Rodney McCoy
Team: ScoSoft
Game: Scoto
Feature: Weapons

1. Brief introduction

My feature for our game, Scoto, is implementing the weapons the player will use to attack the enemies in the game. I will be implementing their physics, their animations on the screen as they are fired, and the methods to identify when an enemy creature has been hit.

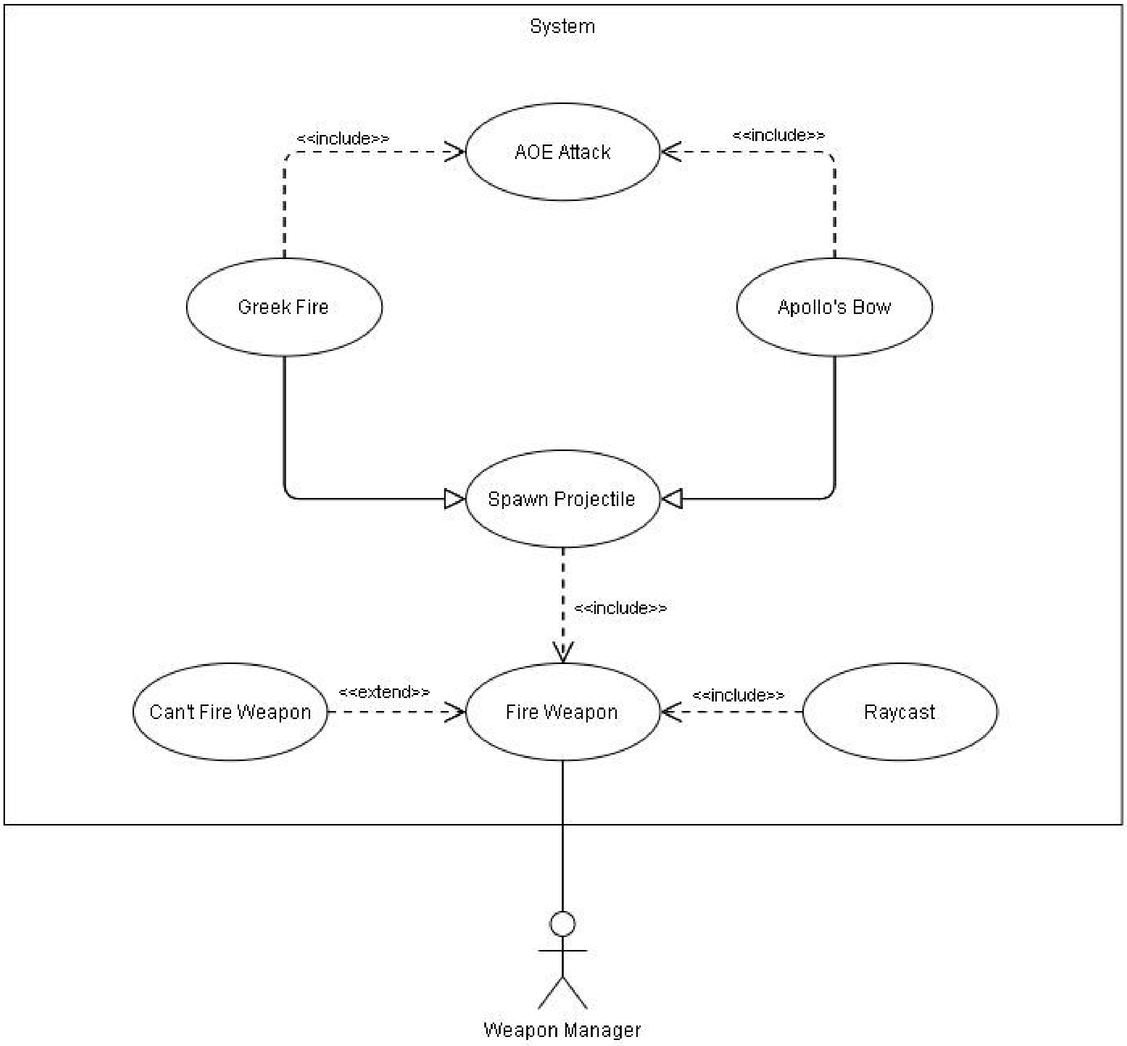
The weapons in our game are placed across the map in loot boxes, which we call “Pandora’s Boxes”. The goal of the player is to traverse through as much of the maze as possible, with the items they find to help them get as far as possible. A script I will create is called the WeaponManager, which will store the current weapons discovered, and the amount of ammo currently held (some weapons have infinite attacks, some weapons have a finite amount of ammo). The WeaponManager will also store the current weapon that is selected. Because there is a small amount of weapons available, weapon selection will most likely show up on the hud and be traversed with scroll wheel / 1-9 hotkeys / or another keybind. The WeaponManager will also store if a weapon can be fired right now. It will know if the game is paused, if a weapon is currently being fired, or if there is some other condition inhibiting weapon firing. When a left click is recognized by the input manager, it will check to see if a weapon can be fired. If the player is ready to attack, It will then check what the current weapon selected is, and then begin firing that weapon.

