Logo

Description automatically generated

**22/FA-COP-2939-68831**

**Capstone Project**

**Detailed Design**

Created By: Johnathan Webster

Student Number: 1040332

Instructor: Dr. Banisakher

Date: 10/22/2022

Table of Contents

[1. Overview 3](#_Toc117327078)

[1.1 Basic Structure 3](#_Toc117327079)

[1.2 Basic Routine 3](#_Toc117327080)

[1.3 Player Controls 4](#_Toc117327081)

[1.4 Dimirti: Player\_Controls 4](#_Toc117327082)

[1.5 Player Direction 4](#_Toc117327083)

[1.6 Player Jumping 4](#_Toc117327084)

[1.7 Player Physics 5](#_Toc117327085)

# Overview

The purpose of this document is to further expand upon the preliminary design. The document will provide a basic outline of the application and expand upon all aspects of the design.

# Basic Structure

Diagram

Description automatically generated

# Basic Routine

The application begins on the first frame executed by the game. From that first frame the application calls Delta every second sixty times. Delta is used as the core timing for the application. All core functions rely on the delta for timing purposes to provide accurate and consistent positioning of the player character and any objects the player could interact with.

# Player Controls

Player controls are foundation for interaction between the application and the end user. The player controls keep’s track of the position the player is currently in. By calling this method every second sixty times a second the end user will have a smooth and responsive experience with the player character.

# Dimirti: Player\_Controls

Dimitri inherits from the player controls class. By inheriting the position of the player, we can now implement a more realistic feeling by processing the player direction, gravity, and physics. We pass delta as a parameter to the function to synchronize with the internal timer we declared earlier.

# Player Direction

Player Direction is responsible to tracking the direction the player is currently moving, if the user is pressing a button, and when they are not. This will allow the application to respond quickly and somewhat seamlessly to player input.

# Player Jumping

Player jumping is a state tracker. The purpose is to monitor with the player is in the air. If the player is in the air, they have different controls they are allowed to interact with.

# Player Physics

Finally, we have player physics. Player physics will computer all the needed physics involved with gravity, friction, momentum, acceleration, deceleration, and object contact. This will create a realistic feel when the character move across the screen and interacts with the environment around them.