

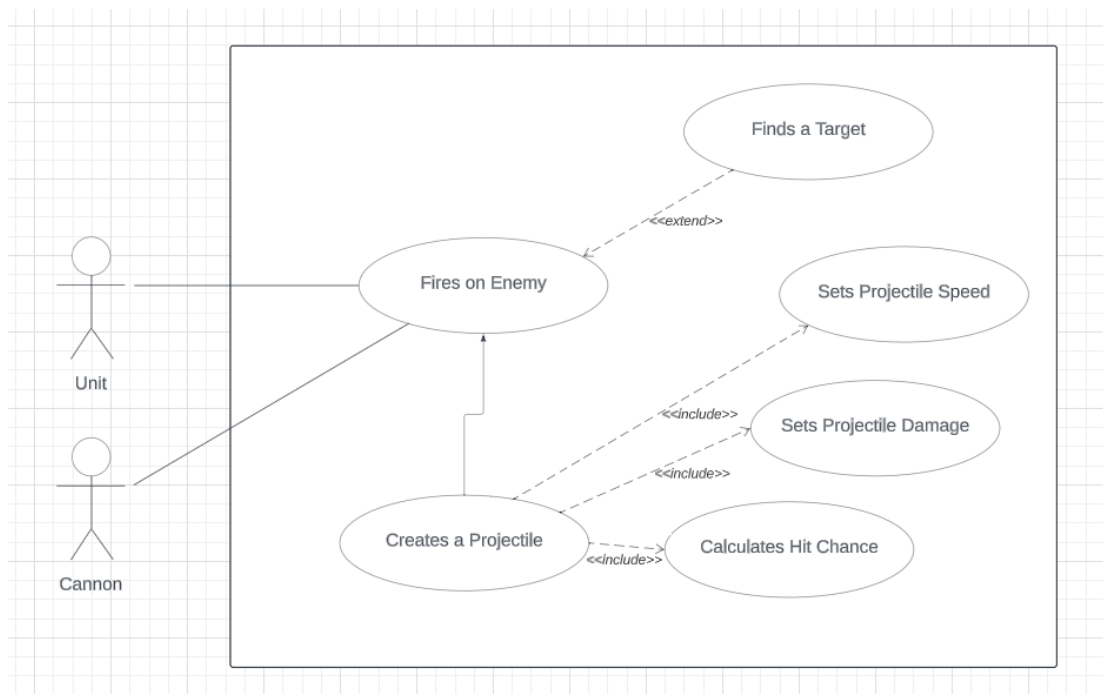
1. Brief introduction _/3

My feature for the Corsair Clash game is the battle physics system and how the player ship interacts with an enemy ship.

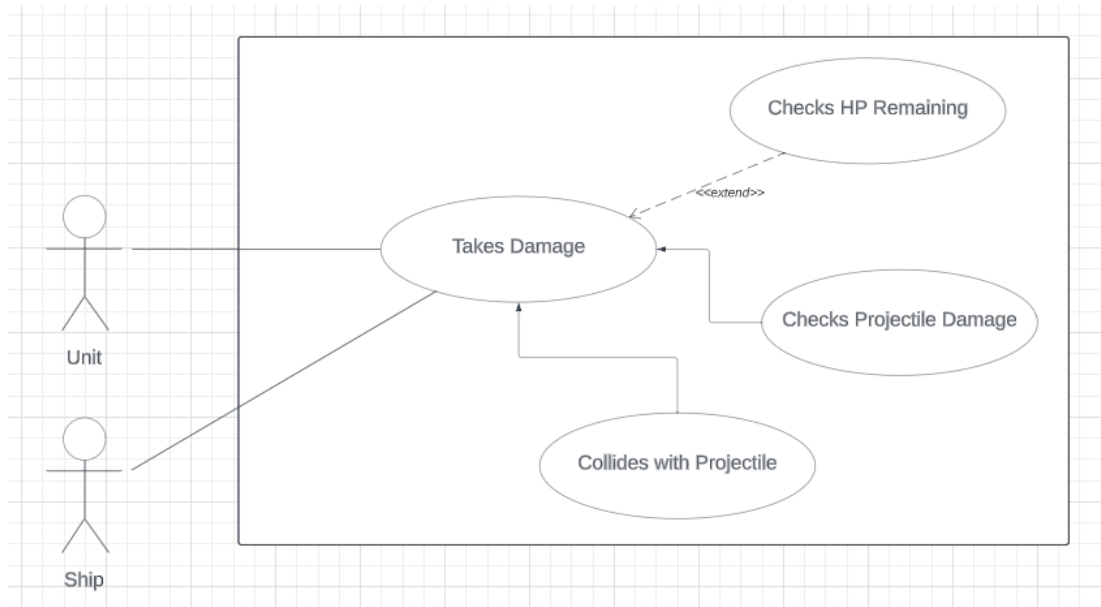
When a player initiates combat with an enemy, a battle should start. My job is to create the ways the player and enemy ship damage each other during this battle phase and how the weapons and units within the game function. I am also responsible for how the ships themselves and the units on them aim their weapons as well as receive damage.

