FLAIL BASH

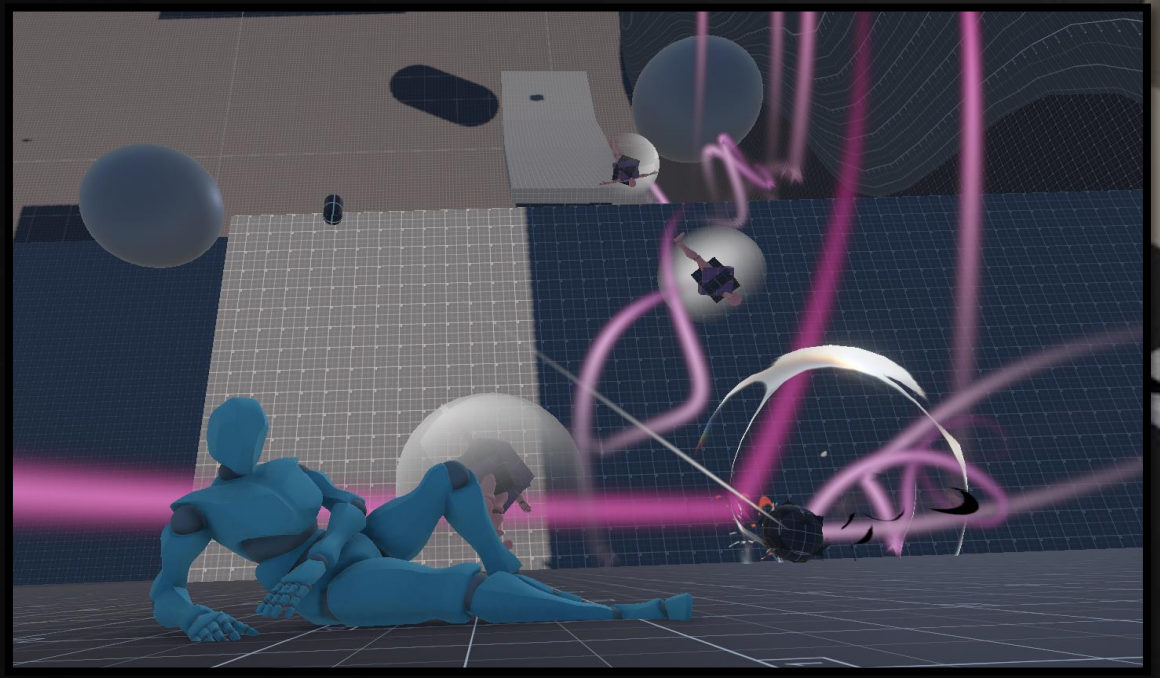
Stylized Physics-Based Movement & Weapon System

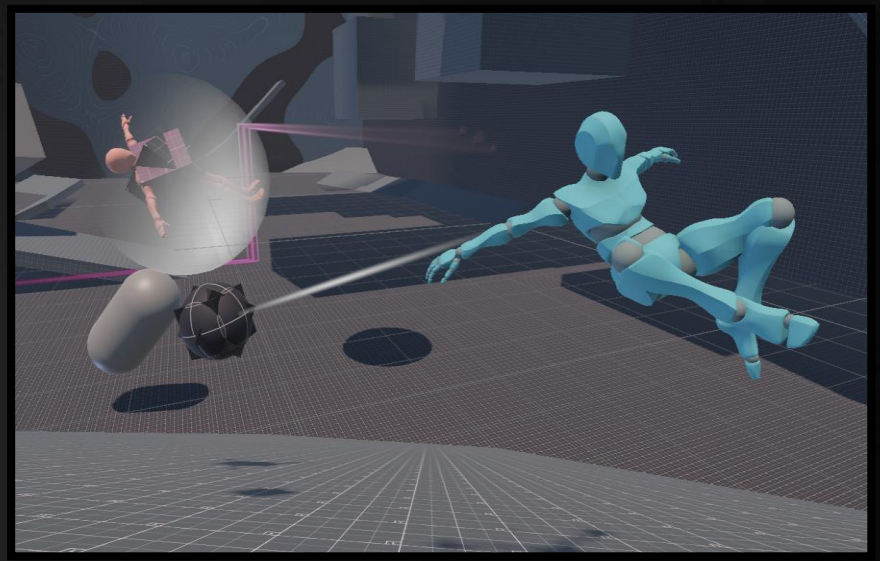


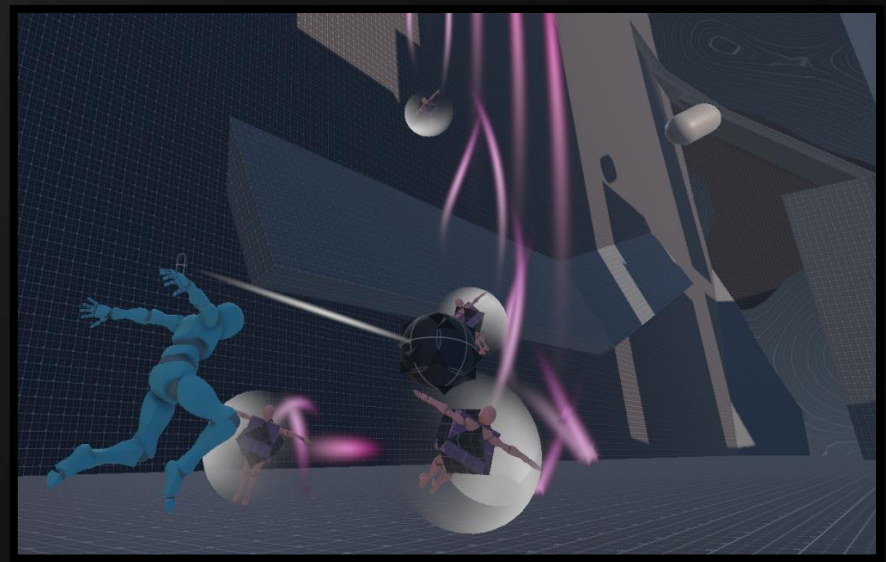
- GANDI HARI TANMAI, G013000M

PROTOTYPE GOALS

- Roguelike Survival
- Build Momentum
- Different Terrains due to Gravity Shift
- Build Flail's Swing Power to Strike







