Name: Samantha Heck Mark _____/50

1. Brief introduction __/3

My feature is the player controller. It accepts input from the player to move and rotate the character and cause the weapons to fire. It also triggers the game over condition if the player loses all their health.

2. Use case diagram with scenario __/14

Use Case Diagrams

Scenarios

Name: Accept input
Summary: The player gives mouse and keyboard input.
Actors: Player.
Preconditions: The game has been launched.
Basic sequence:

Step 1: Check that game state is play.
Step 2: If yes, accept input from mouse and keyboard.

Exceptions:

Step 1: The game state is not play: ignore input.

Post conditions: Input can now be used by the system.
Priority: 1*

ID: C01

Name: Move character
Summary: The character model is moved on the screen.
Actors: Player.
Preconditions: The game has been launched and the game is in the play state.
Basic sequence:
Step 1: Accept input successfully.
Step 2: Check if the input relates to movement.
Step 3: If yes, create appropriate movement vector.
Step 4: Clamp movement inside bounds.
Step 5: Move character
Step 6: Make sound.
Exceptions:
Step 1: If no movement-related input: do nothing.
Post conditions: Character moves.
Priority: 1*
ID: C02
*The priorities are 1 = must have, 2 = essential, 3 = nice to have.

Name: Rotate character
Summary: The character model is rotated on the screen.
Actors: Player.
Preconditions: The game has been launched and the game is in the play state.
Basic sequence:
Step 1: Accept input successfully.
Step 2: Check if the input relates to movement.
Step 3: If yes, rotate character
Exceptions:
Step 1: If no rotation-related input: do nothing.
Post conditions: Character rotates.
Priority: 2*
ID: C03
*The priorities are 1 = must have, 2 = essential, 3 = nice to have.

Name: Request shoot
Summary: The weapon system is activated.
Actors: Player, Weapon System.
Preconditions: The game has been launched and the game is in the play state.
Basic sequence:
Step 1: Accept input successfully.
Step 2: Check if the input relates to shooting.
Step 3: If yes, contact Weapon System and ask to activate.

Exceptions:

Step 1: If no shooting related input: do nothing.

Step 2: It no weapon system is found, do nothing.

Post conditions: Weapon system attempts to shoot.

Priority: 1*

ID: C04

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

Name: Take Damage

Summary: The character loses health

Actors: Enemies

Preconditions: The character is damaged by an enemy.

Basic sequence:

Step 1: An enemy says it has done damage.

Step 2: Subtract health.

Exceptions:

Step 1: If health less than zero, initiate game over.

Post conditions: Character loses health

Priority: 3*

ID: C05

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

Name: Report Game Over

Summary: The game ends.

Actors: Game Manager

Preconditions: The character runs out of health.

Basic sequence:

Step 1: Report game over to game manager.

Exceptions:

Step 1: If no game manager found, do nothing.

Post conditions: Weapon system attempts to shoot.

Priority: 3*

ID: C06

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

Name: Make sound

Summary: Sound is made.

Actors: Player, Asset Pipeline

Preconditions: A sound-making action has occurred.

Basic sequence:

Step 1: Request a sound from the asset pipeline.

Exceptions:

