Max Moore Mark \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/50

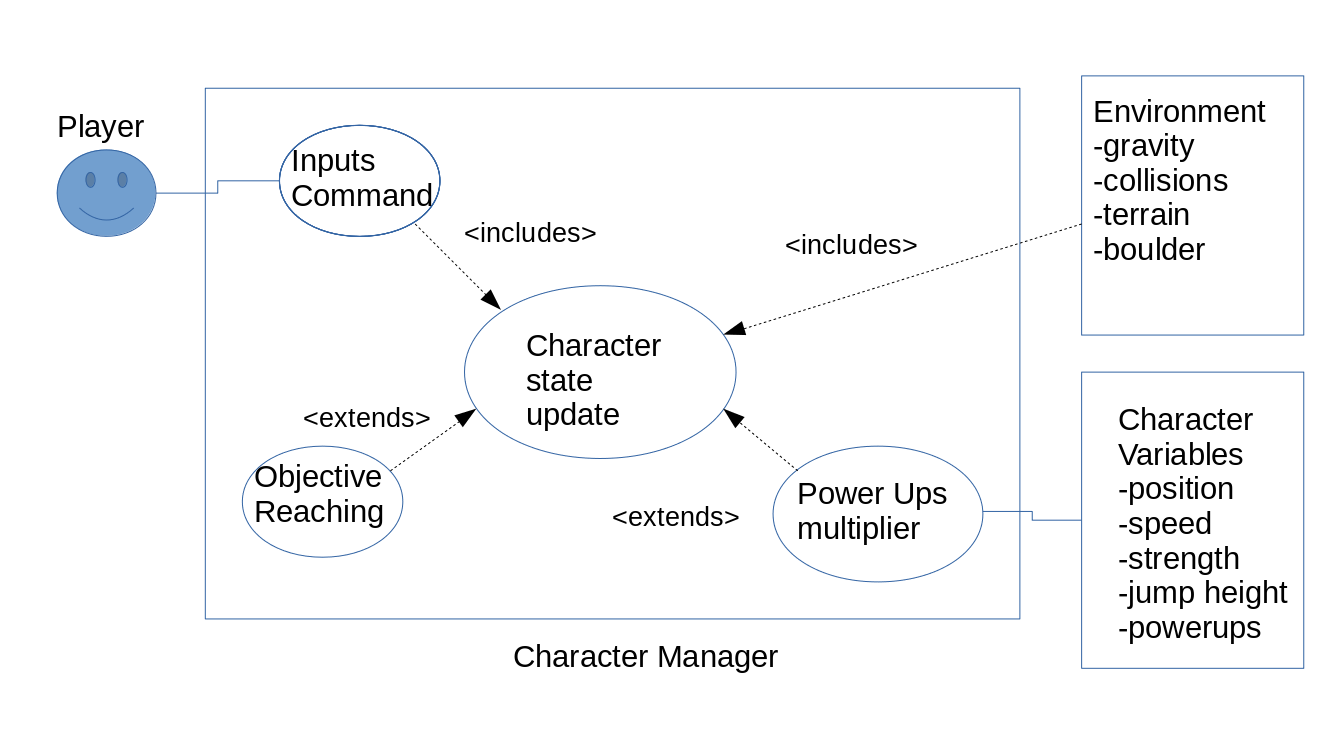
## Brief Introduction \_/3

My features for Sisyphus is the player’s interaction with the game and controlling the onscreen character. These controls are directly fed by input from either a keyboard or controller. The character will be able to push a boulder, run and jump across the map, and attack enemies. These abilities will grow with collected powerup found in the game.

Additionally, I am responsible for an experience point system, which can be used to purchase skills for the character. Whenever a player completes a goal or defeats an enemy, they gain experience points. These skills will help the player progress through the game

## Use case diagram with scenario \_\_14

### Use Case Diagrams



### Scenarios

**Name:** Inputs Command

**Summary:** Reads from player controls and attempts to grant to character’s state

**Actors:** Player

**Preconditions:** World and assets are loaded.

**Basic sequence:**

**Step 1:** Accept input as commands from player.

**Step 2:** Accept parameters from character.

**Step 3:** Determine if input can be performed by testing flags (e.g. player must be standing on ground to make Jump command legal.)