2. Use case diagram with scenario _/14

Use Case Diagrams



Use Case Diagram 1: Fires on Enemy



Use Case Diagram 2: Takes Damage

Scenarios

Scenario 1 (First Use Case Diagram):

Name: Fires on Enemy

Summary: The units and cannons fire on the enemy units and ship.

Actors: Unit, Cannon.

Preconditions: A battle has been initiated.

Basic sequence:

Step 1: Units find a target on the opposing ship.

Step 2: Units continue find unique targets until each unit on screen has locked onto an opposing unit.

Step 3: Each actor creates a projectile.

Step 4: Each actor calculates their hit chance depending on the actor they are targeting.

Step 5: The projectiles have their speed level set depending on the type of actor firing them.

Step 6: The projectiles have their damage level set depending on the type of actor firing them.

Exceptions:

Step 1: A battle is started when the opposing ship has no units to target: The player wins automatically.

Step 2: Units cannot find a unique target on the opposing ship: Units start cycling through previously targeted units from the beginning.

Post conditions: Units and cannons fire on the opposing ship.

Priority: 1*

ID: F01

Scenario 2 (Second Use Case Diagram):

Name: Takes Damage

Summary: When an actor collides with a projectile, the actor takes damage in the form of negative health points.

Actors: Unit, Ship.

Preconditions: A battle has been initiated.

Basic sequence:

Step 1: Projectile collides with actor.

Step 2: Checks the remaining health points of the player against the damage done by the projectile.

Step 3: Actor takes damage to their health points based on how much damage the projectile is set to have.

Exceptions:

Step 2: Actor has fewer hit points than damage points of the projectile that collides with them: Actor is destroyed without calculating remaining health points.

Post conditions: Actor is left with less health than before they collided with a projectile.

