# Kill-a-Round

## Log 001: (11:30am - 15/02/2020)

Session Time: 5 hours

**Goals:**

* NPC “Runner” implemented with a Behaviour tree.
* Runner runs towards the player.
* Doors open.

**Details:**

* Added “AnimStarterPack” to the project to have premade animations.
* Implemented an AI controller “NPC\_AI” and a behaviour tree, at the moment the “runner” just finds a random location then navigates towards it.
* Implemented an Animation blueprint that switches from idle to moving depending on the speed.
* Copied and edited the “idle pistol” animation to lower the hands.
* Copied and edited the “sprint\_rifle” animation to lower the hands.
* Ai perception for sight added to the runner.
* Runner now runs towards the player once he is seen.
* Runners now play the attack animation, when the hitbox is overlapped
* Implemented a door which swings open when the player presses e.

## Log 002: (10:50am – 16/02/2020)

Session Time: 2 hours

**Goals:**

* Polish the Runner
* Open the runner up for merging

**Details:**

* Created the sword in blender.
* Imported it into Unreal Engine 4, with a 0.05 scale. (tried at 1, but was too big)
* Created slots on the mannequin for the weapons.
* Attached weapons to those slots.
* [Bug: 001]: the runners can jump off each other? Probably due to clumping together.

# Living Weapon

## Log 003: (8:30am – 20/02/2020)

Session Time: 2 hours

**Goals:**

* Get the player moving
* Player is made up of multiple parts

**Details:**

* Created core sprite
* Created track sprite
* Created a new player class and each base parent class for each type of part
* Created two sockets “North” and “South” on the core sprite for testing with attaching other paper actors onto the sprite.
* Player is now made up of the core and a track, at run time.
* [Bug: 001] Player now glitches through the floor because of gravity?

## Log 004: (5:54pm – 23/02/20)

Session Time: 2 hours

Goals:

* Fix Collision and motion.

Details:

* Majority of Bug 001 was due to Ue4 still accepting a 3rd dimension. The player sprite was just behind the platform and therefore never collided.
* Bug 001 Cleared, with “tank parts” not colliding with other parts.
* Bug 001 not cleared.

## Log 005: (2:00pm – 25/02/20)

Session Time: 4 hours

**Goals:**

* Fix Collision and motion.

**Details:**

* After spending four hours bashing my brain over google and unreal figuring out the issue with bug 001. I found a single reddit post that referenced my issue: <https://www.reddit.com/r/unrealengine/comments/5dsv4v/replacing_capsule_collision_with_mesh_collision/>
* To fix bug 001, I need to make a player class by scratch.

## Log 006: (8:00am – 26/02/20)

Session Time: 3 hours

**Goals:**

* Allow parts to be added on via a call in the parent class.
* Sprite can handle multiple additions

**Details:**

* Added William’s sprites to my branch to allow easier merging.
* Added the “West”, “East” and “Center” sockets to the core sprite.
* In the Parts parent class I added the “Add Existing Part” function that takes an existing part and adds it to a non-socketed, socket.
* In the parts parent class I added the “Add New Part” function that creates the part before calling the “Add Existing part”. Note: the new part is made at <0, 0, 0> and if “add existing fails” it destroys the new part.
* Started working on making a display (mainly for debugging purposes), for dynamically creating a tank at run time. Although the code could be used in future use for editing the players tank.

## Log 007: 1:30pm (26/02/20)

Session Time: 3 hours

**Goals:**

* Add a debug window to add and remove parts

**Details:**

* Implemented a breath-first search to get an array of all parts to be rendered.
* Created a child widget to be created at run time for each part.
* Issue: When trying to allow the player to add parts, I need to link up all parts appropriately. Currently the design is like a tree, where the root node is the core part. Each node has 4 branches {North, South, East, West}, however the clients expressed a that it would check if a part is still attached. I need to be able to make a cycle, removing the tree like structure.
* Minor Issue: infinite loop within the Breath-first search, fixed with restructuring the function. [Fixed]

## Log 008: 8:15pm (27/02/20)

Session Time: 3 hours

**Goals:**

* Add a debug window to add and remove parts

**Details:**

* Added a function “LinkToMap” that gives a part a position in the map then links all the parts around that position to make a net-like data structure.
* Allowed Display of mouse when debug is pressed.
* Game now adds parts, but display doesn’t refresh, for another session
* Small bug with display not allowing additions above or left.

## Log 009: 2:30pm (28/02/20)

Session Time: 3 hours

**Goals:**

* Finish the debug window with refreshing.

**Details:**

* Cleaned the small bug once I had a rested brain.
* Set visibility of menu to hidden after the button is pressed.
* Refreshed the tank display after a part is added.
* [Bug 002]: due to socketing, the rotation for each part moves the intended socket.