RESEARCH & INSPIRATIONS

- MEGABONK
- GRAVITY FORGE
- TOTALLY ACCURATE BATTLE SIMULATOR
- PHYSICS-BASED FLAILS

















MOSCOW

FLAIL BASH MOSCOW

MUST HAVE

- PHYSICS BASED FLAIL WEAPON WITH ACCESSIBLE CONTROLS THAT CAN LAUNCH THE FLAIL BALL AND ALSO RETRACT IT.

- GRAVITY SHIFTING TO ONE OF THE 6 DIRECTIONS EVERY N SECONDS: FORWARD, BACKWARD, UP, DOWN, LEFT, RIGHT

- A CUBIC ARENA THAT HAS ATLEAST TWO DIFFERENT TERRAINS ON DIFFERENT SIDES TO CHANGE THE MOVEMENT DYNAMICS OF IN-GAME CHARACTERS AFTER THE SHIFT IN GRAVITY'S DIRECTION.

- FORCE / MOMENTUM CALCULATION TO DETERMINE WITH HOW MUCH FORCE THE FLAIL BALL HITS SOMETHING. SCALES WITH PLAYER'S + THE BALL'S VELOCITIES.

- UI SHOWING PLAYER'S CURRENT FORCE.

- SMOOTH SLIDING OR SKATING MECHANIC TO HELP INCREASE PLAYER'S MOMENTUM.

SHOULD HAVE

- FLAIL SIZE SCALING WITH DISTANCE TRAVELLED AWAY FROM THE PLAYER. THE BALL'S SCALE WILL RESET TO INITIAL SIZE AFTER EITHER HITTING SOMETHING OR FULLY RETRACTING IT TO IT'S HANDLE.

- CUBIC ARENA WHERE ALL SIDES OF IT HAS UNIQUE TERRAINS.

- THE ARENA'S TERRAINS DESIGNED TO FACILITATE / ENCOURAGE SLIDING AND JUMPING HIGH TO BUILD UP FORCE.

- SEMI-RAGDOLL ANIMATIONS TO MAKE THE CHARACTERS REACT TO PHYSICS IN A MORE PLAUSIBLE WAY.

- ENEMY HORDES THAT SCALE IN NUMBER AND BOSSES THAT SPAWN EVERY NTH WAVE.

- BASIC VFX AND SFX TO EMPHASIZE THE IMPACT CAUSED BY THE FLAIL.

COULD HAVE

- THE LENGTH LIMIT OF THE FLAIL'S CHAIN SCALED BY THE CURRENT FORCE.

- ONE ACTIVE TERRAIN AT A TIME ON THE ARENA, WITH THE REST OF THE SIDES BEING FLAT / SUPPRESSED TO NOT INTERRUPT THE PLAYER AND THE ACTIVE ONES SHOULD EMERGE FROM THE FLAT PLANE. ARENA'S ACTIVE SIDE COULD EMERGE OUT OF THE PREVIOUSLY FLAT PLANE RATHER THAN SUDDENLY APPEARING. (DEPENDS ON MAP DESIGN)

- DESTRUCTABLE ENVIRONMENT.

- KILL COUNTER / POINTS SYSTEM

- FLAIL POWER-UPS

- A NON-HOSTILE GIANT WHICH IS HARD TO DEFEAT. PLAYERS REQUIRE GREAT FORCE TO BREAK IT AND GET BONUS PTS.

WON'T HAVE

- ENEMY TYPES OR BOSS TYPES AS THEY CAN REPLACE THE EXISTING ONES IN A MODULAR WAY IN THE FUTURE IF NEEDED.

- COMPLEX ENEMY AI BEHAVIOURS.

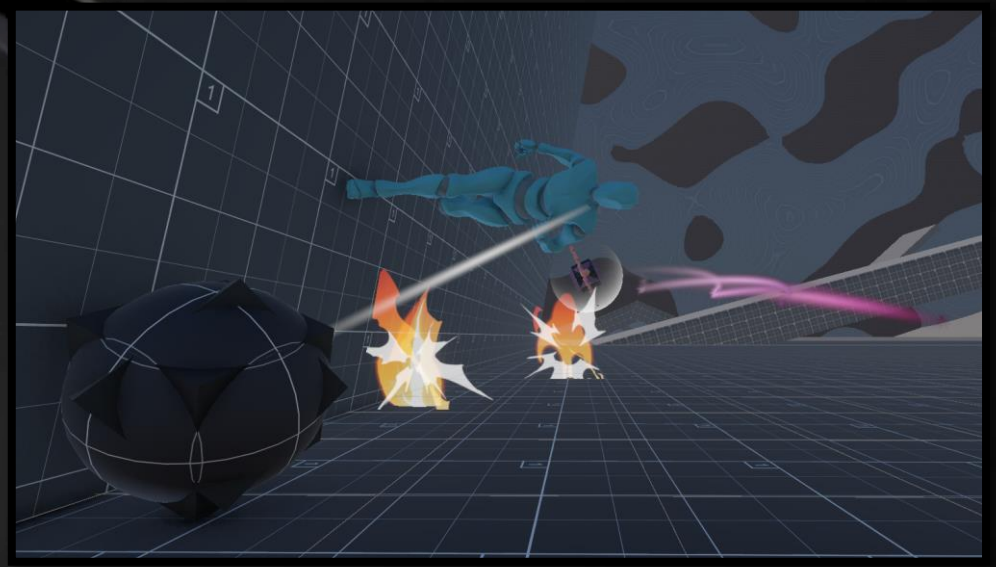
- POLISHED 3D MODELS OF CHARACTERS OR TERRAINS.