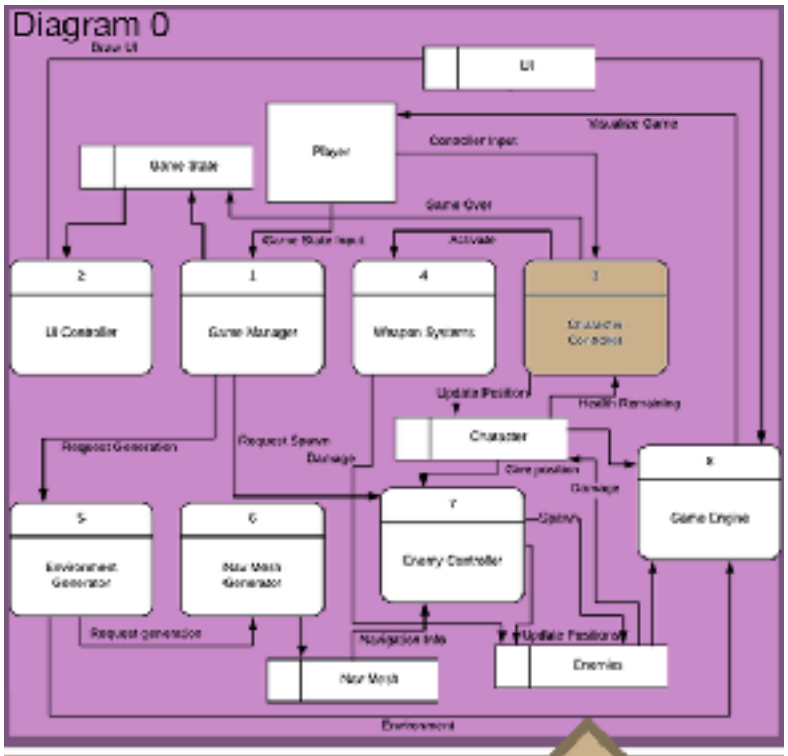
Step 1: If the asset pipeline is not found, do nothing.

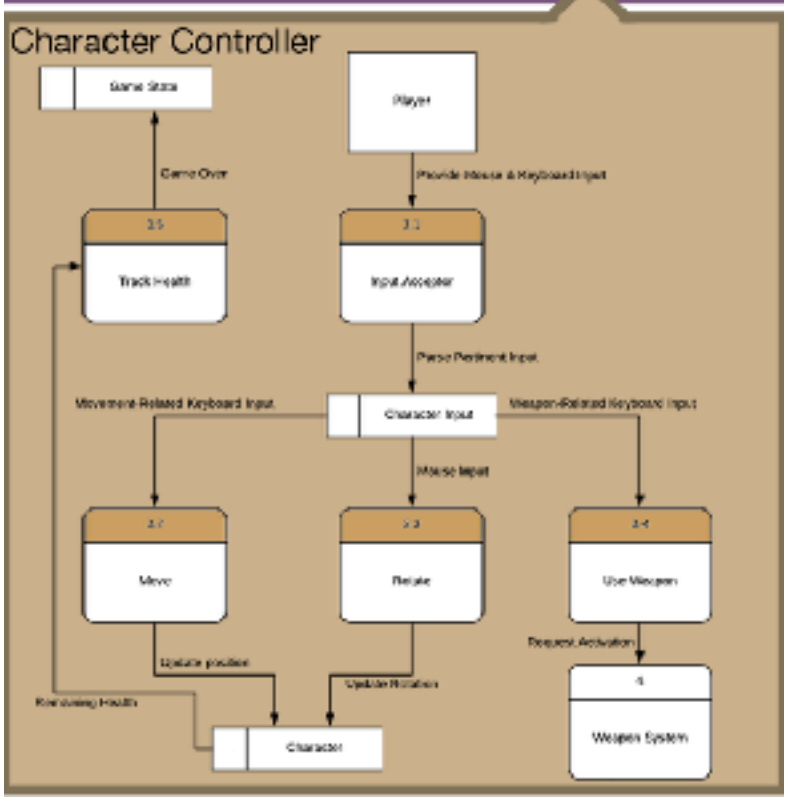
Post conditions: A sound is made.
Priority: 3*
ID: C07
*The priorities are 1 = must have, 2 = essential, 3 = nice to have.

Name: Check state
Summary: The game state is checked.
Actors: Player
Preconditions: The player initiates some action.
Basic sequence:
Step 1: Contact the game manager and record the current game state.
Exceptions:
Step 1: If no game manager found, assume play state.
Post conditions: Game state is known.
Priority: 3*
ID: C08
*The priorities are 1 = must have, 2 = essential, 3 = nice to have.