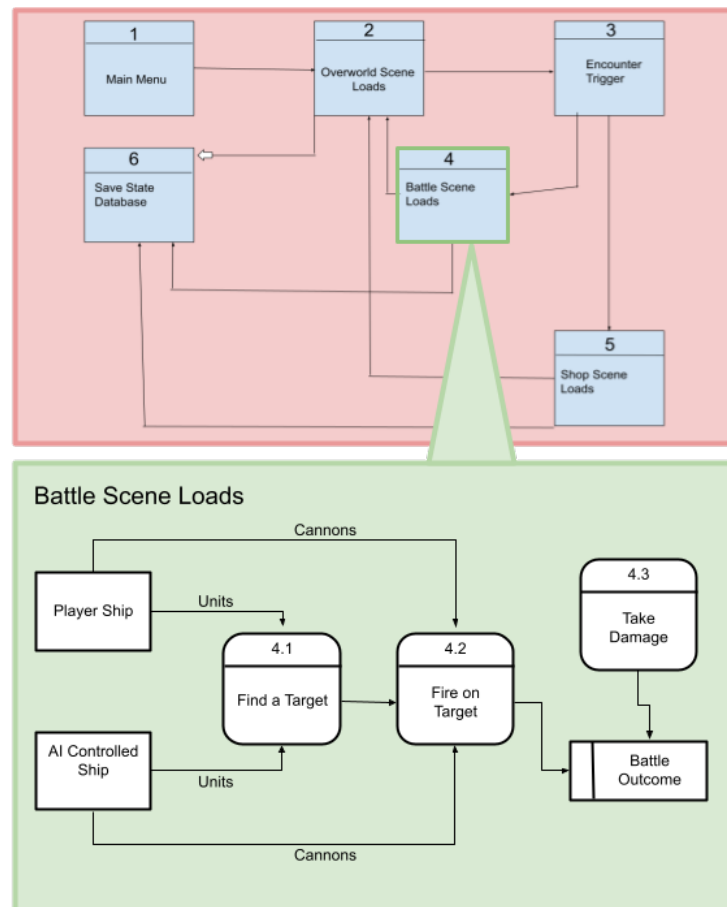
Priority: 1*

ID: T01

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

In the data flow diagram below, the Battle Physics feature data flow is showcased connected to the Battle Scene Loads process. The subprocesses in this feature are Find a Target, Fire on Target, and Take Damage.

Data Flow Diagrams



Process Descriptions

Process 4.1 - Find a target:

WHILE number of targeted units is less than number of units in opposing ship's possession

IF the number of units under current ship control is less than or equal to the number of units under opposing ship control
THEN current ship units are given a unique target numbered unit one by one until each unit in the current ship's possession has a target.

END IF

ELSE IF the number of units under current ship control is greater than the number of units under opposing ship control

THEN IF the unit will be targeting a null opposing unit
THEN cycle the number of opposing unit to target back to 1

END IF

ELSE

The current ship units are given a target numbered unit one by one until each unit in the current ship's possession has a target.