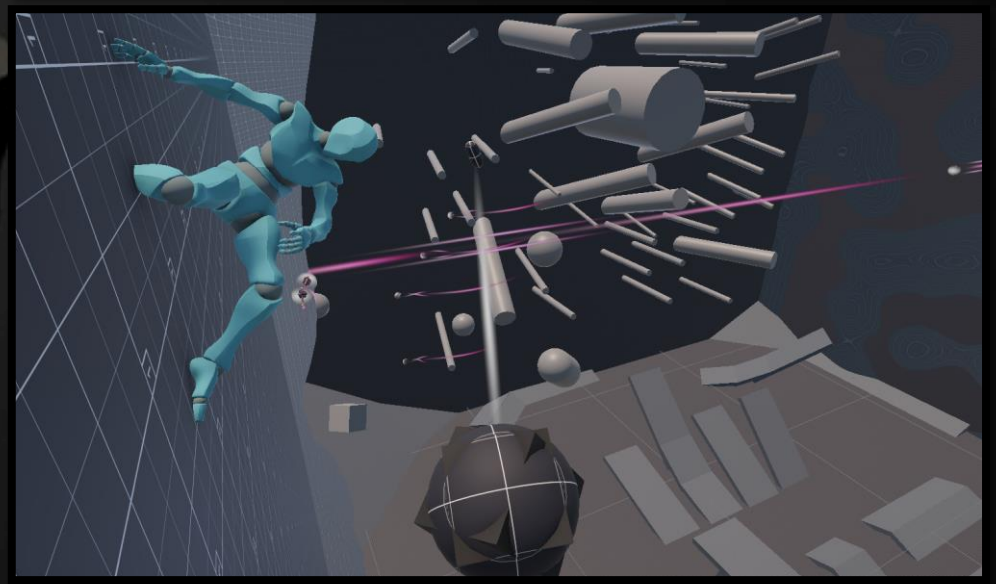
- MULTIPLAYER SINCE IT COULD BE TIME CONSUMING, LEAVING LESS TIME TO FULLY POLISH THE CORE MECHANICS.

- POLISHED UI FEATURES, MAIN MENUS, PAUSE SCREENS, SETTINGS, ETC.

CORE GAME ELEMENTS

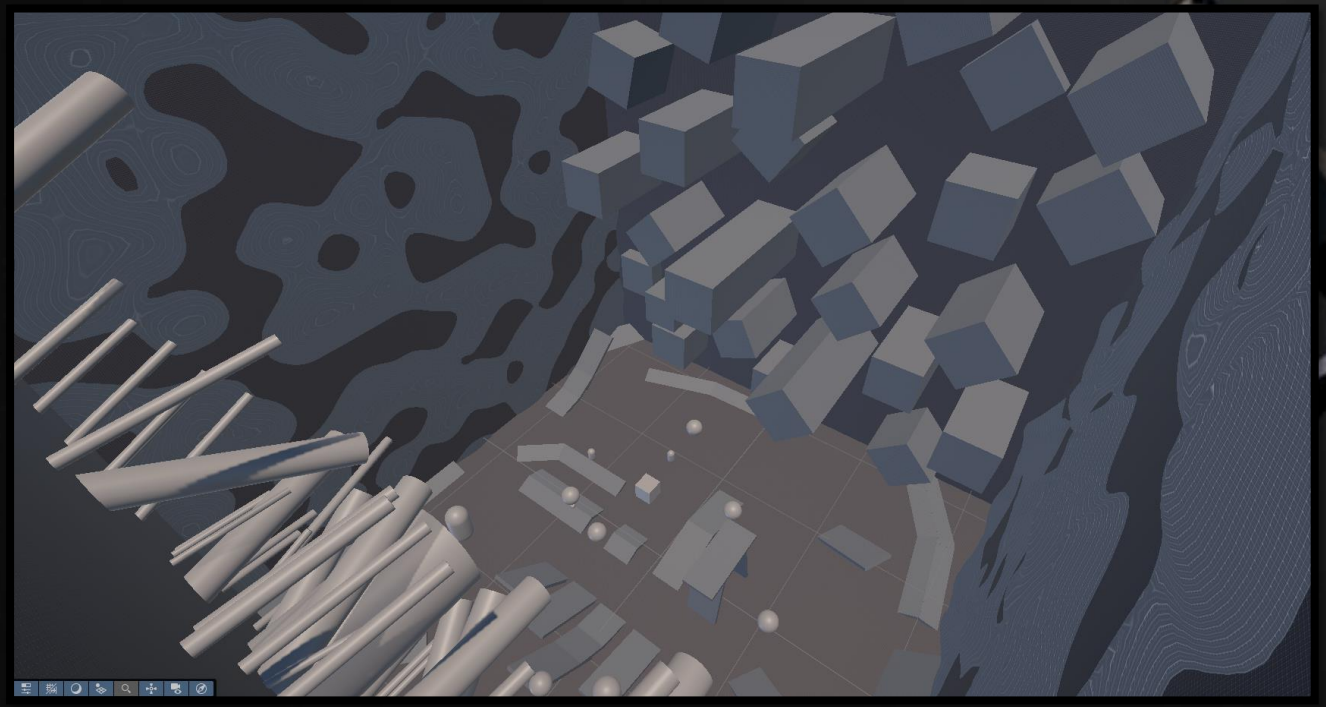
- Gravity Shifting
- Arena Terrains
- Momentum-Based Movement State Machine
- Animation
- Flail Weapon Controls
- Flail Weapon Physics

