Software Requirements and Design Document

For

Group 10

Version 1.0

Authors:

Reece Gabbett

Ryan Beck

Parker Stone

Timur Bickbau

Marcos Sivira

1. Overview (5 points)

The system as it stands now is still very early in development in a programming sense. We have as of now a very basic proof of concept for movement and animations. This system is written in Unity using C#. We have also begun work on a map-making script that we will implement later on to build our game world after all basic game mechanics have been written.

2. Functional Requirements (10 points)

1. The Player Movement was the first requirement we tackled as it was a high priority, this encapsulated all of Lank's movements with the ability to go in eight different directions.
2. Player Animation is another requirement that we have successfully accomplished. It was not of the highest priority but it is essential in making a smooth gaming experience.

Future Functional Requirements:

1. Player System: HP, attacks, attack-stat, defense-stat, speed-stat,
2. Inventory System: money, weapons, items, following camera, hitboxes
3. Dungeon System: enemies, randomization, loot, room camera
4. Graphical and Audio design: Animations Player model, enemy models, world textures, and item textures, music
5. Open World System: Villages with traders and shops where currency can be exchanged for items (economy system), as well as non-linear gameplay with multiple dungeons that can be entered.
6. Quality of life: Save states, main menu

3. Non-functional Requirements (10 points)

Scalability: FPS outside of the test 60 FPS can break the animations and make the game run in less than optimal ways.

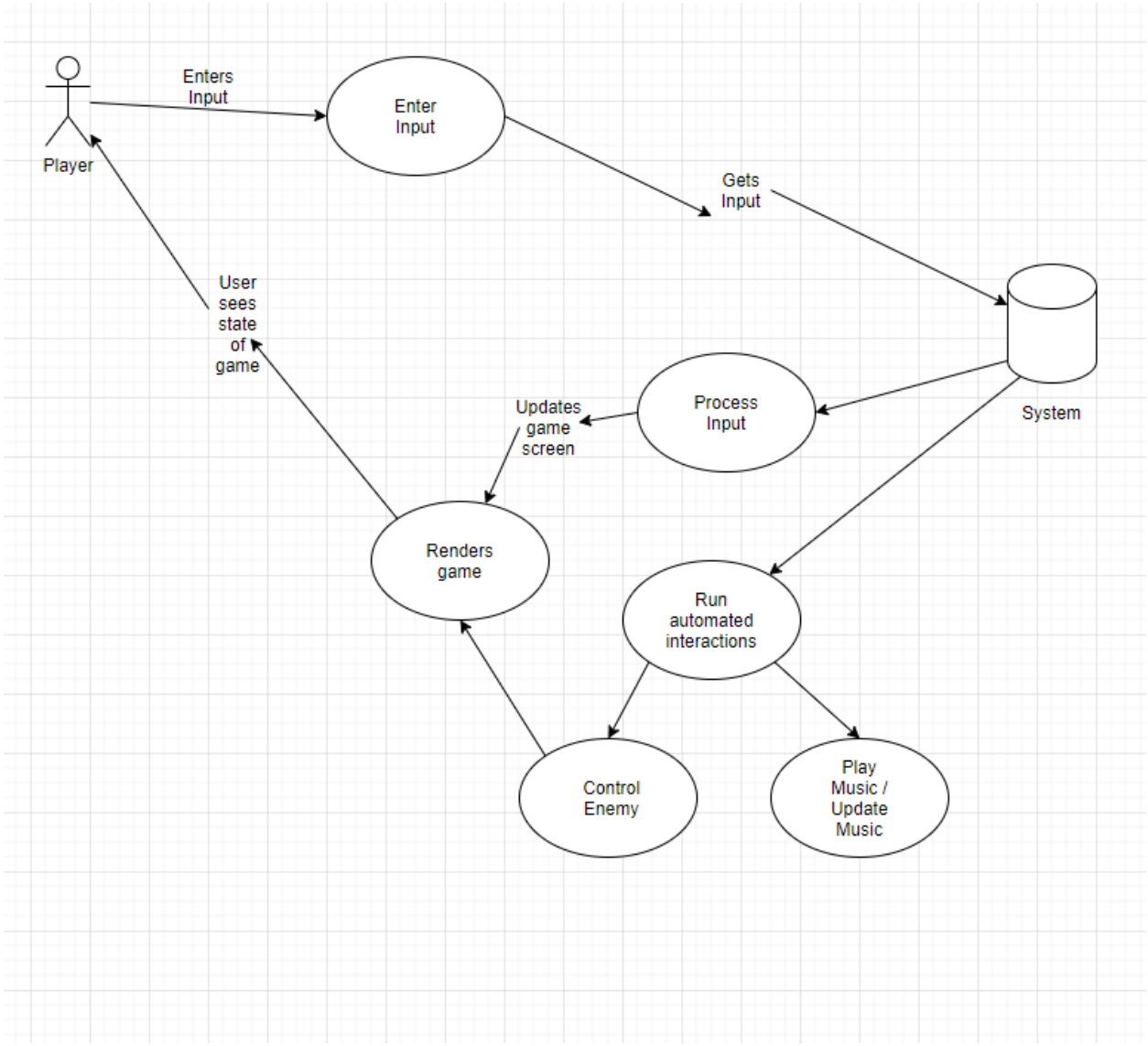
Data Integrity: Encrypt the saved data so that the user cannot compromise its integrity.