When the WeaponManager fires a weapon, I will handle the animation of the weapon attacking. Once the weapon firing animation starts, the weapons will either use the unity Physics.raycast() method to detect if the attack hit an enemy or a new projectile will be instantiated that can be affected by the world and can collide with and damage enemies.

Another type of attack that can damage an enemy will be an AOE attack. This attack will be reserved for opening Pandora’s boxes or spawned from projectiles. The AOE attack will be placed around Pandora’s boxes so the player can safely open them without being attacked by an enemy, or potentially be spawned by a projectile after a certain amount of time, while the projectile moves, or after it collides with the ground. An enemy that collides with this AOE attack will be damaged over a period of time and will be debuffed (stunned, slowed, blinded).

2. Use case diagram with scenario

Use Case Diagrams



Scenarios

Scenario 1 (first Use Case Diagram):

Name: Firing Weapons

Summary: Weapon Manager takes user input and determines if a weapon should be fired

Actors: Weapon Manager

Preconditions: Left Click determined from input manager

Basic sequence:

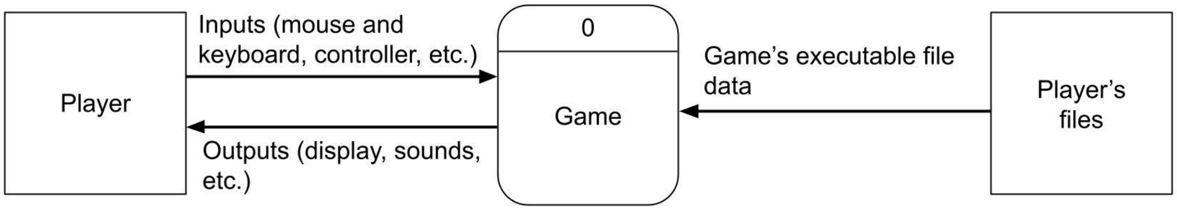
Step 1: Make sure EnableAttack is true

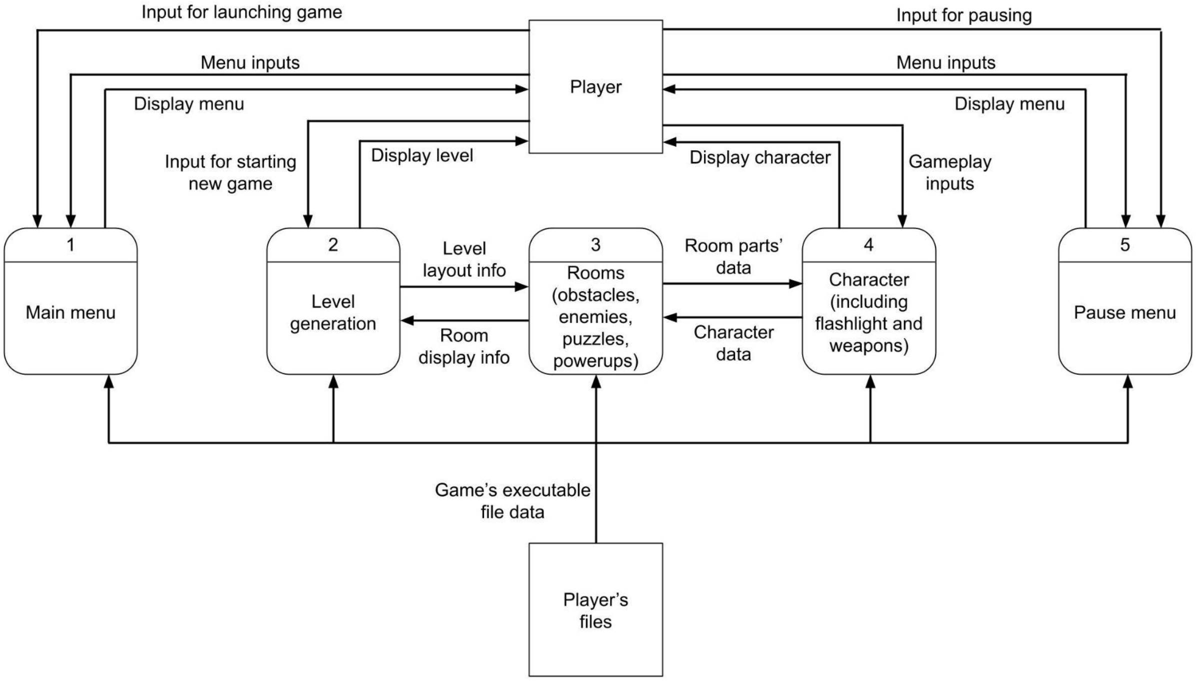
Step 2: If Current Weapon is Poseiden’s Trident, Animate Attack, and Raycast from current player position.