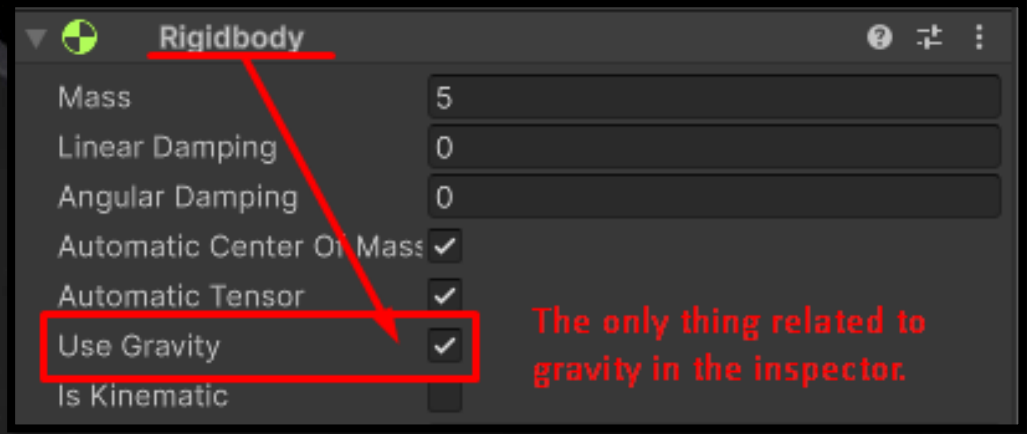




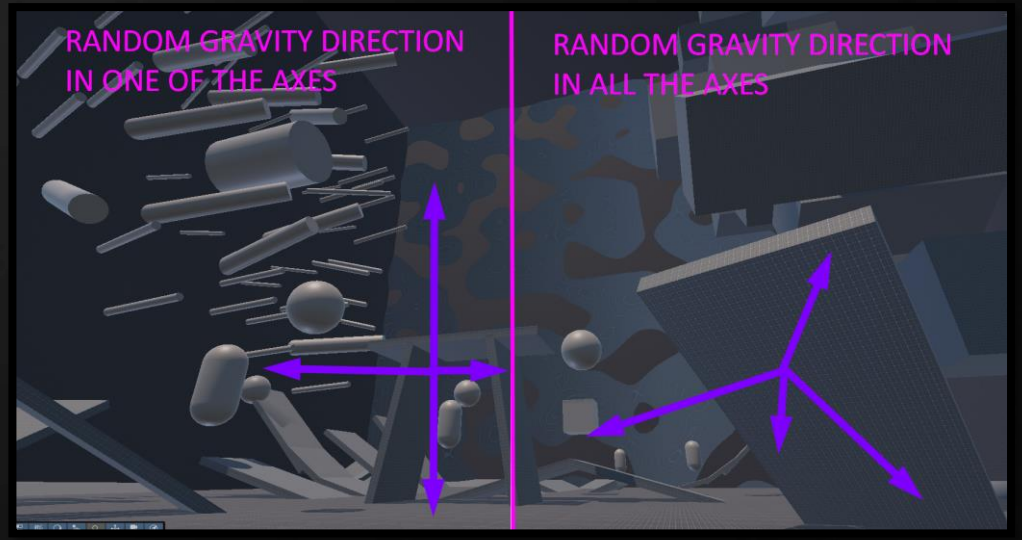
GRAVITY SHIFTING

- Implemented before others
- Random direction
- Physics.gravity



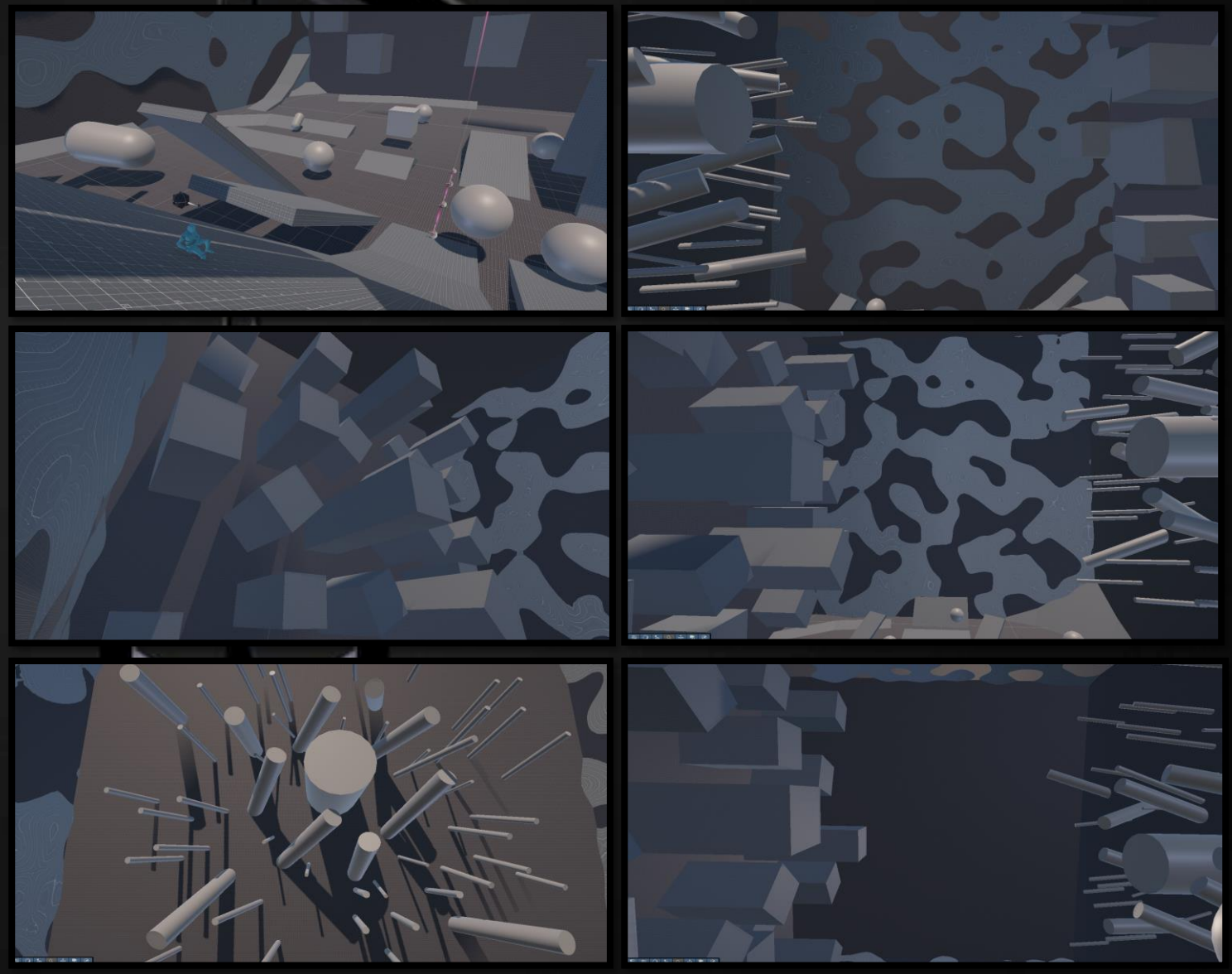



```
Vector3 newGravity = m_Gravity.normalized * -9.81f * m_GravityScale;
Physics.gravity = newGravity;
```



