

1. Brief introduction __/3

My feature for the Midnight Slice Madness videogame is the movement and mechanics tied to the player character.

When the user inputs specific keys, my job is to make sure that the player character sprite responds correctly. For example, inputting certain keys moves the character around the map. I also deal with the different conditions of the vehicle that the player controls, such as health, gas, headlights, and brake lights.

Additionally, I'm responsible for making sure the character properly collides with other objects as well as reacting or send out signals for what happens with those collisions.

2. Use case diagram with scenario __/14

Use Case Diagrams



Scenarios

Name: Drive Vehicle Forward

Summary: The accelerates the vehicle forward.

Actors: Player.

Preconditions: Player is in vehicle and on the map.

Basic sequence:

Step 1: Accept input of drive forward key.

Step 2: While input key is pressed increase forward velocity.

Step 3: While input key is pressed decrease the gas can.

Step 4: If the key is pressed long enough the speed caps out.

Exceptions:

Step 1.1: The drive left input is pressed: The car veers left.

Step 1.2: The drive right input is pressed: The car veers right.

Step 2: Forward drive key is release: Velocity decreases until it becomes 0

Post conditions: Vehicles velocity is changed.

Priority: 1*

ID: C01

Name: Reverse Vehicle

Summary: The accelerates the vehicle backwards.

Actors: Player.

Preconditions: Player is in vehicle and on the map. Velocity is 0.

Basic sequence:

Step 1: Accept input of reverse key.

Step 2: While input key is pressed decrease forward velocity.

Step 3: While input key is pressed decrease the gas can.

Step 4: If the key is pressed long enough the speed caps out.

Exceptions:

Step 1.1: The drive left input is pressed: The car veers left.

Step 1.2: The drive right input is pressed: The car veers right.

Step 2: Forward drive key is release: Velocity decreases until it becomes 0

Post conditions: Vehicles velocity is changed.

Priority: 1*

ID: C02

Name: Brake Vehicle

Summary: Presses the brakes on the vehicle to bring it to a halt.

Actors: Player.

Preconditions: Player is in vehicle and on the map. Velocity is a non 0 amount.

Basic sequence:

Step 1: Accept input of brake key.

Step 2: While input key is pressed decrease velocity faster than holding no forward or reverse key.

Step 3: While the brake key is pressed light brake lights.

Post conditions: Vehicles velocity is 0.

Priority: 2*

ID: C03

Name: Refill Gas

Summary: The player refills their gas tank using an item.

Actors: Player

Preconditions: Player has the correct item and gas is lower than 100% full.

Basic sequence:

Step 1: Press input the key to use the gas refill item.

Step 2: Increase the gas amount the specified amount.

Step 3: Remove gas refill item.

Exceptions:

Step 2: Increasing the gas would be raise it more 100% full: set the amount to 100%.

Post conditions: Calculated value is displayed.

Priority: 3*

ID: C04

Name: Collide with Wall

Summary: The player collides the vehicle into an obstacle and damages the vehicle.

Actors: Player.

Preconditions: The vehicle has a non 0 velocity

Basic sequence:

Step 1: The vehicle collides with an object.

Step 2: Decrease the health of the vehicle.

Exceptions:

Step 2: Decreasing the health would put the health below 0: Set the health to 0.

Post conditions: Calculated value is displayed.

