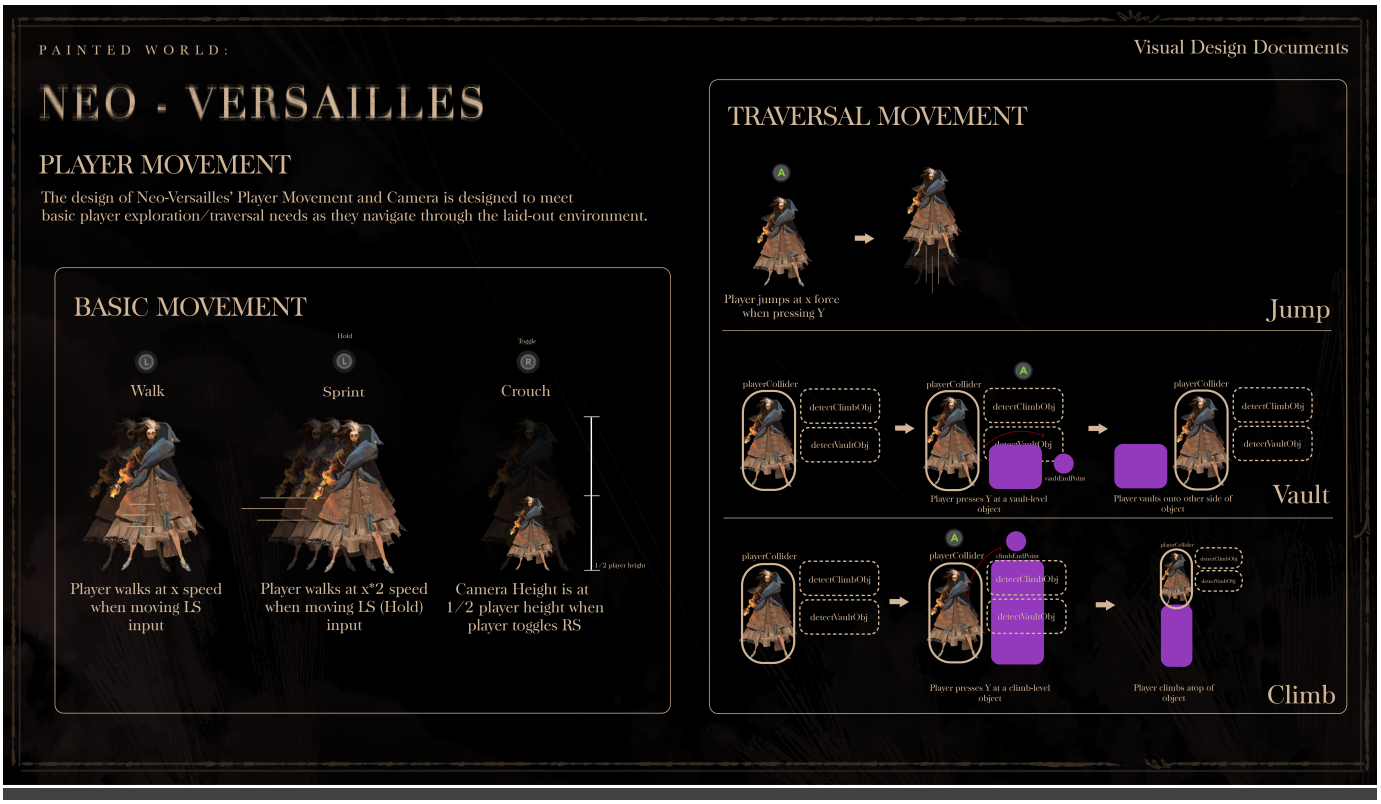
Player Movement/Camera TDD



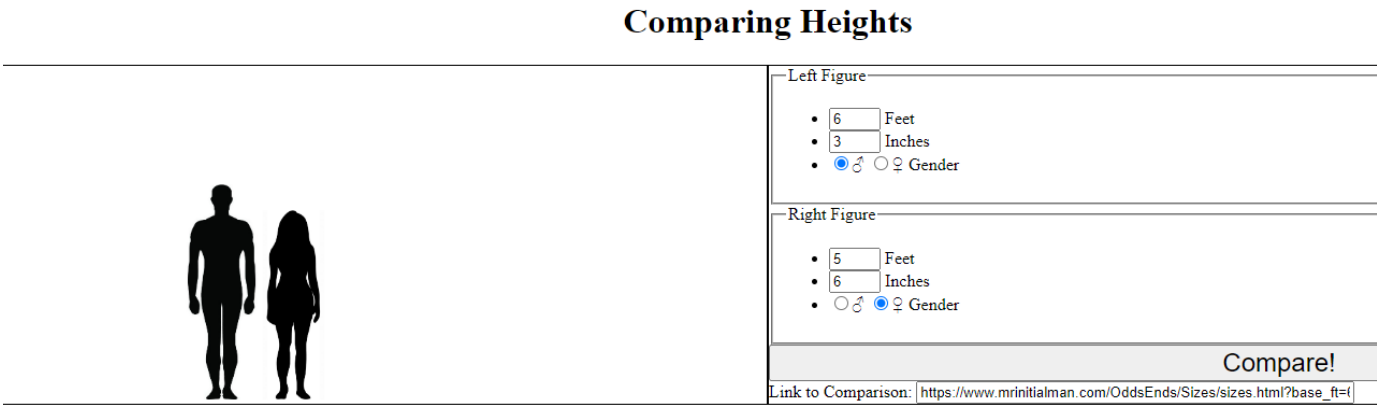
Note: This system presented below is intended to be a bare bones system which will be expanded upon with the direction of the design team. Most of this design is conceptual and subject to change.

Overview

The design of Neo-Versailles' Player Movement and Camera is designed to meet basic player exploration/traversal needs as they navigate through the laid-out environment.
Neo-Versailles_Project 2022-04-12 13-14-16_Trim.mp4

Player Movement Notes

The player's movement will directly impact the enemy's detection range, as outlined in theEnemy AI TDD



The player character model should be approximately 5'6" (**1.7m**) in game and the Demon of Despair should be approximately 6'3" (**1.92m**) for reference.

Fall Damage - Players will take **x** damage if they fall at a height of **x** from their current surface

General Technical Attributes:

Components:

- **Player Camera (Cinemachine Virtual Camera)**
  - Height - **1.7 m** (5’6”)
  - Cinemachine Noise Component : Basic Multi-Channel Perlin
    - This will allow us to play around with the camera shake to make the player controller feel less static
    - Be toggleable from Pause Menu + Options Menu (In-Game)
    - Have the **amplitude** and **frequency** be editable variables that can be changed dependent on the type of player movement
      - **Idle**
      - **Walk**
      - **Sprint**
      - **Vault**
      - **Climb**
      - **Jump**
      - **Slide**
      - **Crouch**
- **Character Controller**
- **Player Model**
  - Player Animations
    - Walk
    - Sprint
    - Jump
    - Vault
    - Climb
    - Crouch
- **maxSpeed :** Fixed float, max speed for walking and sprinting is fixed, due to animation pipeline efficiency
- **curSpeed :** Variable float, ramps up from 0 maxSpeed over based on acceleration variable
- **isGrounded** : Ground Detection to prevent infinite jump
- **timeToCrouch** - 0.1f
- **maxStamina :** Fixed float, max stamina that will allow the player to perform **sprinting**
- **curStamina :** Variable float for player **stamina**

Vertical Movement Parameters

- **Gravity -** 10f
- **Jump Timeout** - 0.1f
- **Fall Timeout** - 0.15f
- **Terminal Velocity** - 45f
- **Max Interaction Distance** - 2.5f