**Step 4:** Preform command on character.

**Exceptions:**

**Step 1:** Incorrect input is provided; ignored.

**Step 3:** Command is not legal; ignored.

**Post conditions:**

**Priority:** 1\*

**ID:** C01

\*The priorities are 1 = must have, 2 = essential, 3 = nice to have.

**Name:** Character State Updated

**Summary:** Distributes the command to affect the character and the environment.

**Preconditions:** Player input is legal.

**Basic sequence:**

**Step 1:** Determine the type of command (i.e. jump, attack, and move.)

**Step 2:** Call appropriate function on character.

**Step 3:** Inform environment of updated character status.

**Step 4:** Test if status resolved.

**Exceptions:**

**Step 3:** Environment is unable to support status (e.g. player is stuck.), attempt to resolve.

**Step 4:** Status was not updated; ignored.

**Post conditions:**

**Priority:** 1

**ID:** C02

**Name:** Objective Reaching

**Summary:** Senses when the player has completed an objective and is ready to proceed to the next level.

**Preconditions:** Player has reached an end goal marker.

**Basic sequence:**

**Step 1:** Determine if win conditions are satisfied.

**Step 2:** Proceed to update score.

**Step 3:** Proceed to end level.

**Step 4:** Reset player position.

**Exceptions:**

**Step 1:** Win condition not met; ignored.

**Step 3:** Game is unable to load next level, display error. Exit game.

**Post conditions:** Load next level.

**Priority:** 2

**ID:** C03

**Name:** Power Ups Multiplier

**Summary:** Augments the player’s state with additional power granted by collected power ups. Can include various things such as attack strength, jump height, and speed.