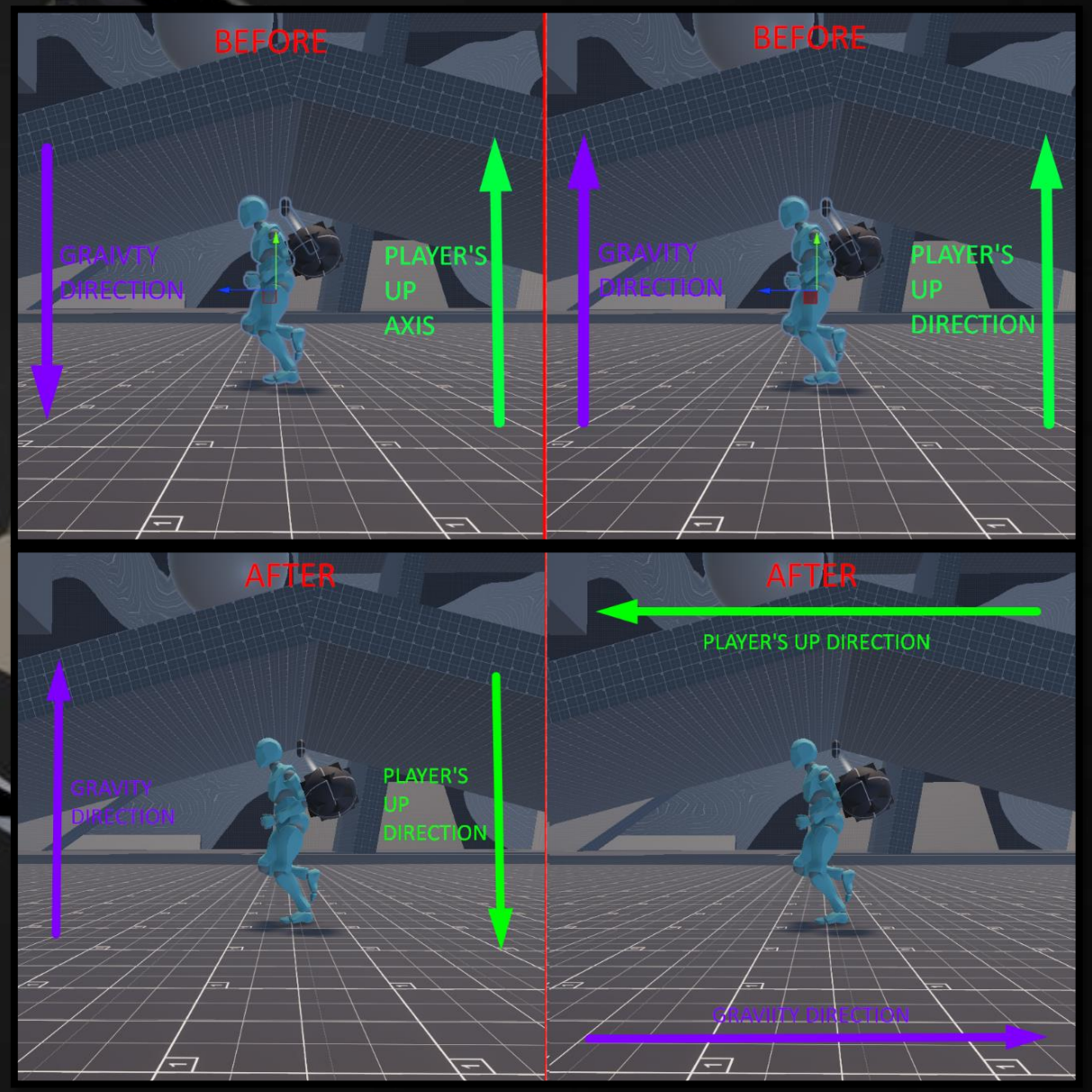
ARENA TERRAINS

- Skate Park
- Cuboid/Buildings
- Pillars
- Perlin Noise Plane 1
- Perlin Noise Plane 2
- Flat Plane