Priority: 3*

ID: C05

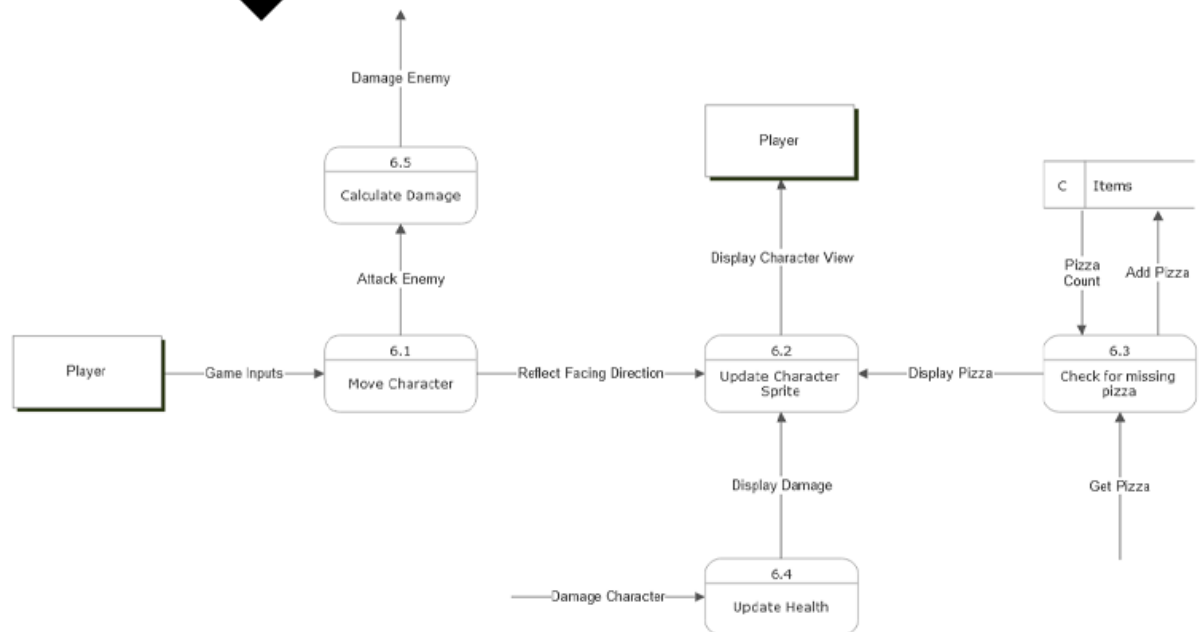
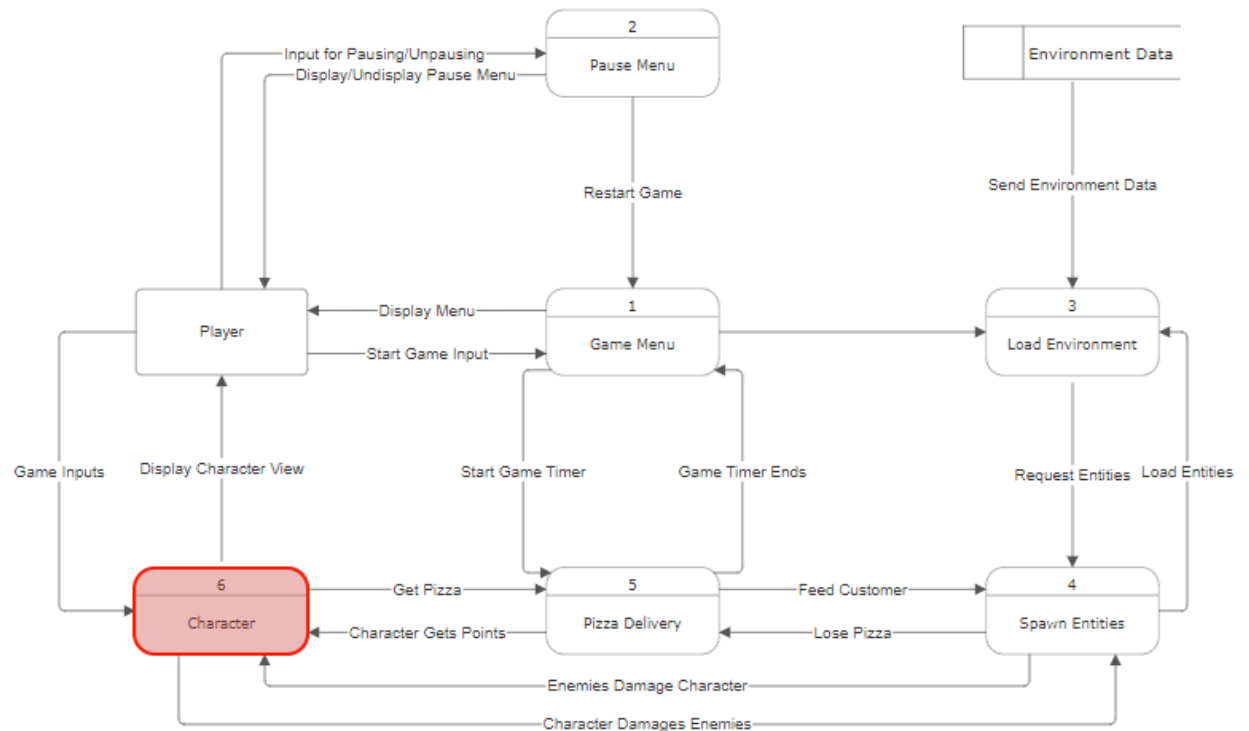
*The priorities are 1 = must have, 2 = essential, 3 = nice to have.

