Off the Ground Games

What Lies Beneath

Fiverr Milestone Overview - Milestone #5 Completion

Links

Combat Logic

Using the Gun

In the **_get_input()** function, if **gun_out_state**, **aim_state** are true, and the **"use_weapon"** was just used, we take the **bullet_ray** and find the first collision in the ray it hits. It will check to see if the collision object is in the group of **"Monster"**. If it is, the collision object has **receive_damage** called and blood spray to occur on that position.

For damage, base Gun Damage is determined by **PlayerInventory.get_gun_damage()**, which just pulls the equipment damage value. A damage multiplier is added if the collision the **bullert_ray** is intersecting with a collision in the group **"CriticalHit"**.

Using the Crowbar

The Crowbar attack is an animation that's triggered in **_get_input()** by pressing the key for **"use_weapon"**, while not in any other important states. This plays the animation and turns on monitoring for the crowbar's collision box. If no melee weapon is collected/equipped, and base attack is 0, no animation will be played.

If the melee collision box intersects with any areas, **check_hit(area)** is called. We track to see if we have already hit this monster before(avoiding hitting the same creature multiple times in one swing), and pass on the damage with **melee_hit(melee_damage)**, where the **mele_damage** is from **PlayerInventory.get_melee_damage()**. We also pass on the **blood_spray** like the gun function.

Taking Damage

Monster's have a small collision box around the creature's **GroundPosition**. If a collision box enters in this box, and is in the group **"Player"**, we call **receive_damage(damage)** to the parent node of the colliding unit.

The player node then just passes on the value to **PlayerState.receive_damage(damage)** and allows for the new calculation in the AutoLoaded Script.

Death

When **PlayerState** updates the Player's health with **set_Player_Health(new_health)** is under or equal to 0, a signal is sent out from the **PlayerState** AutoLoad script called **"player_died"**.

Currently, on **_ready()** for the player node connects the AutoLoaded Player Node for **"player_died"** signal connected to **_on_player_death()**.

This will set the dead flag to true, change the animation to "Death" and plays the death screen.

The **DeathScreen** is a simple canvas scene that plays a color hue'd fog, with just a simple script that makes the text flash, and reload the scene if the any key is pressed once the original tween is completed (and the "Press any key" is shown on the screen).

Syringe Logic

To use a Syringe you just need to call **PlayerInventory.use_syringe()**.

This then pulls a list of Syringes based off of the const array, **SYRINGE_LIST**. This list should be ordered from weakest to strongest.

After pulling out all the potential syringes to use in your inventory, it goes through the array and uses the first syringe that does not 'over heal' you (attempting to heal past your Max Health). If it must 'over heal' the player, it will use the lowest valued syringe.

Milestones

Player Movement Milestone (Commit 07baaef in Movement)

- ☑ ~~General Movement~~
    - ☑ ~~Walking~~
    - ☑ ~~Running (Hold-Shift)~~
    - ☑ ~~Hidden Stamina Bar~~
    - ☑ ~~Climb Ladder~~
    - ☑ ~~Fall/Drop~~
- ☑ ~~Interactions~~
    - ☑ ~~Push/Pull Box/Climb Box~~
    - ☑ ~~Kneeling (Looting/Interacting)~~
    - ☑ ~~Interactable Object (Shimmer)~~
- ☑ ~~Basic Combat~~
    - ☑ ~~Aiming (across the walkable axis)~~
    - ☑ ~~While Aiming you can only Walking~~
    - ☑ ~~Button to pull out gun~~
- ☑ ~~Game Camera~~
    - ☑ ~~Dynamic Camera Primarily on Character~~
- ☑ ~~Additional Features~~
    - ☑ ~~Splashscreen~~
    - ☑ ~~Project Image~~
    - ☑ ~~Level Change~~

Menu System Milestone(Commit f218118 in Menu)

- ☑ ~~Silent hill style (scrolling inventory)~~
    - ☑ ~~Options to: Use, Reload, Equip, Remove~~
    - ☑ ~~Gun Equipment Slot~~
    - ☑ ~~Still Screenshot of the Game in the top left with colour overlay indicating health~~
- ☑ ~~Map option - Placeholder Screen~~

- ☑ ~~Memo - Journal Pages~~
  - ☑ ~~Left right over Journal Objects~~
- ☑ ~~Important Objects~~
  - ☑ ~~Shows Keys~~
  - ☑ ~~Shows Map Fragments~~
- ☑ ~~Options with: Music and Sound Control~~
- ☑ ~~Global Player State~~
  - ☑ ~~Health~~
  - ☑ ~~Equipped Item~~
  - ☑ ~~Inventory~~
  - ☑ ~~Important Inventory Items~~
  - ☑ ~~Map Fragments~~
  - ☑ ~~Location~~

Dialogue Milestone (Commit 8f60f51 in Dialog)

- ☑ ~~Dialog with branching options as well as object Inspection~~
- ☑ ~~Interact-able Objects in Level where it triggers dialogue~~
- ☑ ~~Interact-able Objects in Level where it triggers text above the player's head~~
- ☑ ~~Save Dialog Choice Tree in Global Player State~~
- ☑ ~~Additional Features~~
  - ☑ ~~Animation for scene changes~~
  - ☑ ~~Voice Over for Journal Entries~~

Map System Milestone (Commit 632a2d6 in Map-System4)

- ☑ ~~Collecting fragments of maps to reveal more of the map (accessible via the menu system)~~
- ☑ ~~Metroid Style Map where map highlights which location you are in~~
  - ☑ ~~Track if you have been in the space~~
  - ☑ ~~Track if the door is locked (show by X)- agreed in chat that coloured door direction is easier to develop/manage~~

Combat Milestone (Commit 840a9e2 in Combat-Milestone-5)

- ☑ ~~Player Combat~~
    - ☑ ~~Shooting Gun While Aiming~~
    - ☑ ~~Melee Animation~~
    - ☑ ~~Check on Hit/Provide Damage~~
    - ☑ ~~Taking damage~~
    - ☑ ~~HP Recovery (Syringe)~~
    - ☑ ~~Game Restart~~
- ☑ ~~Player Inventory~~
    - ☑ ~~Check Equipped Item (Gun)~~
    - ☑ ~~Provide Damage on Check~~

Enemies Milestone / Save Point Milestone

- ☐ Enemy AI Types Created
    - ☐ The Lost – The remnants of a person. Doesn't focus on the player and shuffles on its own, until it sees the player and goes straight to the player. Takes 2-3 hits
    - ☐ Rabid Dog – Sprints to player on line of sight.
- ☐ Enemy Interactions
    - ☐ On Player Touch, Damages Player
    - ☐ Attack Animations
        - ☐ Attack hitboxes
        - ☐ Critical hitbox damage
    - ☐ On Kill, Enemy is destroyed and removed
    - ☐ On Entry, Enemy States are stored and loaded back

- ☐ Object in world where the character can save
- ☐ Start Menu
    - ☐ New Game
    - ☐ Load Game
    - ☐ Quit
- ☐ On Death, Load Last Save