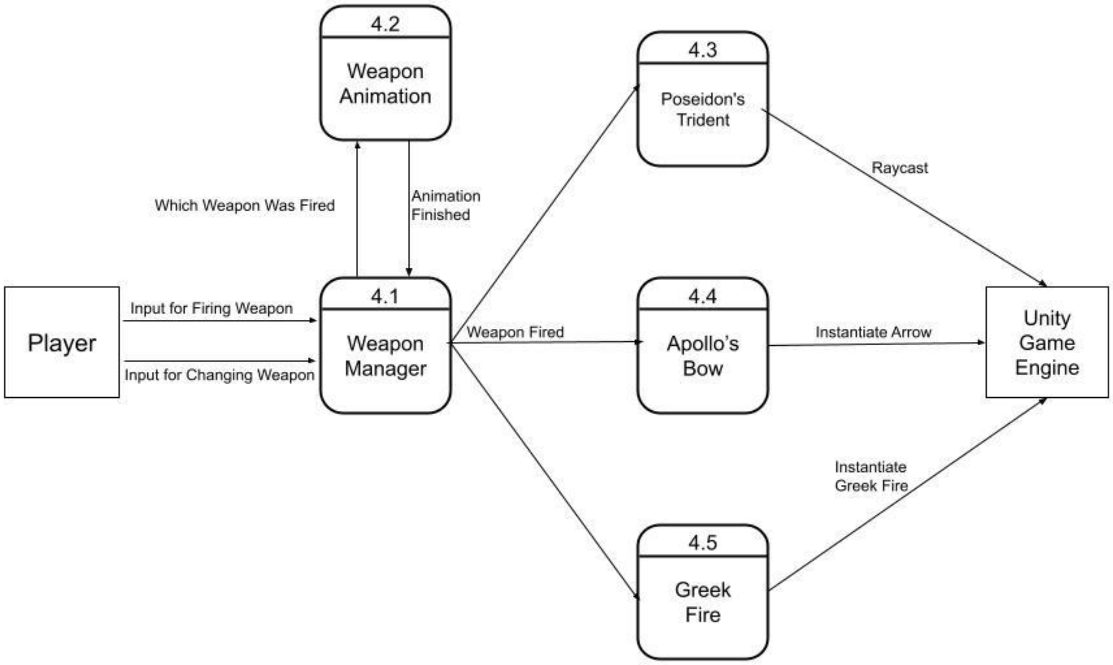
Step 3: If Current Weapon is Greek Fire, Animate Attack, Instantiate a Greek Fire projectile on current player position.
Step 3a: Once Timer increments to MAX_TIME, delete greek fire projectile and instantiate AOE attack.
Step 4: If Current Weapon is Apollo’s Bow, Animate Attack, Instantiate an arrow on current player position
Step 4a: Once the arrow strikes the ground, delete the arrow projectile and instantiate AOE attack.
Exceptions:
Step 1: If out of greek fire, don’t animate the attack or instantiate a greek fire projectile.
Step 2: If EnableAttack is false, dondon’tt animate any attack or fire weapon.
Post conditions: If raycast determines a collision with an enemy, or an enemy contacts an AOE effect, damage the enemy by “Damage” hp.
Priority: 2
ID: RM01

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

Data Flow Diagrams


Player
Inputs (mouse and keyboard, controller, etc.)
Outputs (display, sounds, etc.)
0
Game
Game's executable file data
Player's files





Process Descriptions

4.1 Weapon Manager:

```
Void Update()
{
        IF scroll wheel moved:
                IF scroll wheel moved up:
                        Decrement Current Weapon
                        Update Current Weapon
                IF scroll wheel moved down:
                        Increment Current Weapon
                        Update Current Weapon
        IF left click detected AND Enable Attack True:
                Set Enable Attack to false
                Animate Attack
                IF Poseiden's Trident is current weapon:
                        FireTrident()
                IF Apollo's Bow is current weapon:
                        FireBow()
                IF Greek Fire is current weapon AND greek fire ammo > 0:
                        FireGreek()
}
```

4.2 Animator:

```
IF Animation finished:
        Set Enable Attack to true:
```

4.3 Poseidon’s Trident:

```
FireTrident()
{
        audioSource.play()
        Physics.raycast()
}
```

4.4 Apollo’s Bow:

```
FireBow()
{
        audioSource.play()
        Instantiate Arrow
}
```

4.5 Greek Fire:

```
FireBow()
{
        audioSource.play()
        Instantiate Greek Fire
}
```

4. Acceptance Tests

The only user input this feature is dependent on is a Left Click. It is also dependent on values tracked by the Weapon Manager, specifically an integer representing which weapon is currently selected, and a boolean which is flagged false if a weapon or action is currently being performed. These conditions that weapon firing is dependent on are all discrete values that are extremely well defined. So any acceptance test created for this would be redundant, wasteful, and have no purpose.