3. Data Flow diagram(s) from Level 0 to process description for your feature ____14

Context Diagram



Data Flow Diagrams



Process Descriptions

6.1 Move Character:
IF input is forward

Increase the speed until it maxes out, decrease gas amount
 IF left or right input veer the correct direction
IF input is backward
 Drive backward, decrease gas amount
 IF left or right input veer the correct direction
IF brake is input
 Turn on brake light and decrease speed
6.2 Update Character Sprite:
 Change sprite to visually reflect character state
6.3 Check for Missing Pizza:
 IF no current pizza in items
 Add pizza AND update sprite to show pizza
 Store that there is now a pizza
6.4 Update Health:
 Decrease player's health by the damage amount.
 IF below certain threshold
 Update sprite to show damage taken
6.5 Calculate Damage:
 Based on type of attack and given factors then

4. Acceptance Tests _____9

Overview: The character movement feature involves processes from movement, stats, lighting.
Making sure these all work correctly

Health Points (HP) Tracking

Inputs:

 Set character's initial HP to 100.
 Character receives damage of 20 points.

Expected Outputs:

 Character's HP should be reduced to 80.
 The HP should not drop below 0.

Boundary Cases:

 Test with initial HP set to 0.
 Test with receiving damage exactly equal to current HP.

Gas Tracking

Inputs:

 Set character's initial gas level to 50%.
 Character consumes 10% of their gas.

Expected Outputs:

 Character's gas level should reduce to 40.
 If the gas level drops to 0, character should not be able to move vehicle.

Boundary Cases:

 Test with initial gas level set to 0.
 Test with consuming gas exactly equal to current gas level.

HP Recovery

Inputs:

- Set character's initial HP to 50.
- Character repairs vehicle that restores 30 HP.

Expected Outputs:

- Character's HP should increase to 80.
- Ensure HP does not exceed maximum HP (e.g., capped at 100).

Boundary Cases:

- Test with character's HP at maximum (100).
- Test with character's HP at 0.

Gas Replenishment

Inputs:

- Set character's initial gas level to 20%.
- Character refills gas tank with 40%.

Expected Outputs:

- Character's gas level should increase to 60.
- Ensure gas level does not exceed maximum gas capacity.

Boundary Cases:

- Test with character's gas at maximum capacity.
- Test with character's gas at 0.

Object Collision

Inputs:

- Load character and an object.
- Move character forward with the object in the path

Expected Outputs:

- Character should stop when it meets the object

Boundary Cases:

- Start with both objects already colliding

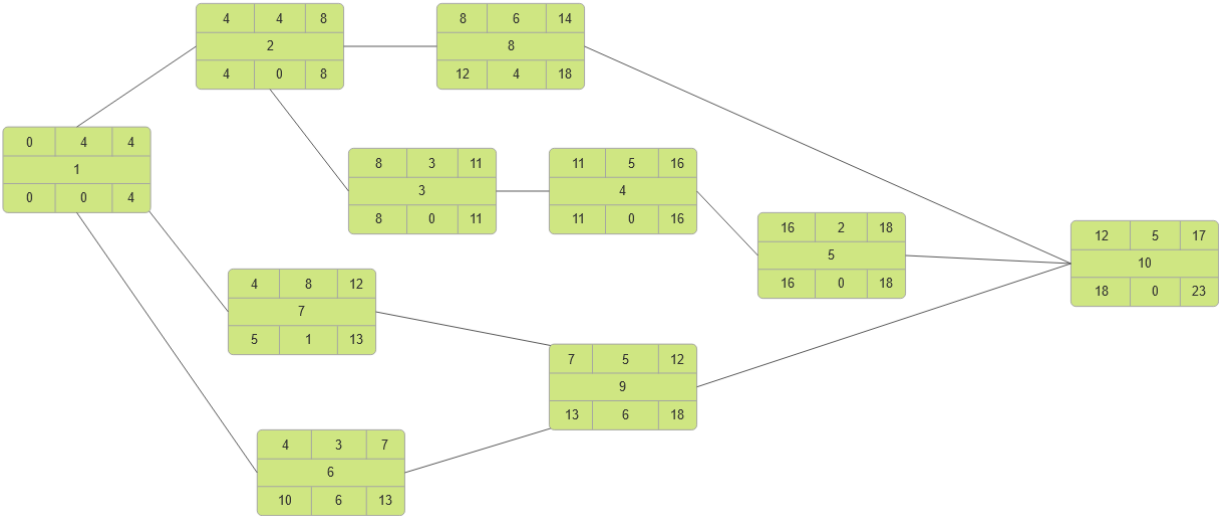
5. Timeline ____/10

Work items

Task	Duration (Hours)	Predecessor Task(s)
1. Acceleration/Deceleration	4	-
2. Turning/Steering	4	1
3. Braking	3	2
4. Brake Lights	5	3
5. Headlight Lights	2	4
6. Gas	3	1

7. Health/Collision Damage	8	1
8. Friction/Traction	6	2
9. Animations/Visual Status	5	6, 7
10. Testing/Installation	5	5, 8, 9
TOTAL	45	-

Pert diagram



Gantt timeline

