CS2 Semester Long Project Writeup

2D, Tile-based, Single-player, Side-scrolling, Platformer Game

Eric Ragle

CISP 1020 K70

#### Table of Contents

[Inspiration 3](#_qklgtlf4gwvj)

[How it will achieve its goals 3](#_wlind5f1b8m)

[Consistent Coding Style 3](#_ebh4ht7wjy6q)

[Searching and Sorting Algorithms 3](#_1xi99i6jk517)

[String Manipulation 3](#_580shgul1pfw)

[Inheritance and Polymorphism 4](#_tfffd0i95ae0)

[Design Patterns 4](#_3fr5h3uhzt0p)

[Proper Error Handling 4](#_l4nspjxik2dj)

[Logging 4](#_8w18y5pfu41m)

[Text/Binary File Manipulation 4](#_oc3wcvmetnay)

[Graphical User Interface (GUI) 4](#_v0xi5ed7c4ry)

[LINQ on a C# Data Structure 4](#_hwgw9yuq67r8)

[State Machine Diagram 4](#_14llx66eg1ze)

[Deployment Diagram 5](#_f2yu8nr0sjt8)

[API List 6](#_vqq0cong1qa1)

### Inspiration

My game will take inspiration from freeware platformers like *I Wanna Be The Guy*, *Shobon no Action*, and *The Big Adventure of Owata's Life.* For this initial prototype, I have made the decision to not make it as difficult as these titles due to my lack of knowledge in game design.

### How it will achieve its goals

#### Consistent Coding Style

The best way to achieve a consistent coding style in your work is to use a style guide. I will be frequently referring to Microsoft's official guidelines for C#. I will also make sure to use standard naming conventions for all aspects of my program, such as camelCase for variables and PascalCase for classes and methods.

#### Searching and Sorting Algorithms

My plans for the implementation of algorithms may change as my understanding of them improves, but as of now I plan to use sorting algorithms for level progression and searching algorithms for collision detection.

#### String Manipulation

I will use string manipulation when displaying instructions and level titles.

#### Inheritance and Polymorphism

One of the ways inheritance will be used in my game is through tile types. Different types of tiles will have different properties and interact with the player in distinct ways. The different types of tiles will inherit from a parent tile class. This makes it easy to manage them and their unique properties. This is also an example of polymorphism because the tiles will each behave differently than their parent class.

#### Design Patterns

I will use a global state design pattern where my entire game is managed in a central system. This seems the easiest to me. This too may change as I develop a more in-depth understanding.

#### Proper Error Handling

I will make sure my game uses proper error handling. If a resource such as a texture is missing, the game should detect this and provide a warning message instead of crashing the game.

#### Logging

My game will use error logging for troubleshooting purposes. If there is a crash or error detected, a .txt file will be created.

#### Text/Binary File Manipulation

I plan to use a binary file to save and load progression information/player states. Binary files are more efficient than text files.

#### Graphical User Interface (GUI)

There will be multiple examples of GUIs integrated into my game, such as a title screen and dialogue cutscenes.

#### LINQ on a C# Data Structure

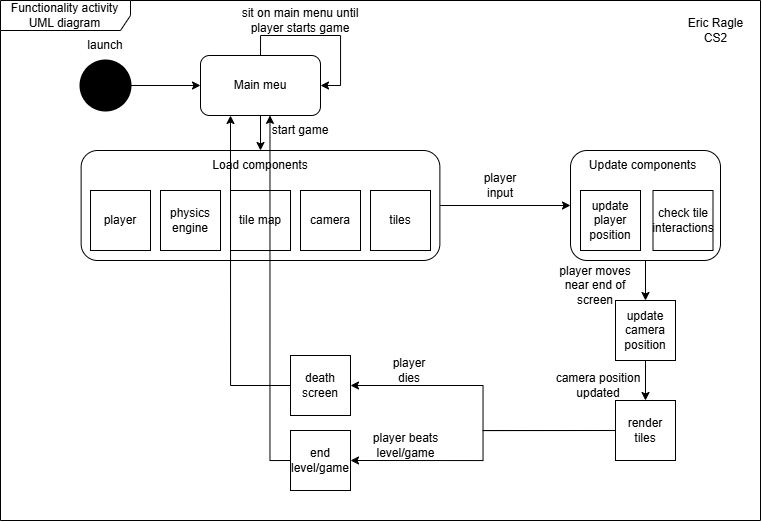
This is something I know nothing about and will plan and prepare for when it is covered in class.

### Expected Outcome

The expected outcome is a basic, 2D, tile-based, single-player, side-scrolling, platformer game with text-based story telling and dialogue.

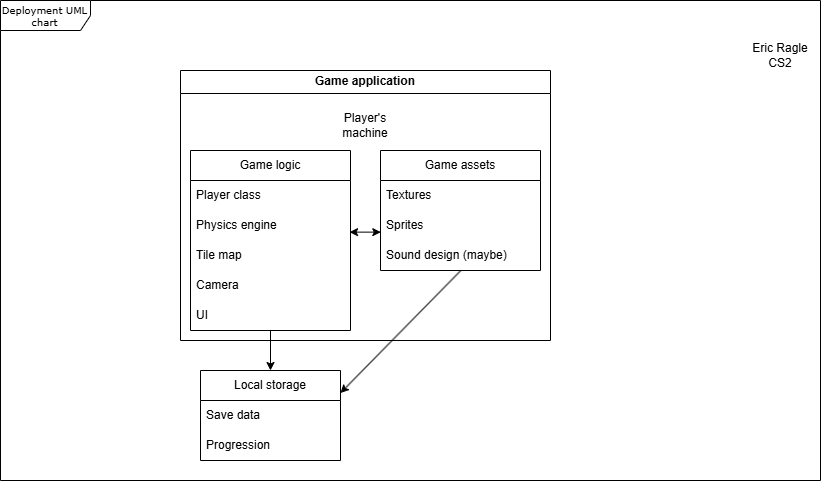
### State Machine Diagram

Here is a state machine UML diagram of the basic functionalities of my game.



### Deployment Diagram

Here is a deployment UML diagram of the physical deployment of my game.



### API List

As of now, the only API I plan to integrate is Unity’s 2D toolkit.