Usability: Windows 7 or newer, with keyboard and mouse.

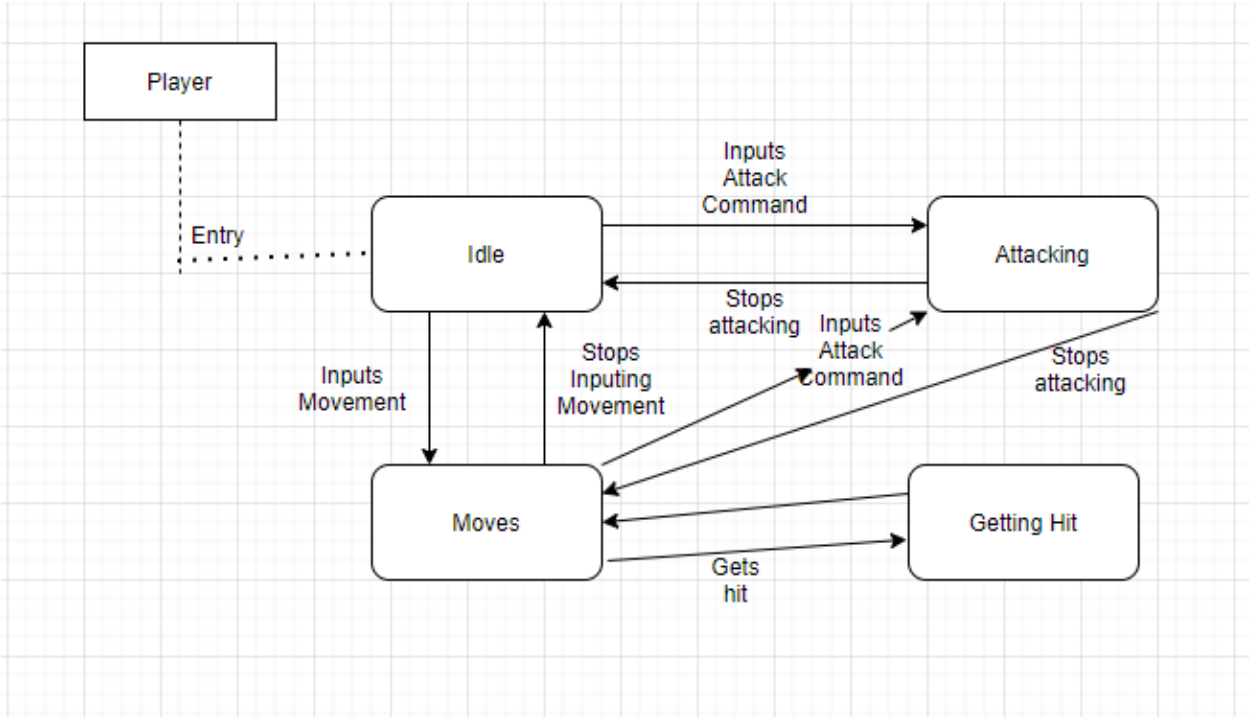
Reusability: We can reuse the assets (e.g. sprites and movements) for future installments.

Reliability: Ensure that there are no game-breaking bugs.

