

# Feature Specifications – Enemies, Aiming & Shooting, Player Movement

## 1. Enemies

### 1.1 Functionality in stage 2

Enemies are placed in the level, they are **not** spawned during runtime

Enemies also attack when they are off-screen (i.e. when the player is not looking at them during combat)

Standard enemy type

- Patrols within a three-dimensional area assigned to them

  - After a delay (random duration between two assigned integers), it receives a new objective point within the area and moves towards it

  - The delay for the next objective point only starts on reaching the current objective

- Once they detect the player, they start attacking

  - The player is detected if they enter the enemy's three-dimensional cone of vision **and** the line of sight between enemy and player is **not** interrupted by collision

  - The player is detected if they hit the enemy with any arrow

- Returns to the idle state if the player does **not** fulfill any of the two detection conditions for a while

- Shoots X amount of projectiles after a short wind-up to attack

  - Small delay between each of the shots

  - Wind-up -> projectile 1 -> delay -> projectile 2 -> delay -> ... -> projectile X -> longer delay -> repeat

  - Each shot has a random offset from the direction drone-to-player (on the yaw axis), giving it a chance to miss

  - Amount of shots per attack loop needs to be tested

- Keeps patrolling movement pattern even in combat, does **not** follow the player

### 1.2 Ideas for future stages

- Enemies respawn on player death

- Enemies regenerate health when returning to their patrol state

- Enemies have health bars attached that are shown once they take damage

- Damage numbers appear on hitting an enemy with an arrow

- Sound effects let players know when enemies are winding up their shots off-screen

- Shotgun enemy type

  - Rapidly moves towards the player

  - Fires one powerful shot when it reaches a close distance to the player

  - Gets knocked back from their own shot

Player does **not** get knocked back (because it would be disorienting in first-person)

Follows the player within the boundaries of a large area

Boss will be the third and last enemy type for this game

## 2. Aiming & Shooting

### 2.1 Functionality in stage 2

Holding LMB draws the bow, releasing LMB fires the arrow

Draw time affects the speed of the arrow and thus the impact of gravity on it

Draw time also affects the damage dealt by the arrow

Increases linearly with the draw time

Distance of the enemy or speed of the arrow on impact is **not** relevant for damage

Damage example

2 secs draw time, enemy 2 meters away = 25 damage

2 secs draw time, enemy 4 meters away = 25 damage

4 secs draw time, enemy 2 meters away = 50 damage

Can only draw bow to certain degree (cap)

After reaching this cap, the speed of the arrow on release and the damage dealt **no** longer increase

There are **no** spray patterns or random offsets for shot arrows

There is **no** limit on basic arrows, they are an infinite resource

**No** HUD element for ammunition

### 2.2 Ideas for future stages

Arrows have a trail effect attached (so you can tell where they went)

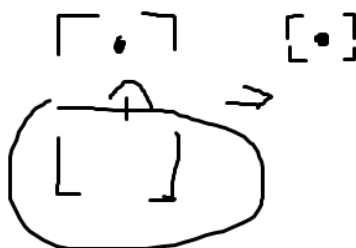
Arrows stick to solid surfaces they hit (enemies, walls, ...)

Purely cosmetic, **no** actual collision

Amount of visible arrows at a time is capped (for performance reasons)

Crosshair animation resembles drawing the bow and its impact on the 'arrow drop'

Tall rectangle with the lower part closing in as you draw the bow



### 3. Player Movement

#### 3.1 Functionality in stage 2

WASD is used for basic movement

Arrow keys are **not** used

SPACE launches a jump

You can dash mid-air

You can aim and shoot mid-air

SHIFT performs a dash

Boost in velocity in the direction the player is moving in

**No** effect if player is stationary

You can aim and shoot while dashing

#### 3.2 Ideas for future stages

Bow lights up during dash (either in approval or disapproval depending on narrative)

**No** additional movement features

Will be outsourced into special quivers / arrow types