Running acceptance tests over the physics of the projectiles instantiated could be beneficial, but there's no determined success or fail case. It's only if the physics is implemented realistically or not which matters, and that is subjectective. There is no right or wrong.

Pandora’s Box Protection:

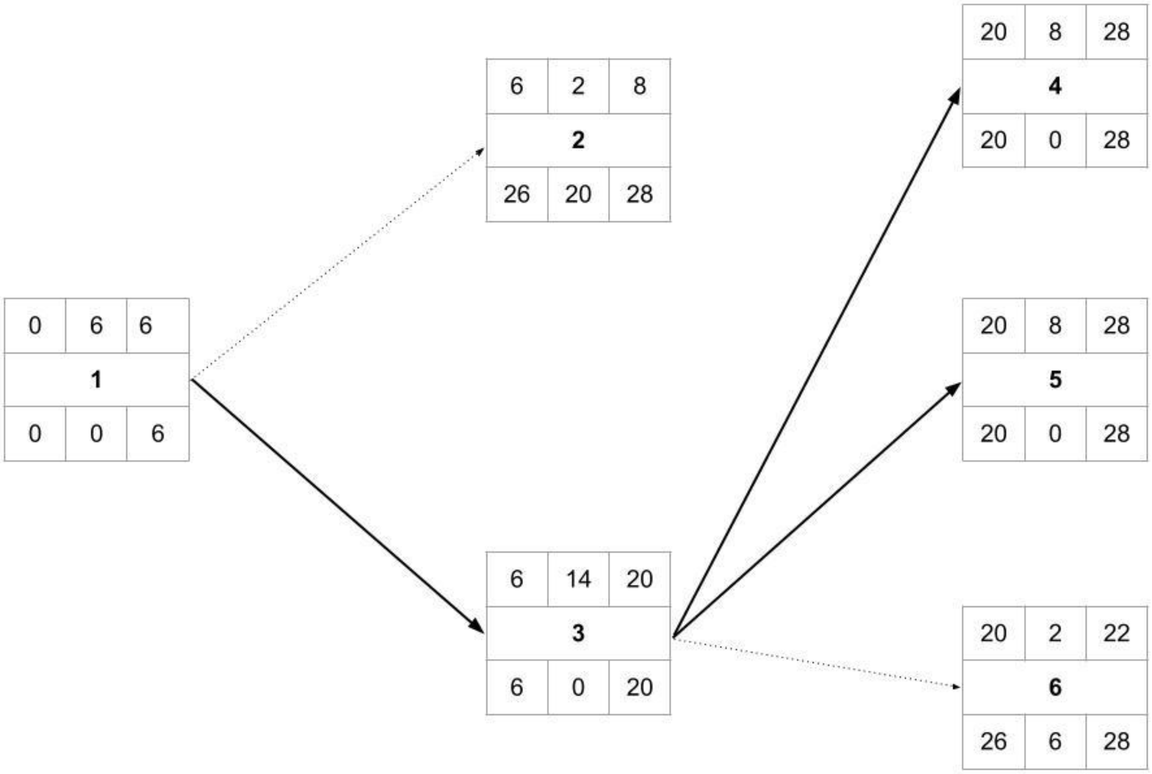
The user should be able to access a pandora’s box safely if they can get it open even if there are enemies in the room they are currently in. An enemy damaging AOE effect should be instantiated when a player opens a box. And no enemy should be able to damage them.

10 enemies will be instantiated (the actual amount depends on how we balance the game, which will determine realistically the upper limit of how many enemies will attack you at once). If the player gets hit once by a melee attack, the acceptance test fails.

5. Timeline

Task	Duration (PHrs)	Predecessor Tasks
(1) User Input and Weapon Manager	6	-
(2) AOE Light Attack	2	1
(3) Animate Weapons	14	1
(4) Arrow Physics	8	3
(5) Greek Fire Physics	8	3
(6) Poseiden’s Trident Physics	2	3

Pert diagram



Gantt timeline

Required Time

Slack time

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	🟩	🟩	🟩	🟩	🟩	🟩																								
2							🟩	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥		
3							1	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩										
4																					3	🟩	🟩	🟩	🟩	🟩	🟩	🟩		
5																					3	🟩	🟩	🟩	🟩	🟩	🟩	🟩		
6																					3	🟩	🟥	🟥	🟥	🟥	🟥	🟥		

Person Hours