Vault / Climb Parameters

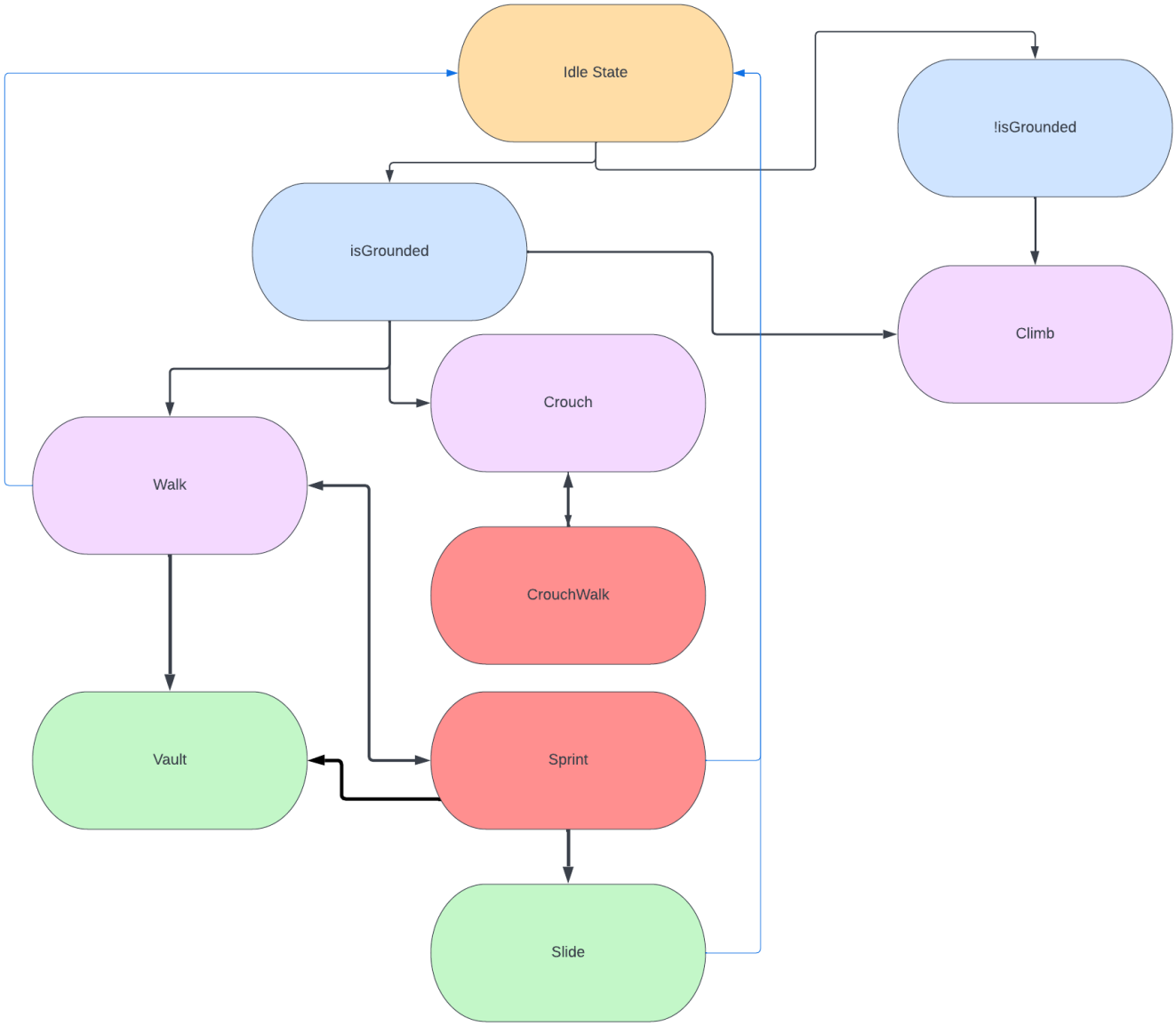
- **Climb Speed -** 2f
- **Climb End Point** - A Fixed Transform that the player Climbs to the Point of (Translation)
- **Vault End Point -** A Fixed Transform that the player vaults over to

Camera Parameters

- **Upper Clamp** - 45f
- **Lower Clamp** - 50f

- **Vault Object Tag (Wall)** : Tag that checks for a vaultable object based on collision
- **Climb Object Tag (WallClimb) :** Tag that checks for a climbable object based on collision
- **SlideObj :** Tag placed onto ramps that temporarily set maxSpeed to 1.8x the normal amount if a player slides on them
  - Note players can still slide on normal horizontal ground, but maxSpeed will build up as opposed to slow down
- **groundCheck :** Checks for if player **isGrounded**
- **Fall Damage :** Player takes **x** damage based on the **z** translation taken into account (distance of fall)
  - TakeFallDamage()
    - **playerHealth -= (dmgMultiplier)changeInZ - damageBuffer**
      - changeInZ - taking into account the fall distance (zStarting) point from when the player last was **isGrounded** to the next
    - Damage should **not** occur if the changeInZ is **less than or equal to damageBuffer based on the formula**

Player Movement Actions



The Player Movement / Traversal System should be able to perform the following tasks:

- Basic Movement
- Traversal Movement
- **Xbox Controls / PC Controls**

Basic Movement

The Basic Player Movement Controller should be able to perform the following tasks:

- Movement
  - Walking
  - Sprinting
- Jumping
- Player Positions
  - Standing
  - Crouching

Walking

Triggered on **Left Stick / WASD**, default speed of player character when moving. **3f moveSpeed**.