## Log 010: 10:30am (29/02/20)

Session Time: 3 hours

**Goals:**

* Fix bug 002
* Player movement

**Details:**

* Fixed Bug 002, due to each socket rotation moves the intended socket, I saved the socket information of each part. To add additional logic based upon the saved socket. For example, if the saved socket was south, therefore there is no logic to where the intended socket is. However, if it was east the intended socket would rotate 90\*.

## Log 011: 6:30pm (01/03/20)

Session Time: 2 hours

**Goals:**

* Player movement

**Details:**

* Imported the improved tracks from William.
* [Bug 003]

## Log 012: 1:30pm (02/03/20)

Session Time: 3 hours

**Goals:**

* Player movement
* Merge with William

**Details:**

* Merged William’s files to my own.
* Reworked his code with him to optimise the project.
* [Bug 003]: Adding a gun to the top of a hull which is facing the east doesn’t add correctly.

## Log 012: 9:30am (03/03/20)

Session Time: 2 hours

**Goals:**

* Player movement

**Details:**

* [Bug 004]: Turns out the main reason for my movement issues was because the main sprite wasn’t the same as the core being displayed. The core should be referenced instead of sprite.
* Move the core instead of the treads.
* Added a flip book for treads.

## Log 013: 11:30am (04/03/20)

Session Time: 5 hours

**Goals:**

* Remake the debugging ui so that no menus overlap.
* Enable the tank to destroy/drop parts.
* Fix [bug 003].

**Details:**

* Made a new widget which consists of buttons to look like a drop-down list.
* Issue: I copied and pasted from part display and forgot to change the visibility. Solid 30minutes wasted.
* Added additional menus to be opened upon pressing an option. The intention is to allow the debugger to have more buttons when more and more parts are made.
* Included Special parts even though they have yet to be coded.
* Added Grid box within the buttons to accurately space the title of the button and an arrow.
* Started work on removing a “Part” from the weave at a specific position. Note: The way it is coded allows any part which is apart of the weave to destroy another part provided the position is within the weave.
* Allowed the function to bypass complex code if the part was already found (aka if it was trying to remove itself).
* The function goes like:
  + Get List of all outward links of the part to be removed.
  + Remove any non-valid options (Null)
  + For each link out, set their link inwards to null (essentially severing the part from the weave) and are they reachable to the core (located at 0, 0)?
    - If they aren’t the core itself, then reattach to the weave. Else remove yourself.
  + If they need to drop, then spawn a paper\_drop with its own information. Else destroy self.
* [Bug 006]: When any part is added near the tracks, it is also removed when that part is destroyed.

## Log 013: 1:30pm (10/03/20)

Session Time: 4 hours

**Goals:**

* Fix [bug 006].
* Fix [bug 003].

**Details:**

* Changed the part menu to display sub menues when the mouse hovers, instead of click, to speed up development.
* Found that there was no check if a socket exists when linking to the map. Which caused [bug 006]
* Fixed Bug 006
* Fixed bug 003 (it was a typo)

## Log 014: 1:30pm (14/03/20)

Session Time: 4 hours

**Goals:**

* Movement

**Details:**

* To start adding different kinds of movement, I thought moving a lot of the functionality into the core.
* Moved the calculate weight and part arrays into the core, from William’s original work.
* Now calculates the lowest point and checks if it is the hull.
* Move centre of mass to below the core.

## Log 015: 8:30am (11/05/20)

Session Time: 8 hours

**Goals:**

* Create the turret and tank enemies for the game.
* Allow the tanks to be implemented without much recoding for the level designer.

**Details:**

* Created enemy turret.
  + Static enemies which aren’t affected by gravity.
  + A scalable box for the level designer to visually see the range of the static enemy.
  + Scalable health, damage, attack rate.
* Created enemy tank.
  + Dynamic enemies which is affected by gravity.
  + Different behaviours available to the level designer, Aggressive/Defensive.
  + Scalable health, damage, attack rate.
* Created enemy helicopter.
  + Dynamic enemies which hover up and down.
  + Similar to the turret.
  + Intended to replace the hard to reach turrets, to allow the player to hit the helicopter.
* Created boss tank.
  + Dynamic enemy which hovers up and down.
  + Similar to the helicopter.
  + Two cannons and a tank constructor.

## Log 016: 9:30am (12/05/20)

Session Time: 6 hours

**Goals:**

* Polish the enemies created yesterday.
* Have a visual indicator for the player to know when an enemy is almost dead and about to fire.

**Details:**

* Added bars to the side of each enemies’ cannon. Scales up to full based on a delay timer. Scaling up on size and colour (green to red).
* Added bars to the top of each enemy to indicate the current health of the enemy.
* Additionally, polishing to each enemy for the level designer.
* Added a wall location for the final boss to remove coding requirement for the level designer.
* Reworked the projectile physics to inherit the velocity of the launcher. Allowing the player and enemies to move and shoot.