 Software Engineering
Dr. Inukollu
Mr. Kashyab Ambaran

Introduction

Miles Ferro is a game where you will play as and follow Miles Ferro through his past where he revisits how he ended up in his dream job of being a software engineer. In the first act Miles recalls his first taste of software engineering at his grandparents. You as the player will play as Miles where you discover what adventures he goes through after watching a cartoon full of superheroes and controlling their mechs to defeat their enemies. As Miles ferro you will venture through various lands right inside your grandparents house collecting pieces to your own mech and assembling the Lower Body, Upper Body, and acquiring your very own Energy Saber. While exploring the various places in your grandparents house you will need to complete puzzles, overcome challenges, and defeat enemies who are controlled by something lurking in the basement. Once your suit is completed with the help of your grandparents and cousin you will have enough courage to defeat the dangers in the basement for one last battle to reclaim the home of your grandparents.

Purpose and Features

The purpose of Miles Ferro is to teach the player about key ideas of software engineering and how they can implore those ideas to overcome many challenges. We intend to tackle things such as team building, resource management, planning, and more. Features we intend to include in the game are as followed:

Basic Player Controls

To play a game the player will need to be able to move. We plan on making this game a 2D Isometric game so the player will have the ability to walk around, jump, crouch, and crawl.

Interaction with the Environment

We plan to implement various puzzles and items on the maps that the player can move around and use to help them complete the tasks given to them. We will also provide some items that the player can use to give them an advantage at any given time.

Inventory and GUI

We want the player to be able to collect pieces to their suit as well as consumable items so having a clean GUI and an inventory for the player to work with will be essential for them to cleanly work with those items.

Save System

With Miles Ferro being a single player we believe it is essential for the player's progress to be saved automatically at checkpoints but also manually. Though the game's first act won't be extremely large, having the freedom to stop and come back at any time will be valuable to the experience of the game.

Combat System

With a few battles being scattered around the first act as well as the final battle in the basement a functional and fluid combat system is going to be vital to the game. The combat system will transition the screen to a new scene where you and the enemies involved will be seen from a side view. During this the player will have a menu in which they can choose from various attacks or items they can use. The battles will be turn based.

Stakeholder and User descriptions

Stakeholders

Dr. Inukollu - Instructor

Determine if the developers accomplished the goals of the project and provide feedback and a grade based on that assessment.

Mr. Kashyab Ambaran - Project Owner

Evaluate the final product and determine if it meets given expectations.

Developers

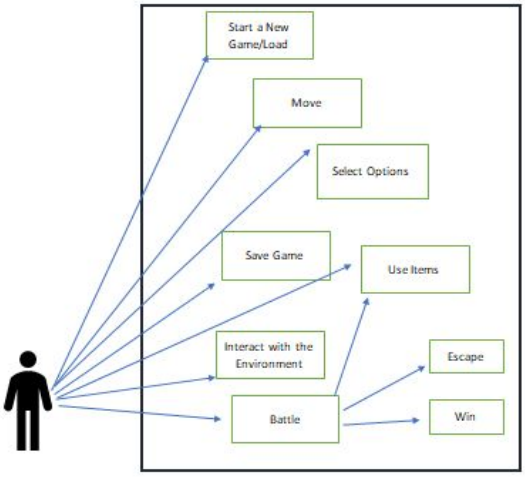
Design, Implement, Test, and deploy Mile Ferro.

Users

Player

Play through the game and optionally if available give feedback on their thoughts of the final product.

Use Case Diagram



Complete Story Board for Miles Ferro

Miles Ferro (Working Title):

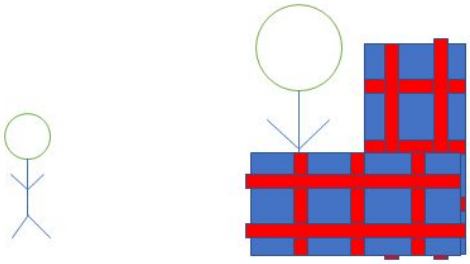
Steel Soldier

ALL ART IS PLACEHOLDER AND ONLY USED FOR EXAMPLES

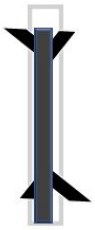
Basic Start Screen

Assuming New Game

STORY INTRO: A soldier is preparing to sortie out in a combat. They remember their childhood nearly a decade and a half ago. The war that has happened against the invaders made that fantasy of playing around near Christmas at their grandparent's house.

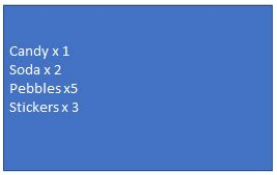


You start out in the Living Room, watching TV with your Grandparents The game is set at an Isometric view instead of a direct top down or side view.