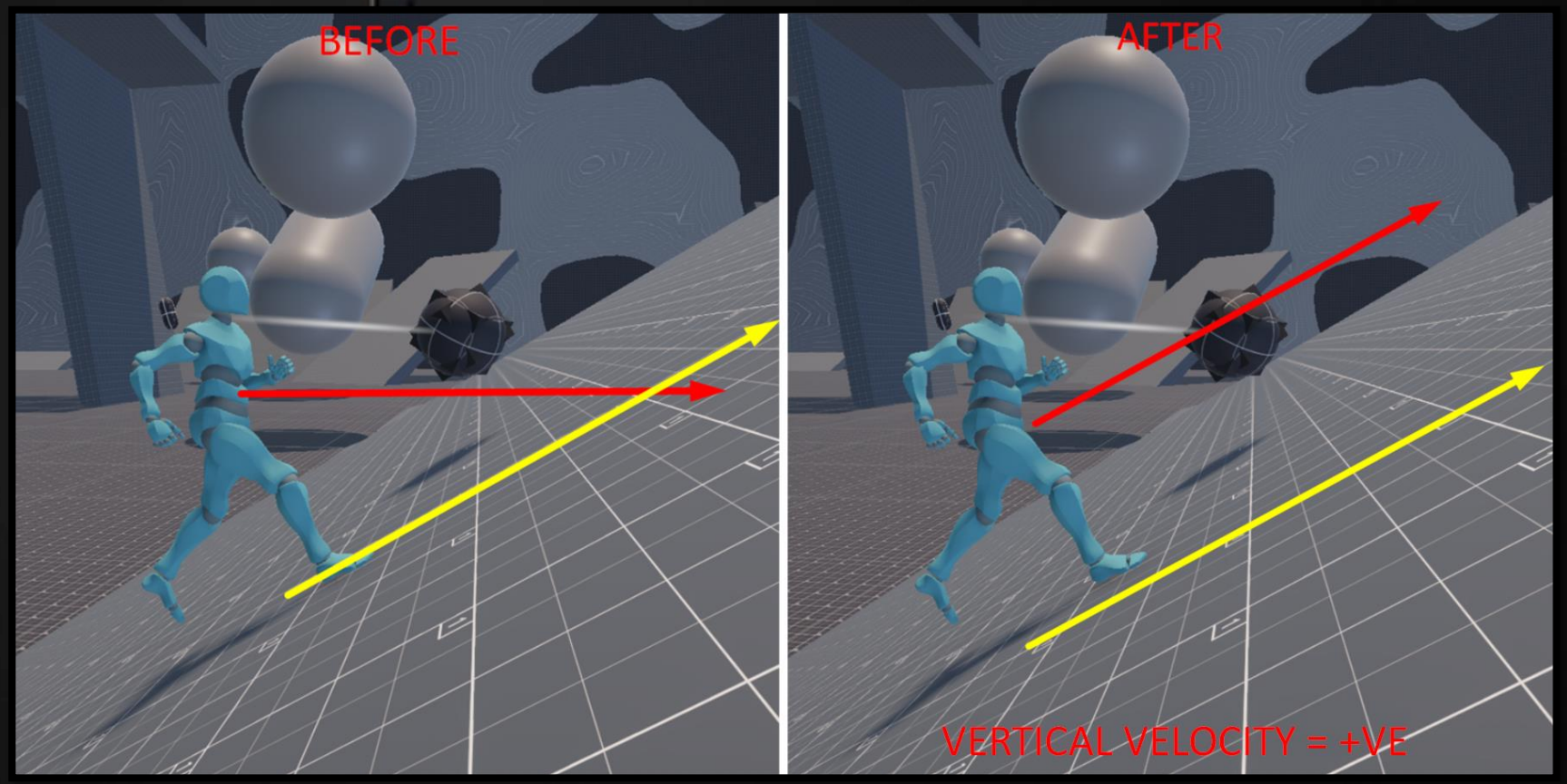
MOVEMENT MECHANICS

- Movement
- Slope Movement
- Jump
- Slide
- Movement State Machine



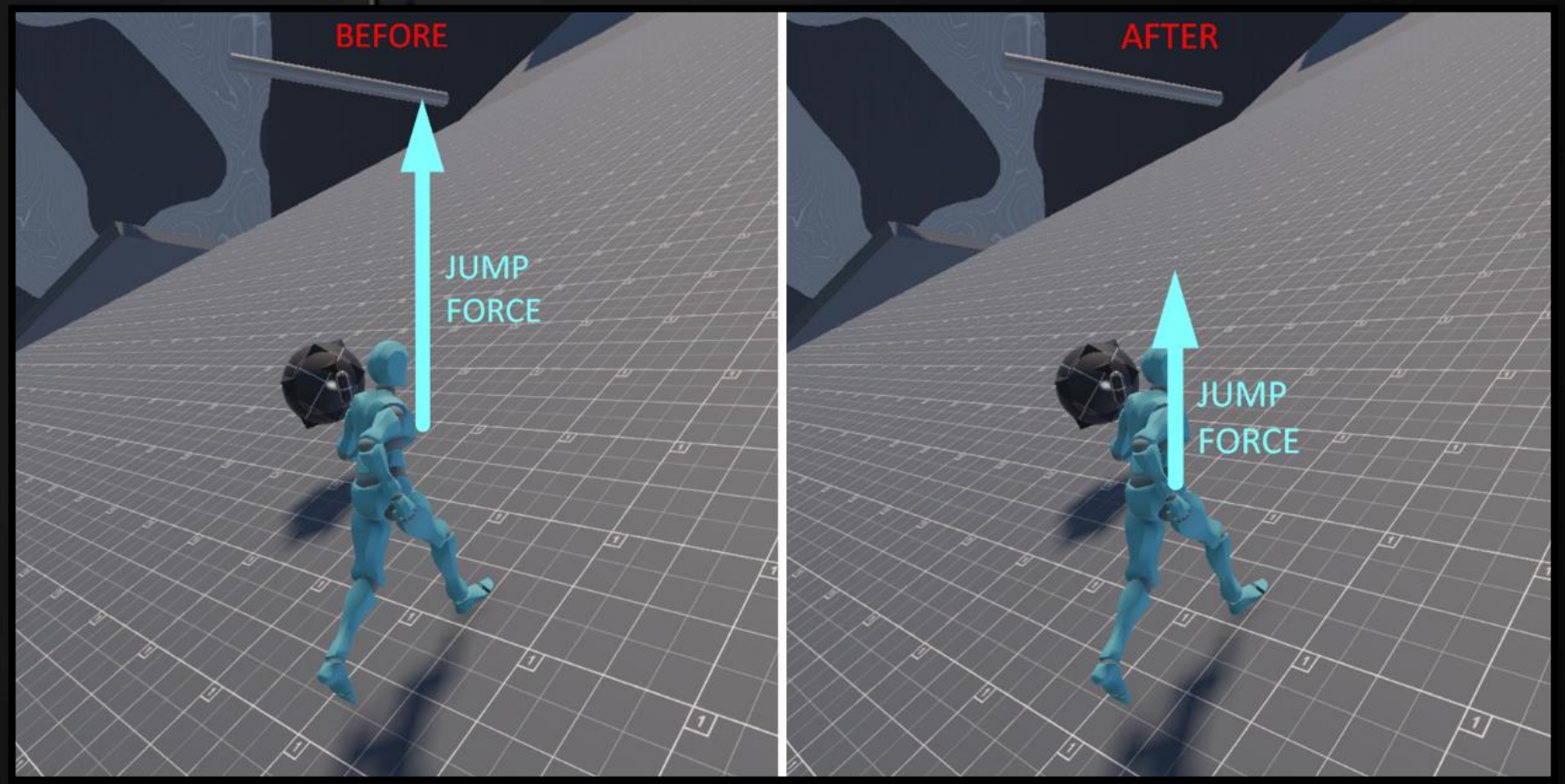
MOVEMENT MECHANICS

- Movement
- Slope Movement
- Jump
- Slide
- Movement State Machine



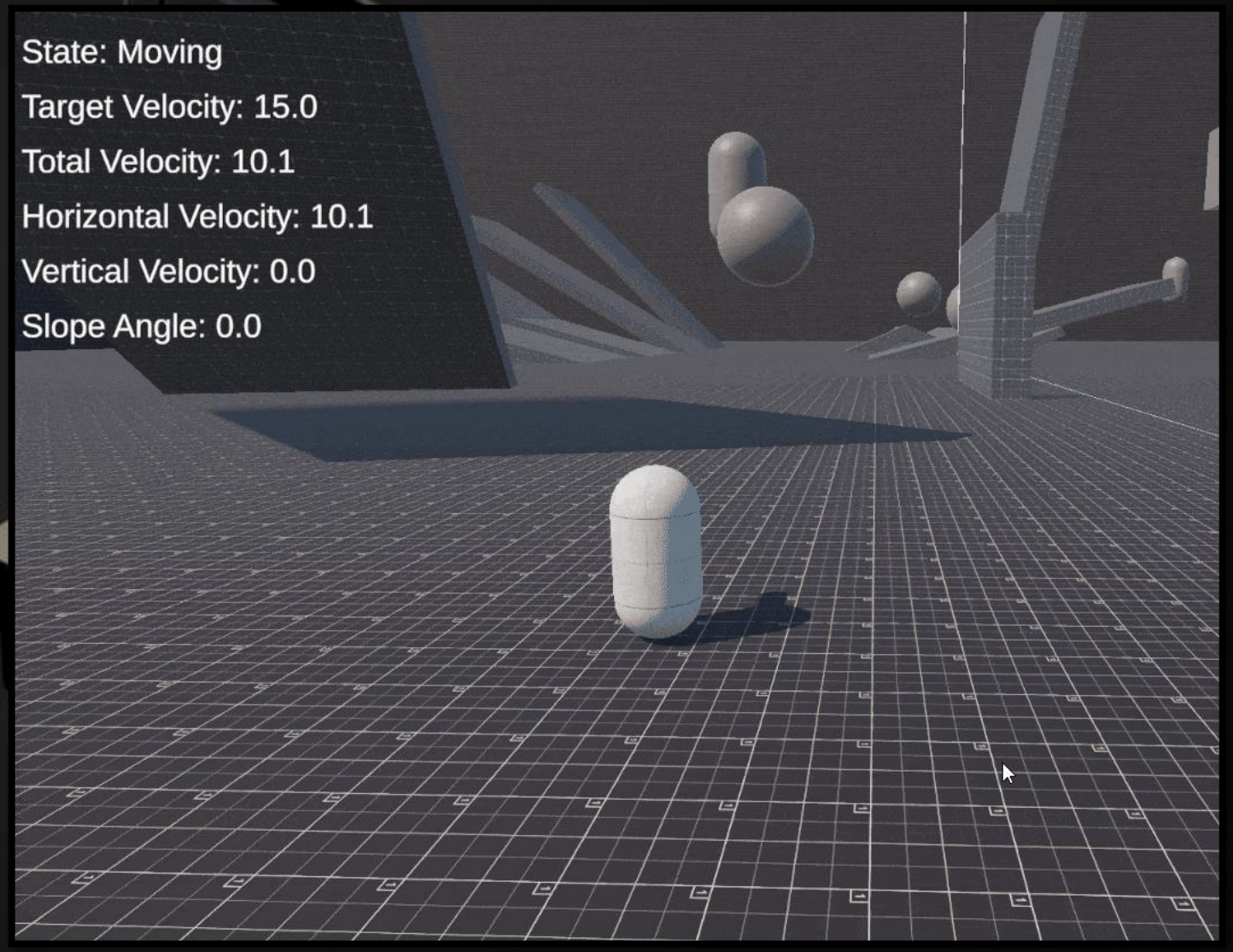
MOVEMENT MECHANICS

- Movement
- Slope Movement
- Jump
- Slide
- Movement State Machine



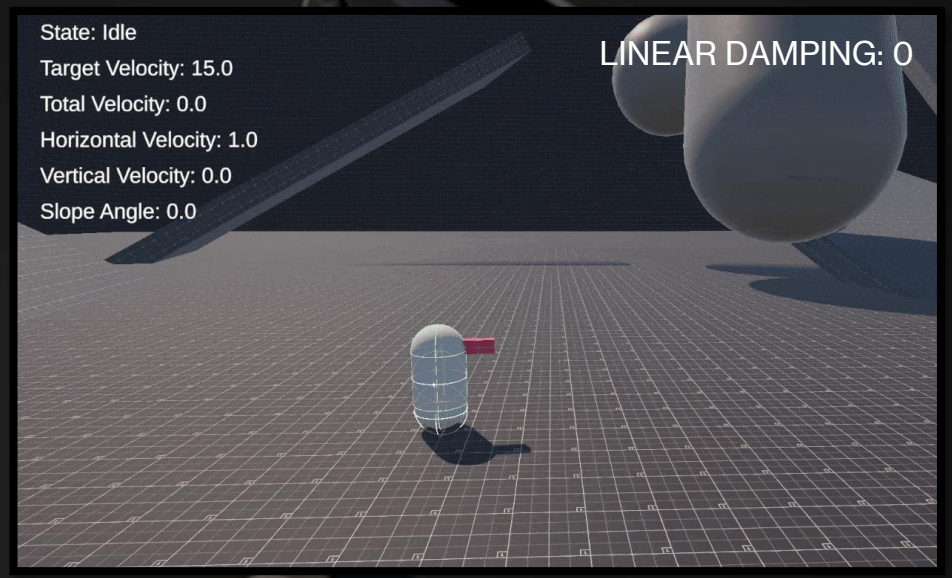
MOVEMENT MECHANICS

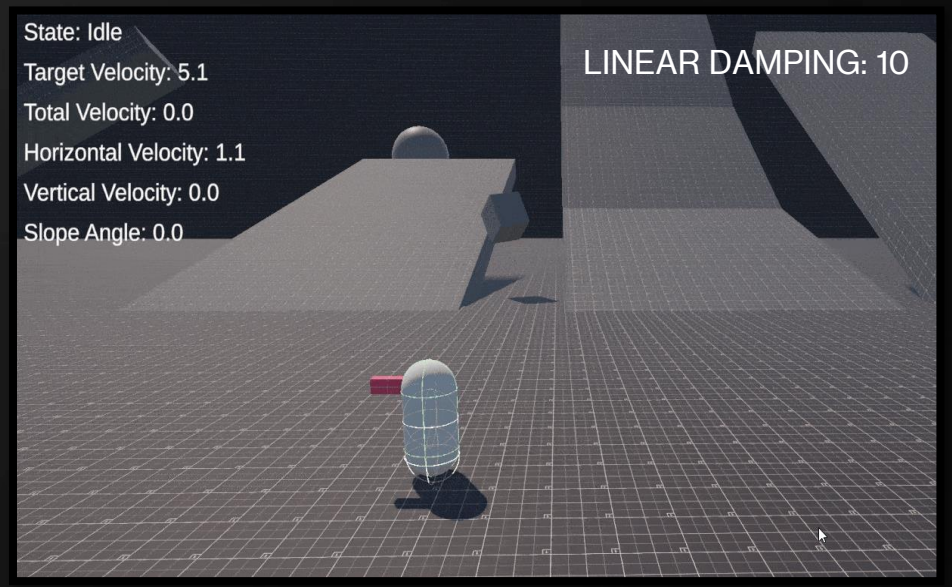
- Movement
- Slope Movement
- Jump
- Slide
- Movement State Machine



STATE-BASED MOVEMENT

- Idle
- Crouching