**Preconditions:** Player has collected at least one power up.

**Basic sequence:**

**Step 1:** Determine all power-ups and their products.

**Step 2:** Determine current player state and increase it by the factors found in step 1.

**Exceptions:**

**Step 1:** Player has not collected power-ups, ignored.

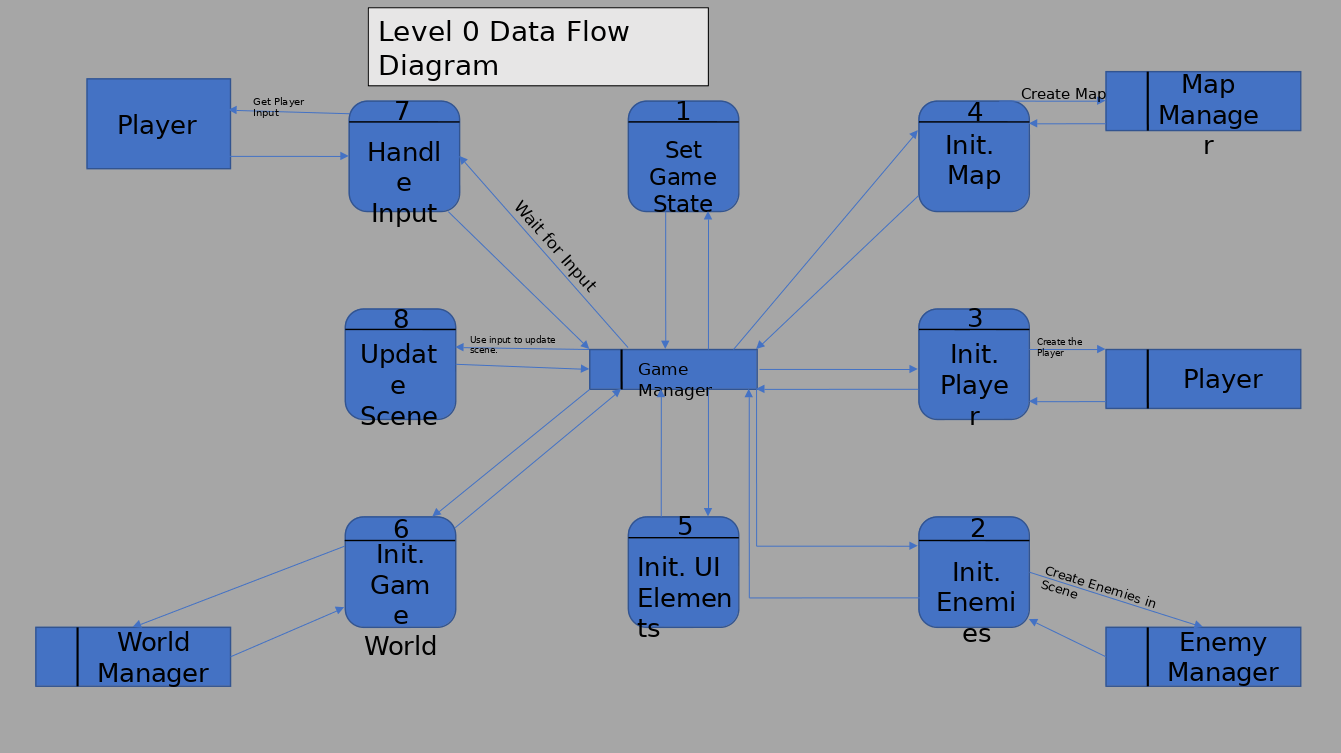
**Post conditions:**

**Priority:** 3

**ID:** C04

## Data Flow diagram(s) from Level 0 to process description for your feature \_\_\_\_\_\_\_14

### Data Flow Diagrams



### 

### Process Descriptions

7.1 Interacts with Environment:

IF player is contacting ground

set flag FALLING false

set fall speed 0

ENDIF

IF player is not contacting ground

set flag FALLING true

set fall speed 1

ENDIF

IF player FALLING FALSE and movement input given

move player in direction at speed

ENDIF

IF player is touching powerup

add powerup to player

delete powerup

ENDIF

IF player is touching goal

tell gamemanager end level

add score

ENDIF

7.2 Interacts with Enemies

IF player is attacking and enemy is colliding

damage enemy by strength amount

ENDIF

IF player is colliding with enemy and enemy attacks

damage player by enemy’s strength

ENDIF

3.1 Create player and initilize variables

IF player health is <= 0

tell gamemanager kill player

3.2 Player is set into position of new level

WHILE player is outside bounds

subtract 10000 from health

## Acceptance Tests \_\_\_\_\_\_\_\_9

The features for player movement are to be tested with a demo level including multiple terrain options and enemies. Powerups and experience will be awarded to the player to test how those systems affect player movement.

These tests should look for irregular behavior such as clipping through walls, lack of contact, or ineffective input.

**Example for Jump command**

Have the character jump off of various surfaces. Should cover all 180 degrees of floor surface, as well as uneven collisions (ledges, slopes, etc.) Should also include jumping with movement input with same parameters

Report any oddities such as:

* Clipping through walls, floor and ceiling
* Inability to jump
* Jumping with inappropriate velocity
* Jumping on thin air

**Example for Move command**

Have the character move off of various surfaces. Should cover all 180 degrees of floor surface, as well as uneven collisions (ledges, slopes, etc.)

Report any oddities such as:

* Clipping through walls, floor and ceiling
* Inability to move
* Moving with inappropriate velocity
* Moving on thin air

**Example for Attack command**

Have the character attack various objects. Include any and all world objects to be expected in game.

Report any oddities such as:

* Inability to attack
* Collisions not resolving
* Inappropriate reactions from object
* Attacks occurring too quickly

## Timeline \_\_\_\_\_\_\_\_\_/10

### Work items

|  |  |  |
| --- | --- | --- |
| Task | Duration (PHrs) | Predecessor Task(s) |
| 1. Player Creation | 5 | - |
| 2. Input Mapping | 2 | 1 |
| 3. Collisions | 6 | 1 |
| 4. Movement design | 4 | 2, 3 |
| 5. Attack design | 4 | 2,3 |
| 6. Skill tree design | 5 | 4,5 |
| 7. Power ups | 3 | 6 |
| 8. Programming | 10 | 7 |
| 9. Testing | 40 | 8 |
| 10. Documentation | 3 | 8 |
| 11. Artwork | 6 | 1 |

### Pert diagram