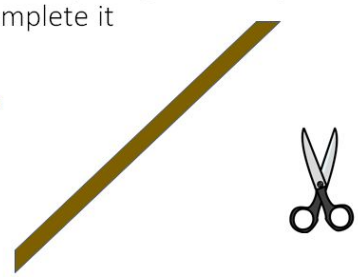
MENU SCREEN/PAUSE SCREEN

Basic screen, Selecting a menu will bring you to a sub menu. Standardized menu movement expected of an RPG

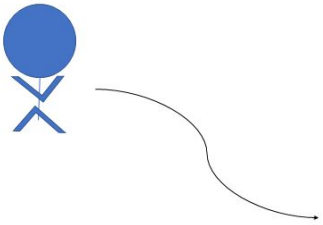


You get a quest to some items to make your suit a'la the Mechazord/Gundam/Voltron/Pacific Rim. Your grandparents act as quest givers and you'll return to them to complete it

But First, a Carboard Roll to become your Sword. This will act as your weapon during battle.



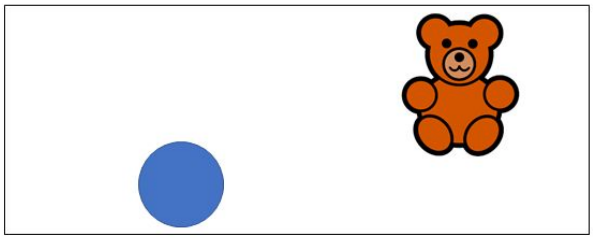
After you watch the short intro and get your first quest, you'll be free to move around the Living Room/Kitchen



This is a safe space, with enough room to move freely but allows the players to navigate and interact with obstacles in the environment.

Clicking the button that the player has used to start the game or scroll through text is the normal interaction button

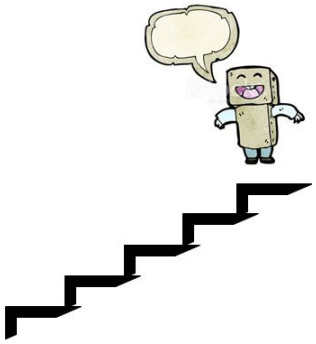
Once you get the cardboard tube, a test battle begins in your character's mind. The enemy is a teddy bear/training dummy



In battle you select attacks from a menu, akin to old school RPGs. No active battle system, but items and other resources are limited.

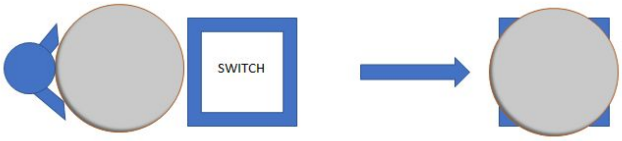
Once the player has moved around and completed the Cardboard Tube Quest, they set out to a new floor on a quest to get another item for the armor piece.

Heading upstairs, they hunt for a large enough cardboard box to make a chest piece.



Once your character gets to the 2nd floor, the map shifts and your imagination changes what was a normal room to a Forest or a Beach

Once there, they begin solving basic environmental puzzles like sliding ice and moving boulders as well as basic combat.



After Exploring a few areas and completing your quest, you end at the basement. Ready to fight the most villainous evil boss. . . The Cousin Who Helped You.

Important Backend Features:

These requirements enhance end user experience and make playing Miles Ferro something the user wants to do. The more engagement we build with the player, the happier and more invested they can be while playing this game. While some players can overlook subpar art, story, or gameplay, poor responsiveness and engagement can turn an enjoyable experience into a frustrating one. With this in mind we will make sure the game has all of the following features to ensure that the player will be able to pick up the game easily and quickly.

Cohesive and Understandable Story

A clear and cohesive story will help immerse the player in the world we will be building around them.

Responsive Menu/GUI/Movement

When features intended for player interaction feel clunky an enjoyable experience can quickly become frustrating.

Clear and Enjoyable Visuals

Good visuals in the form of game art and menu organization help keep the player attentive to what's going on in the game.

Stable Performance

Making the game stable will make sure the player doesn't get sucked out of the fantasy we are trying to push.

Maintainable/Reusable Code

The code must be built in an organized and intentional way so that if we wish to provide updates in the future we can do so without breaking major parts of the game.

Tools, Technology, and Risks

Miles Ferrow will be built using Unity's 2D Game Engine that uses C# as the programming language. The engine provides many features we can use to build the game in the required time frame. We will be using GitHub for version control and collaborative writing of the code. Some risks we are taking while working on this game are as followed:

- Our limited knowledge of the specifics of both Unity's 2D Game Engine as well as C#, if we can't learn these in a timely manner then there's a good chance some important features are missed.

- GitHub management. GitHub is another tool we are using that some of us are not very familiar with, there is a risk where some version of our project might be overwritten due to a mistake.

- Lastly team communication is a risk as well since we are only meeting 1-2 times a week for about an hour at a time there is a chance someone misses a meeting or doesn't remember important information.

Production Schedule:

[Task - Due Date]

SRS Document - Feb. 28, 2021

SRS Presentation - March 4, 2021

Pre-Development - March 11, 2021

C# and Unity Study - March 14, 2021

Begin Development - March 15, 2021

First Sprint Document - March 28, 2021

First Sprint Presentation - March 31, 2021

Finalize Level Development - April 8, 2021

Finalize Key Feature Development - April 20, 2021

Testing and Bug fixing - April 23, 2021

Finalize Extra Feature Development - April 25, 2021

Testing and Bug fixing - April 27, 2021

Final Presentation - April 28, 2021

Final Sprint Document - May 3, 2021