Sprinting

Triggered on **Left Stick +Left Stick (Hold) /** **LShift + WASD**, sprinting is the secondary speed of the player character, considerably faster than default movement speed at around **1.7f speed factor**. As a player **sprints** their **stamina** will decrease by **Time.deltaTime** for however long the sprint lasts. **If stamina reaches 0**, the player will be forced back into a **walkSpeed** and **will not be allowed to sprint** until **stamina replenishes fully (cooldown)**. If the player **does not use up all their stamina**, this **cooldown** will not be applied before the player can sprint again.

Jumping

Triggered on **A Xbox Gamepad Button /** **Spacebar**, jumping can be performed by the player through this input as well as the requirement of needing to be **isGrounded**.

For vertical jump movement, the **jumpHeight** should be set to **1.2m** in the game editor.

For horizontal jump movement, the maximum horizontal translation the player should be able to achieve from performing this action is **4.5m**

Standing / Idle

Basic Player Position, standing upright.

Crouching / Crouch Walking

Triggered on **Right Stick (Toggle) /** **Left Ctrl Key** Secondary Player Position, crouching, speed is lessened by **x.**

Interaction with Enemy Detection - When the player is **crouched** and **behind an obstruction** (wall, rubble, etc.), the **Enemy Detection Range** will be **lessened**.

Traversal Movement

Vaulting

Triggered on **A Xbox Gamepad Button** / **Spacebar** vaulting can be performed by the player through this input as well as the requirement of needing to be **isGrounded** and has their **detectVaultObj** collider triggered, as well as if the player is **Sprinting** or **Walking**.

When the player is **vaulting**, the player translational movement should have a defined **playerOffset** for the vertical movement.
Regarding **vaulting**, the **vertical** and **forward** movement should occur **separately** to account for potential animation movement and clipping errors.

With various vaultable objects in mind of differing heights and width, the following list of vaultables are listed **below (Possibly Redact This)**

- Pillar Objects - **1 up (x) 1 across (z) translation**
- Pipe Objects - **1 up (x) 1 across (z) translation**
- Couch - **1 up (x) 1 across (z) translation**
- Balcony01 - **.7 up (x) .2 (z) translation**
- Balcony02 - **1 up (x) .2 (z) translation**
- Fencing - **1 up (x) .2 (z) translation**

Climbing (from Ground)

Triggered on **A Xbox Gamepad Button /** **Spacebar** vaulting can be performed by the player through this input as well as the requirement of needing to be **isGrounded** and has their **detectClimbObj** collider triggered.

When the player is **climbing**, the player translational movement should have a defined **playerOffset** for the vertical movement.
Regarding **climbing**, the **vertical** and **forward** movement should occur **separately** to account for potential animation movement and clipping errors.

Climbing (from Air)

Triggered on **A Xbox Gamepad Button /** **Spacebar** vaulting can be performed by the player through this input as well as the requirement of needing their **detectClimbObj** collider triggered.

Sliding

Triggered on **LS Press Gamepad Button** / **LCtrl + WASD** sliding can be performed by the player through this input as well as the requirement of needing to be **isGrounded** and if the player is **Sprinting.** In the **case** the player's collider also collide with a **detectSlideObj** collider, the player velocity increases for whatever time the player is sliding and in contact with the slide Object.

Camera Control

Basic Camera Control, ability to look in all directions (vertical, horizontal) based on player **Right Stick / Mouse Input** Input.

https://docs.google.com/spreadsheets/d/1ZMKe0aU06cNMm97HE0bOdqYpirUIbwSI97Y2W5T_yLY/edit#gid=1634443476

Art Needs

- Player Character
  - Model
  - Texture
  - Rig
- Player Movement Animations
  - **NOTE All translational movement of the player character will be handled programmatically in-game**
  - Idle
  - Walk
  - Sprint
  - Jump
  - Crouch
  - Slide
  - Vault
  - Climb
  - Stumble

Engineering Needs

- Player Movement Functionality
  - Basic Movement
    - **Walk** at **x speed**
    - **Sprint** at **1.7x walk speed**
    - **Jumping** at **x jumpHeight**
    - **Crouching** at **1/2 player height (0.85m)** and **0.5 walk speed**
  - Traversal Movement
    - **Vaulting** at **1/2 player height (0.85m)**
    - **Climbing** at **player height (1.7m)**
    - **Sliding**
- Collision detection
  - **isGrounded** : Ground Detection to prevent infinite jump
  - **detectVaultObj** : Collision detection that checks for a **vaultable** object
  - **detectClimbObj :** Collision detection that checks for a **climbable** object
  - **detectSlideObj : STRETCH GOAL** Collision detection that detects **ramps** that would increase player velocity when **sliding** is performed
- Fall Damage
  - Take Damage function based on **(changeInZ - damageBuffer)**