

# Feature Specification – User Interface

## 1. Crosshair

### 1.1 Functionality

Crosshair has two different modes, exploration and combat

Always starts in exploration mode

Goes into combat mode when pressing left mouse button (to shoot)

Returns to exploration mode when player does **not** shoot for some time

Timer starts once you fire the first arrow and is reset every time you shoot another one (always starts/resets on releasing the arrow)

While you are drawing the bow, the timer is paused (combat mode **cannot** be left while drawing the bow)

If timer reaches a certain value (which should be an exposed variable), the crosshair transitions back to exploration mode and the timer is stopped

Exploration mode has **no** special functionality

In combat mode, parts of the crosshair (that are **not** visible in exploration mode) are animated to resemble the progress of drawing the bow

A special effect signifies when you have reached the maximum bow tension

### 1.2 Audiovisual requirements

Positioned in the very center of the screen

Needs to contrast well with environment and enemies

Exploration mode crosshair must be subtle so players can focus on the environment and DevNotes

Both crosshair modes should not take up too much of the screen space

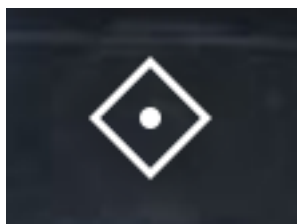
Combat mode needs additional visual elements that can be animated to resemble drawing the bow

E.g. vertical progress bars filling up, lines moving towards the center

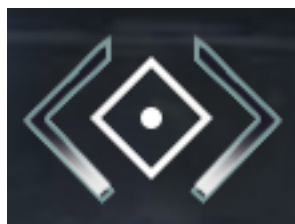
Some sort of effect will signify when you have reached maximum bow tension

E.g. progress bars flash or change colors

### 1.3 Example



Exploration mode



Combat mode



Combat mode – bow fully drawn

### 1.4 Ideas for future stages

Crosshair could change its appearance on killing an enemy

Visual effect could give feedback on releasing the arrow

## 2. Enemy health bar

### 2.1 Functionality

Represents the (percentage of) health an enemy has left

Updates every time the enemy loses or gains health points

Health bar is attached to enemy in space

Always rotates to face player camera

Appears when enemy receives damage

Vanishes when enemy resets (and thus regenerates health) or when enemy dies

### 2.2 Audiovisual requirements

Positioned in world space, above the model of the corresponding enemy

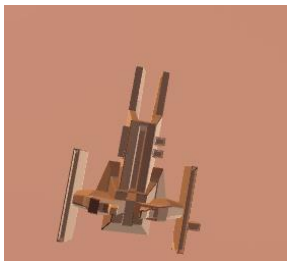
Bar background needs to contrast well with the environment

Fill color needs to contrast well with the bar background, the environment, and the enemy itself

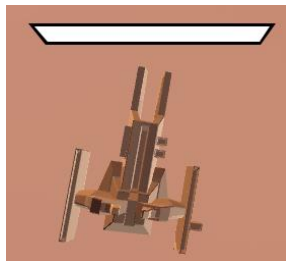
Dimensions must roughly match the dimensions of enemies

4-5 health bars must be visible at the same time without cluttering the screen too much

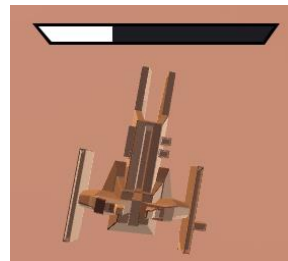
### 2.3 Example



Enemy not yet attacked



Enemy with full health



Damaged enemy

### 2.4 Ideas for future stages

Bar could give feedback when enemy was hit by an arrow

Amount of dealt damage could be highlighted on the bar itself

Could add visual elements to indicate the absolute amount of health

## 3. Player health

### 3.1 Functionality

Represent the health points of the player

Updates every time the player loses or gains health points

Resets when player respawns

Visual feedback effect on receiving damage

Player enters critical state when health points drop below 30%

Change of visuals indicates this

### 3.2 Audiovisual requirements

Positioned in the lower left part of the screen

Can be a bar, text, or even both

Should not take up too much screen space vertically

Some sort of visual effect should happen when player takes damage

Should indicate when player is in critical state health-wise

E.g. by changing color

Needs to contrast better from the rest of the screen than the non-critical state

Needs to be independent from the visual effect for receiving damage

### 3.3 Ideas for future stages

Bar could be animated instead of jumping to the next value immediately

Another type of visual effect could indicate regeneration