**Software Requirements and Design Document**

**For**

**Group <4>**

Version 1.0

**Authors**:

Matthew B

John R

Mina A

Ethan P

Jared G

# Overview (5 points)

A 16-bit 2D platformer where the player is a monkey called Primate Ape, traveling its way through multiple levels, defeating lions and liberating levels by traveling to the end of each one. The enemies are lions.

# 2. Functional Requirements (10 points)

# The enemies walk around the level. (high)

# The enemies cause the health bar to go down when touched from the side. (high)

# There is an always-visible health bar that reflects the health level of the main character (high)

# Lions attack by shooting a bow or using a sword. (high)

# The character is controlled through user input of the keyboard (high)

# The character can jump. (high)

# The character has a static health bar at the top left of the screen. (high)

# Enemies can die. (high)

# The main character dies when the health bar runs out. (high)

# The level restarts every time the main character dies. (high)

# The camera moves as the character moves. (high)

# There's collision detection between the character and the levels. (high)

# The next level must be unlocked at the end of a level, excluding the third level. (high)

# The default ability is being able to jump on enemies. (high)

# The user can get new abilities after beating each level. (high)

# The player can jump on enemies to decrease enemy health. (high)

# There are enemies in each level that have their own behaviors. (high)

# Enemies aim towards the player. (high)

# Once life energy is collected, the health bar can be refilled. (high)

# Once a level is finished, the health bar is refilled. (high)

# Once the health bar runs out, the user starts from the beginning of a level. (high)

# Each level you beat in this platformer unlocks a new level. (medium)

# Collecting health allows the player to increase health. (medium)

# In order to beat a level, a button must be pressed. (medium)

# Several monkeys in a cage must be freed at the end of the level. (medium)

# Each level has a different design: a jungle, a desert, and a massive fortress. (medium)

# Defeating the first level unlocks a punch attack. (medium)

# Enemies only shoot projectiles when you see them on the screen. (medium)

# When an enemy is defeated, there’s a 25% chance of it dropping life energy. (medium)

# There are three levels in the game. (low)

# There is a final boss at the end of the game. (low)

# There will be a pause screen, which will pause everything in the game. (low)

# The pause screen will allow the user to either resume or exit the game. (low)

# The game will start by displaying text bubbles that describe the storyline (low)

# Defeating the second level unlocks a gun. (low)

# Projectiles from the gun cannot escape the view of the player camera. (low)

# Non-functional Requirements (10 points)

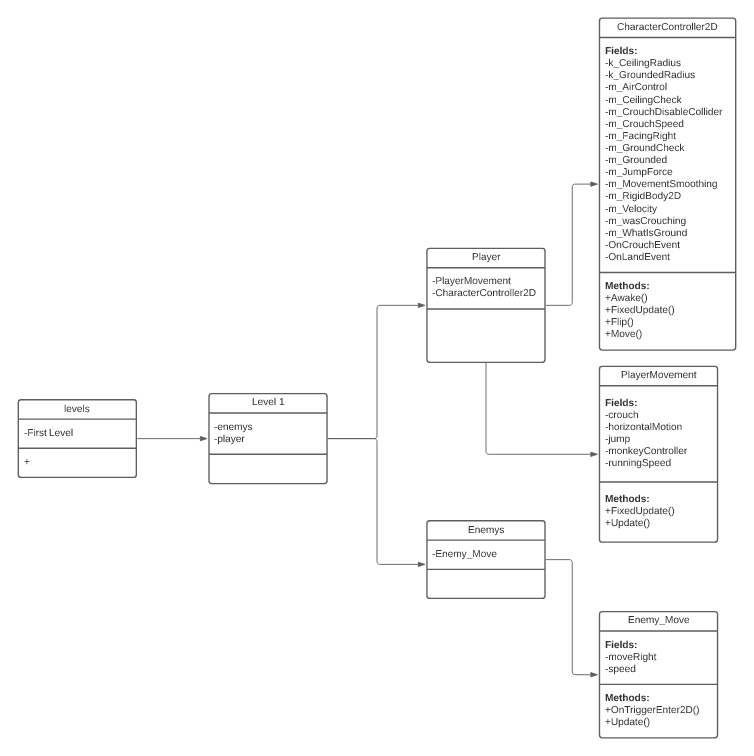
1. The game should start up in less than 5 seconds.
2. Loading between levels should take less than 5 seconds.
3. There should be no noticeable input delay (>30ms).
4. The game will be well optimized for the user.
5. The game will not crash unexpectedly.
6. The game will function on Windows and MacOS operating systems.
7. The health bar will accurately reflect the player's health.
8. Damage done to enemies will stay consistent between attacks ex: (jump deals one damage).