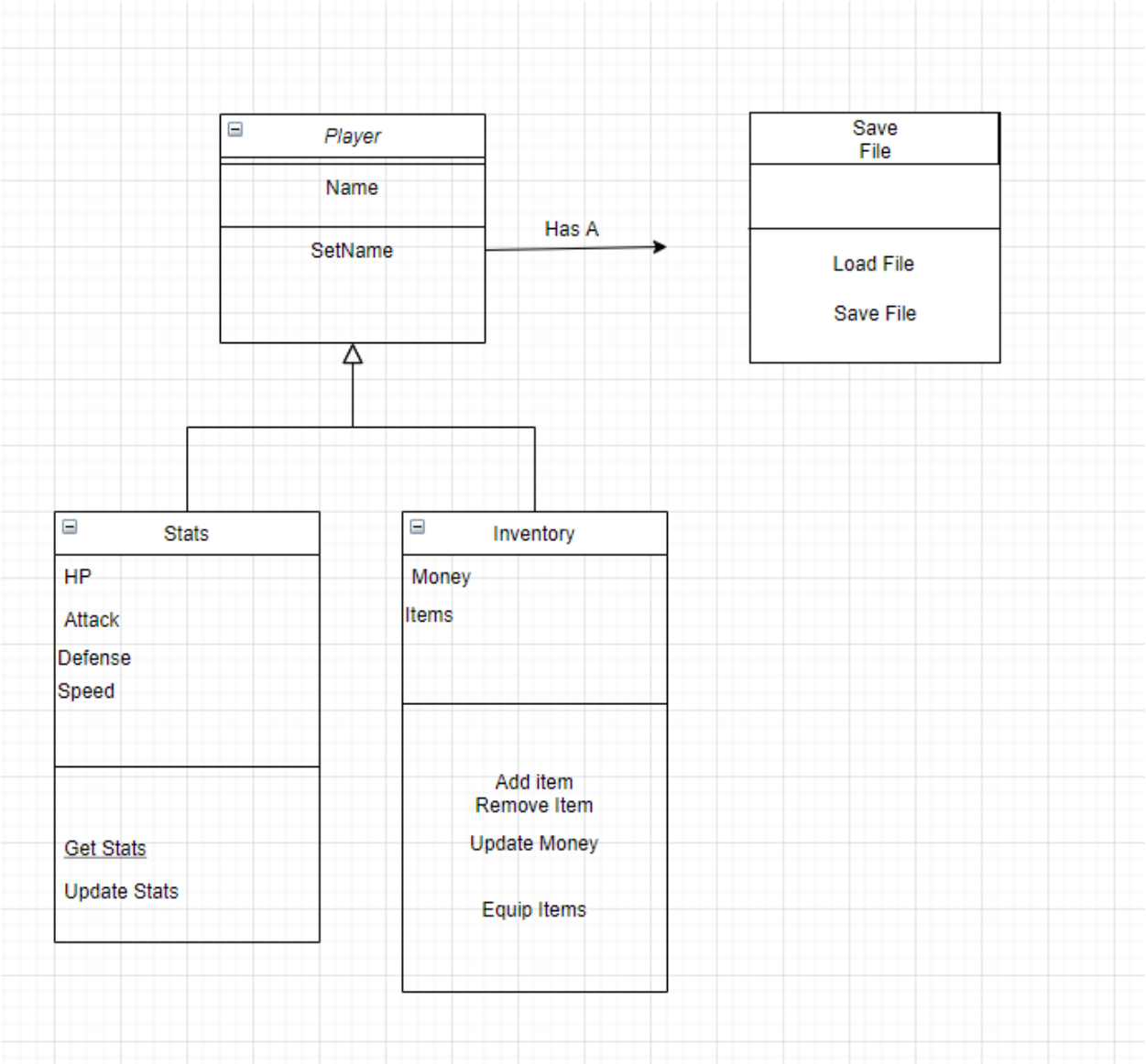
4. Use Case Diagram (10 points)



5. Class Diagram and/or Sequence Diagrams (15 points)



The Animator follows the above sequence diagram for controlling the characters' animations. Each state is a blend tree that has 4 possible animations based on the last direction inputted.



6. Operating Environment (5 points)

The operating environment shall be a computer running a Windows operating system using a keyboard and mouse control scheme. The computer will need to be running a Windows version that is supported by Unity (no older than Windows 7).

7. Assumptions and Dependencies (5 points)

We are assuming the user has a keyboard and mouse and will be using a Windows version supported by Unity (no older than Windows 7).

We are dependent upon Cinemachine for our cameras.
We are dependent upon outsourced and/or found sprite sheets as none of us have the ability to create good-looking ones.