3. Data Flow diagram(s) from Level 0 to process description for your feature _____/14

Data Flow Diagrams





Process Descriptions

Accept Input:

```
WHILE game is running
    IF game state is play
        Accept user input
        IF input relates to movement, weapons, or rotation
            parse as character input
        ELSE
            ignore input
        ENDIF
    ENDIF
END WHILE
```

Move:

```
WHILE game is running
    create empty vector for movement
    IF character input contains ‘w’
        Add vector (0,1) * speed to move
     ENDIF
    IF character input contains ‘a’
        Add vector (-1,0) * speed to move
    ENDIF
    IF character input contains ‘s’
        Add vector (0,-1) * speed to move
    ENDIF
    IF character input contains ‘d’
        Add vector (1,0) * speed to move
    ENDIF
    add move to character position
    clamp position to screen bounds
    clamp position to not overlap with enemies
END WHILE
```

Rotate:

```
WHILE game is running
    create rotation facing mouse position
    update character rotation
END WHILE
```

Use Weapon:

```
WHILE game is running
    IF game state is play
        IF character input contains ‘leftClick’
            Request weapon system shoot bullet
        ENDIF
```

```
                    IF character input contains ‘rightClick’
                        Request weapon system drop trap
                    ENDIF
                ENDIF
            END WHILE
        Track Health:
            WHILE game is running
                IF character health is <= zero
                    initiate game over
                ENDIF
            END WHILE
```

4. Acceptance Tests _____/9

This feature is fairly simple and should be tested by a real person. It would take more time to come up with an automated test for such a simple game than to just test it by hand.

Tests 1 - 3 should use the following inputs:

- Move up with the W key.
- Move left with the D key
- Move down with the S key
- Move right with the D key
- Move up/right with the W/D keys
- Move up/left with the W/A keys
- Move down/right with the S/D keys
- Move down/left with the S/A keys
- Press W/S and A/D and create no movement.

The first test is just to move the character around in the middle of an empty screen and observe results, being sure not to move near the edge of the screen, obstacles, or enemies. For each key, try both pressing once, holding down, and pressing rapidly The character should always move the same distance on a given key press, unless the key press shouldn’t create movement.

The second test should take place at the edge of the map. The player should attempt to move outside the bounds of the screen. Hold down the keys and press rapidly.

The third test should include enemies. The player should attempt to move into space occupied by the enemies and should not be able to do so. Hold down the keys and press rapidly.

Test four requires attacking enemies, a game over feature implemented in the game manager, and debug output of the player's health. The enemies should attack the character, and when the debug output of the player's health goes below 0, game over should be initiated.

In test five, the player should start in the middle of an empty map and move their mouse around. The character should always rotate towards the mouse.

Test six requires debug output from the weapon system. It should indicate that it has received a request to shoot a bullet every time it receives the left click input. This should happen both when the button is held down and when it is pressed.

Test seven should only be performed if the trap feature is implemented and requires debug output from the weapon system. It should indicate that it has received a request to drop a trap every time it receives the right click input. This should only happen when the button is pressed.

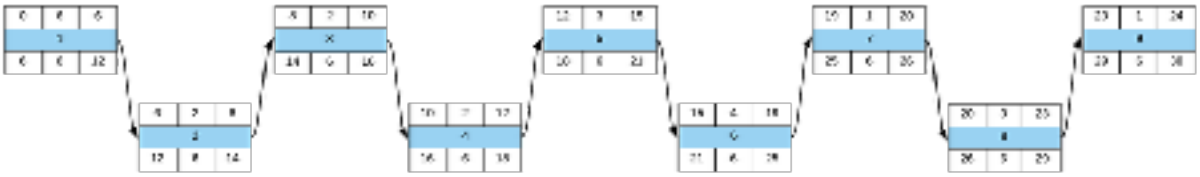
5. Timeline _____/10

Nothing can be done in parallel because it all must be done by me and I cannot do two things simultaneously.

Work items

Task	Duration (Hours)	Predecessor Task(s)
1. Requirements Collection	6	-
2. Create player object	2	1
3. Create camera	2	1,2
4. Collect input	2	1,2,3
5. Add movement	3	1,2,3,4
6. Add rotation	4	1,2,3,4
7. Add weapons	1	1,2,3,4
8. Add movement bounds	3	1,2,3,4,5
9. Add health	1	1,2,3
	24	

Pert diagram



Gantt timeline