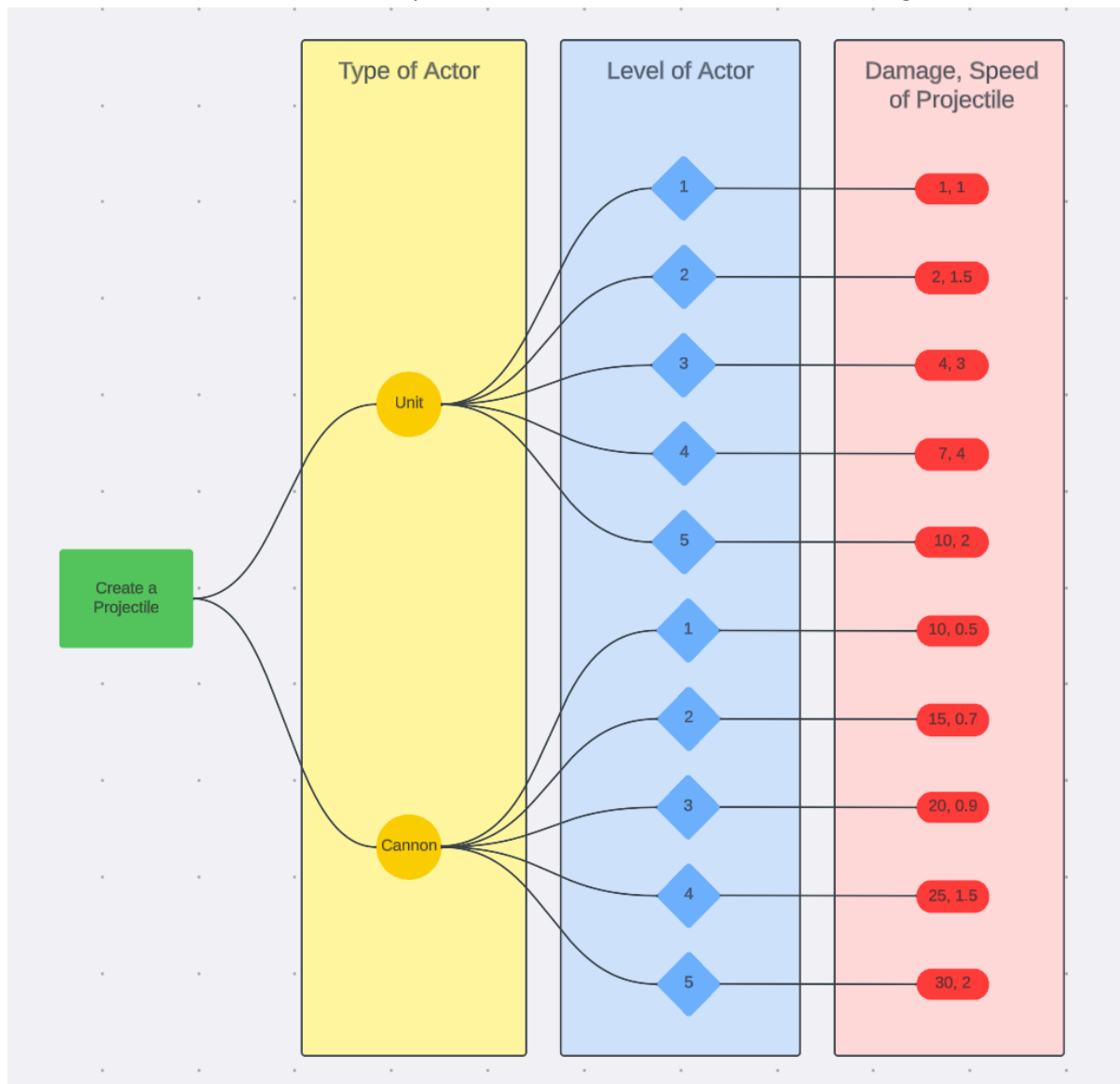
END ELSE

END ELSE IF

END WHILE

Process 4.2 - Fire on Target:

Process Description Decision Tree – Process x.x Fire on Target



Process 4.3 - Take Damage:

```
IF opposing unit's projectile collides with a unit on the current ship
    THEN IF the hit unit's current health is greater than the damage
        done by the projectile
            THEN subtract the amount of damage dealt by the
                projectile from the hit unit's health points
        END IF
    ELSE IF the hit unit's current health is less than or equal to the
        damage done by the projectile
        THEN destroy the hit unit
    END ELSE IF
END IF

IF opposing ship's cannon's projectile collides with the current ship
    THEN IF the hit ship's current health is greater than the damage
        done by the projectile
            THEN subtract the amount of damage dealt by the
                projectile from the hit ship's health points
        END IF
    ELSE IF the hit ship's current health is less than or equal to the
        damage done by the projectile
        THEN destroy the ship and call the game over screen
    END ELSE IF
END IF
```

4. Acceptance Tests _____9

Considering that firing a weapon is done by one party and receiving damage is done only when an actor is collided with, this feature will have two separate acceptance tests, for firing a weapon and for an actor receiving damage.

Firing a Weapon:

This process will run 1000 times where each level of cannon or unit will be tested for creating a projectile and setting the damage and speed levels for these projectiles, sending the output to a separate file.

The output file for this process should display the level of actor firing the projectile, the level of the projectile, the damage number of the projectile and the speed level of the projectile. It should also display a true or false value if the actor needs to target an enemy (if it is a unit). Since cannons and units have different values for their projectiles, no type of actor should appear more than 500 times, and no level of a given actor should appear more than 100 times.

Example test check for the Firing a Weapon process:

Actor Type	Needs Targeting?	Actor Level	Projectile Level	Damage of Projectile	Speed of Projectile
Unit	True	5	5	10	2
Ship	False	3	3	20	0.9
Unit	True	4	4	7	4
Unit	True	1	1	1	1
Ship	False	4	4	25	1.5

Taking Damage:

This process will run 1000 times where a ship or unit level will be tested against each unit or cannon projectile level, sending the output to a separate file. 500 tests will be on a ship and 500 tests will be on a unit with given levels between 1 (the lowest level with the least health) and 5 (the highest level with the most health). In these tests, the level of ship or unit being tested will be tested against a given level of projectile that can collide with it (between 1 and 5).

The output file should display the type of actor tested against, the level of actor tested against, the damage level of projectile tested, the damage of the projectile, the maximum health of the actor, and the remaining health after taking damage. No type of actor should appear more than 500 times, and no level of actor per type should appear more than 100 times. Within those 100 tests per actor, each level of projectile should appear 20 times.

