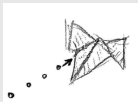


Legend:

- *variable* (in italics), and its [type]
- stretch goal beyond a functional state



Health



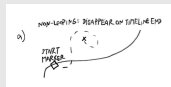
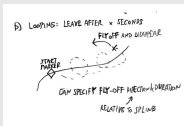
- *starting health* can be set
- *current health* decreases on hit
- destroyed when health reaches 0 (increasing the player's score)

Leaving

If the enemy has not been killed, there are two options [bool] for when and how an enemy may *leave the scene*:

- a) non-looping – once the timeline finishes playing once, simply disappear/self-destruct
- b) looping – loop the timeline for *lifetime duration*
 - afterwards fly off in the *fly-off direction* (relative to the spline) for *fly-off speed* and *fly-off duration*, then disappear

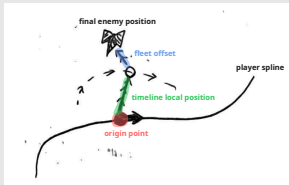
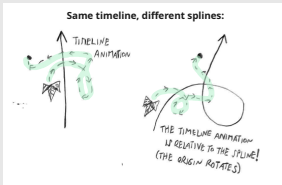
Leaving should not increase player score!



Enemy feature specification

"As a designer, I want to use a tool for creating enemy behaviours so that we can design engaging encounters."

Movement



Enemy position

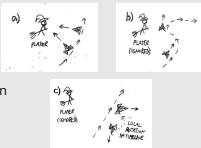
The enemy's position is based on several additive components:

1. the enemy's **origin point**
 - the origin moves on the player's spline
 - it has a rotation – other components are relative to the forward direction of the spline!
 - there are three options for *player dependency* [enum]:
 - a) independent – moves on the spline at a constant *movement speed*
 - b) dependent – maintains a constant *distance* from the player's origin point
 - c) stationary – stays where the enemy was initially placed; does not move or rotate
2. + the enemy behaviour **timeline**
 - each enemy selects a *timeline* to use which defines its behaviour (movement, attacks)
 - a *timeline delay* [float in seconds] defines how long the enemy waits after spawning before starting the timeline animation (this feature is intended for varying enemies in a fleet)
3. + the **fleet offset**
 - the enemy has one additional constant offset [vector] called the *fleet offset*
 - this is used for creating fleets (copies of enemies only need to differ in this one variable)

Enemy rotation

The direction the enemy is *facing* has three options [enum]:

- a) always facing the player
- b) matching the movement from the timeline's local position
- c) unaffected by code: controlled by the timeline's local rotation



Shooting

The times enemies shoot are defined by events in the timeline.

Bullets

Each enemy has a *bullet* they use when shooting.

- These may differ in speed, damage, homing, etc. (outside the scope of this feature spec – handled by the shooting feature team).

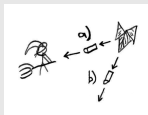
Furthermore, enemies can shoot multiple projectiles per shot event:

- *bullets per shot* [int] specifies the count
- *delay between bullets* [float in seconds] influences the duration
- as an example, with 3 bullets and 1s delay, each event would yield:
 - bullet fired (1s delay) bullet fired (1s delay) bullet fired

Aiming

There are four options [enum] of how an enemy can *aim*:

- a) shoots directly at the player's current position
 - (this means the bullet reaches an outdated position)
- b) shoots in its forward vector (current rotation)
 - (the rotation behaviour varies – see the Movement section)
- c) shoots in front of the player, assuming no input
 - if the player drops the controller, they will get hit
 - if the player boosts/brakes or moves, they likely won't
- d) shoots in front of the player, assuming the same input
 - if the player maintains the same input, they will get hit
 - if the player changes their input, they likely won't



Spawning

Enemies spawn once the player reaches certain points on the spline; these are the *spawn markers*.

Each spawn marker maps to a *fleet* of enemies to spawn defined by:

- *basic stats* [data asset]:
 - *starting health*
 - *bullet, bullets per shot*, and *delay between bullets*
 - for flying off (after looping), the *direction, speed*, and *duration*
 - (in the future, there will also be information related to visuals)
- *behaviour* [data asset]:
 - *timeline*
 - *facing* (rotation behaviour) and *aim* (shooting behaviour)
 - *player dependency* and its *movement speed / distance*
 - *leaving the scene* and its *lifetime*
- *fleet specs* [list of...]:
 - *fleet offset* [vector]
 - *timeline delay* [float in seconds]
 - (the length of the list is the number of enemies spawned, each getting its own values of the *fleet offset* and *timeline delay*)