# Use Case Diagram (10 points)

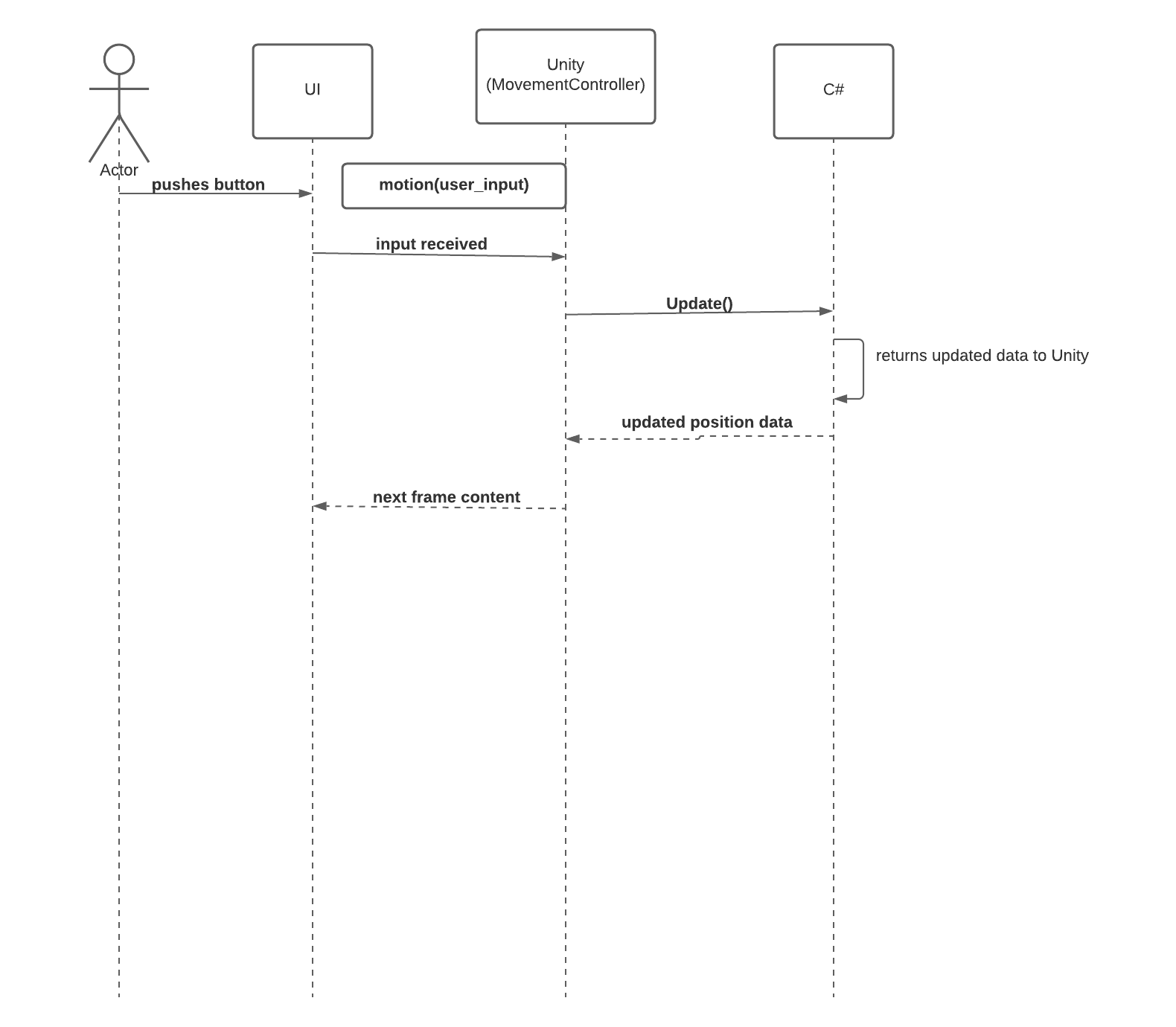


# Class Diagram and/or Sequence Diagrams (15 points)

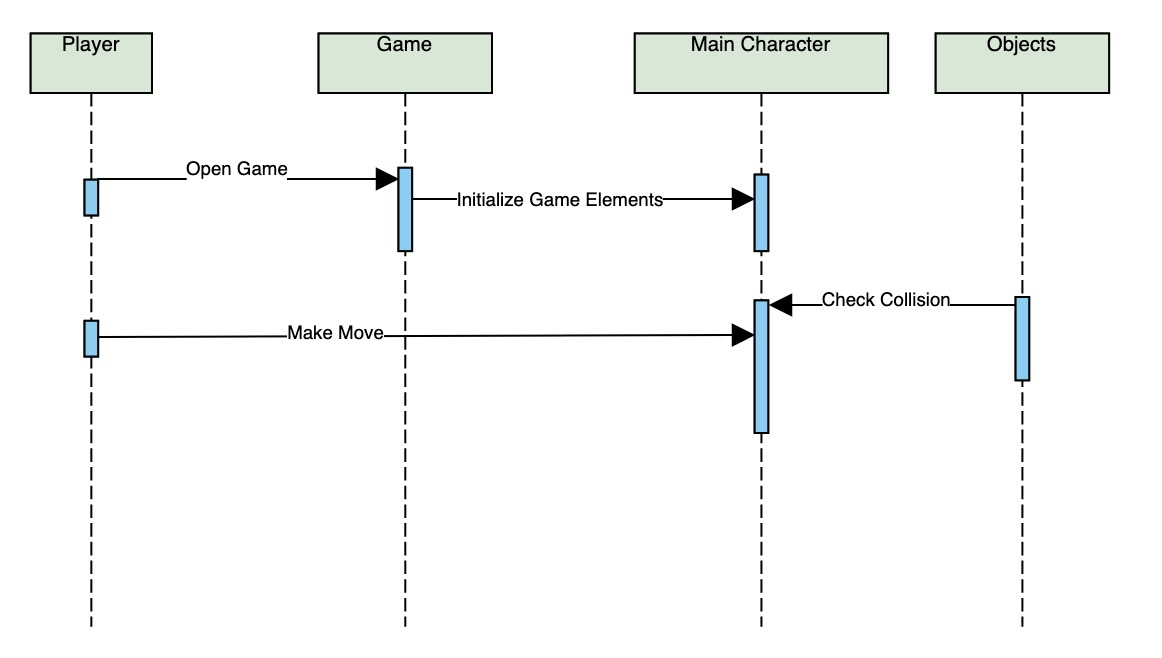
Class Diagram:



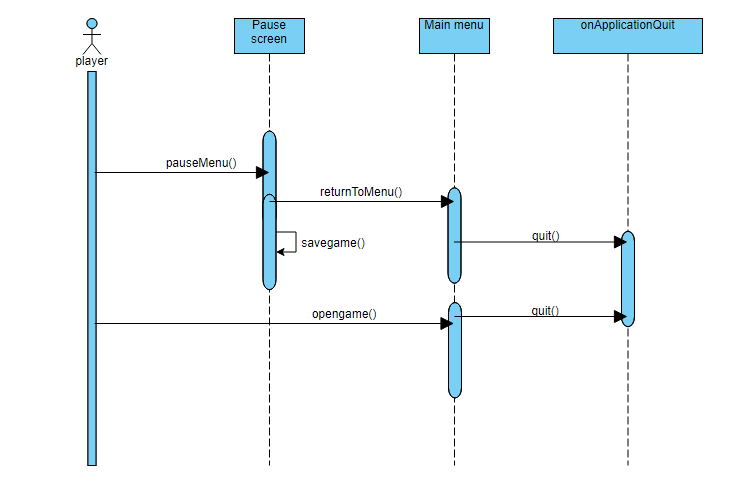
Control Character Use Case:

**

Start Game Use Case:

**

End Game Use Case:



# Operating Environment (5 points)

*Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.*

# Assumptions and Dependencies (5 points)

*List any assumed factors (as opposed to known facts) that could affect the requirements stated in this document. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project.*