Example test check for the Taking Damage process:

Actor Type	Actor Level	Projectile Level	Projectile Damage	Maximum Actor Health	Actor Health Remaining
Unit	4	2	2	40	38
Ship	3	1	10	700	690
Unit	2	4	7	20	13
Unit	1	5	10	10	0
Ship	5	2	15	900	885

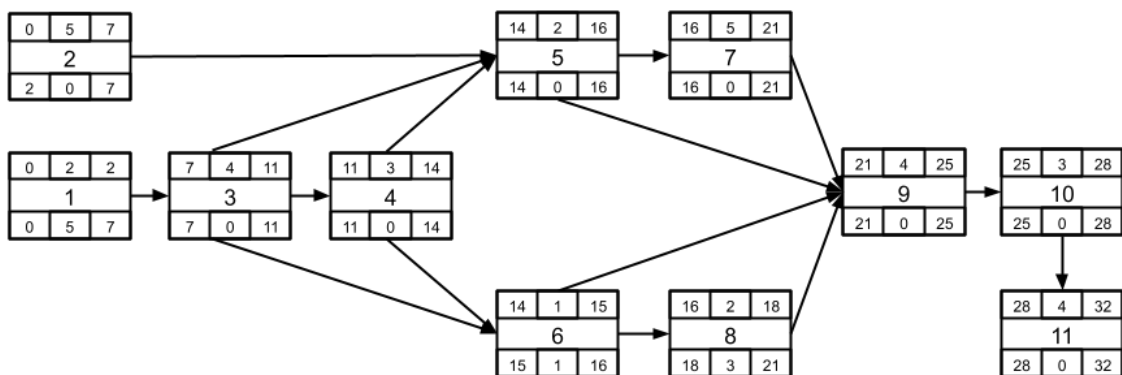
5. Timeline ____/10

Work items

Task	Duration (PWks)	Predecessor Task(s)
1. Projectile Prefab Creation	2	-

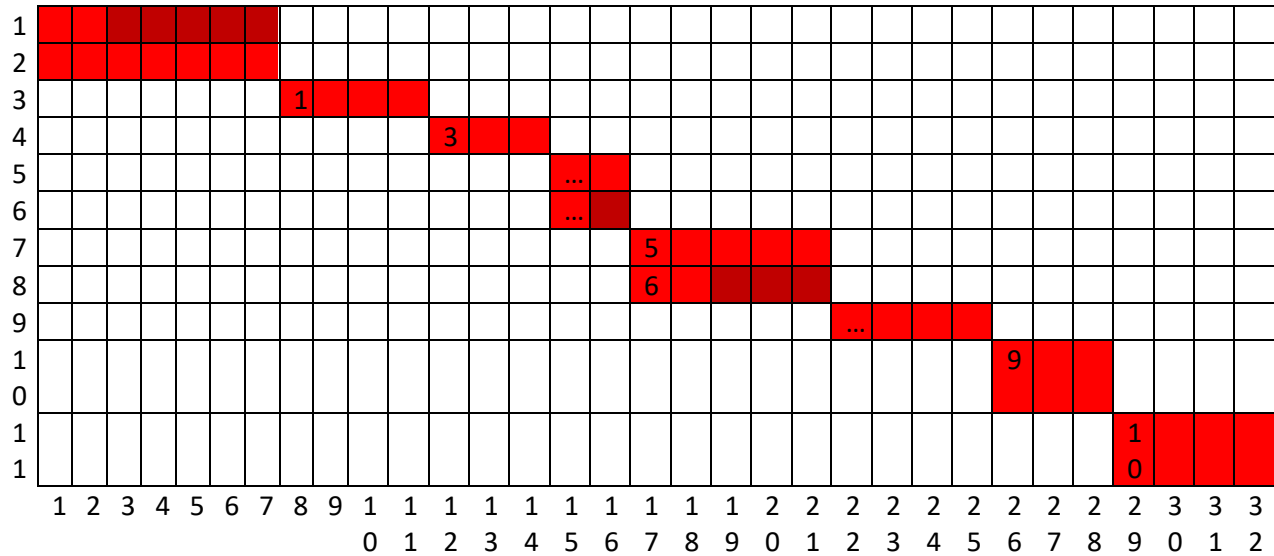
2. Unit Targeting System Programming	5	-
3. Programming Actors Creating Projectiles	4	1
4. Projectile Movement Creation/Tweaking	3	3
5. Programming Units Firing at Targets	2	2,3,4
6. Programming Cannons Firing on Opposing Ship	1	3,4
7. Programming Units Taking Damage	5	5
8. Programming Ships Taking Damage	2	6
9. User Documentation	4	5, 6, 7, 8
10. Testing	3	9
11. Installation into Game	4	10

Pert diagram



Gantt timeline:

Key:



Dependencies (...):

- 5 depends on 2, 3 and 4.
6 depends on 3 and 4.
9 depends on 5, 6, 7 and 8.