- Moving – OnSlope, OffSlope
- Sliding – OnSlope, OffSlope
- Floating
- Gliding
- Rigidbody linear damping





STATE-BASED MOVEMENT

- Idle
- Crouching
- Moving – OnSlope, OffSlope
- Sliding – OnSlope, OffSlope
- Floating
- Gliding
- Rigidbody linear damping



```
[Header("State Based Settings")]
public float m_DefaultLinearDamping = 2f;
public float m_SlideLinearDamping = 0.1f;
public float m_AirLinearDamping = 0.05f;
```

```
switch (PlayerMovementStateMachine.m_CurrentState)
{
    //GROUNDED STATES
    case PlayerMovementStateMachine.State.Idle:
        m_rb.AddForce(-currentVel, ForceMode.Acceleration);
        return;

    case PlayerMovementStateMachine.State.Moving:
        if (PlayerMovementStateMachine.m_bSloped)...
        else...
        return;

    case PlayerMovementStateMachine.State.Sliding:
        if (PlayerMovementStateMachine.m_bSloped)...
        else...
        return;

    //NOT GROUNDED STATES
    case PlayerMovementStateMachine.State.Gliding:...

    case PlayerMovementStateMachine.State.Floating:...
}
```

STATE-BASED MOVEMENT

- Idle
- Crouching
- Moving
- Sliding
- Floating
- Gliding







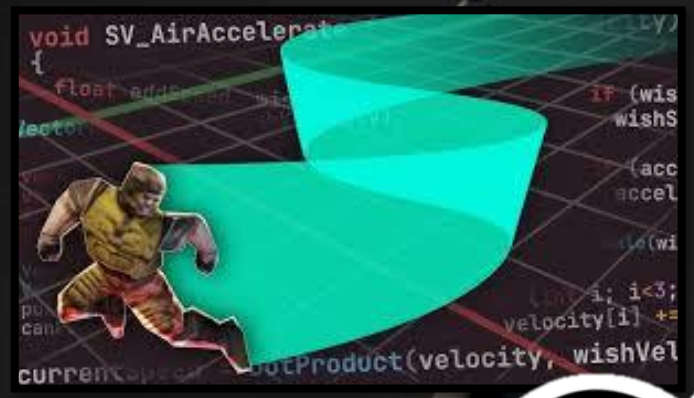






STATE-BASED MOVEMENT

- Bhop/Air Strafing



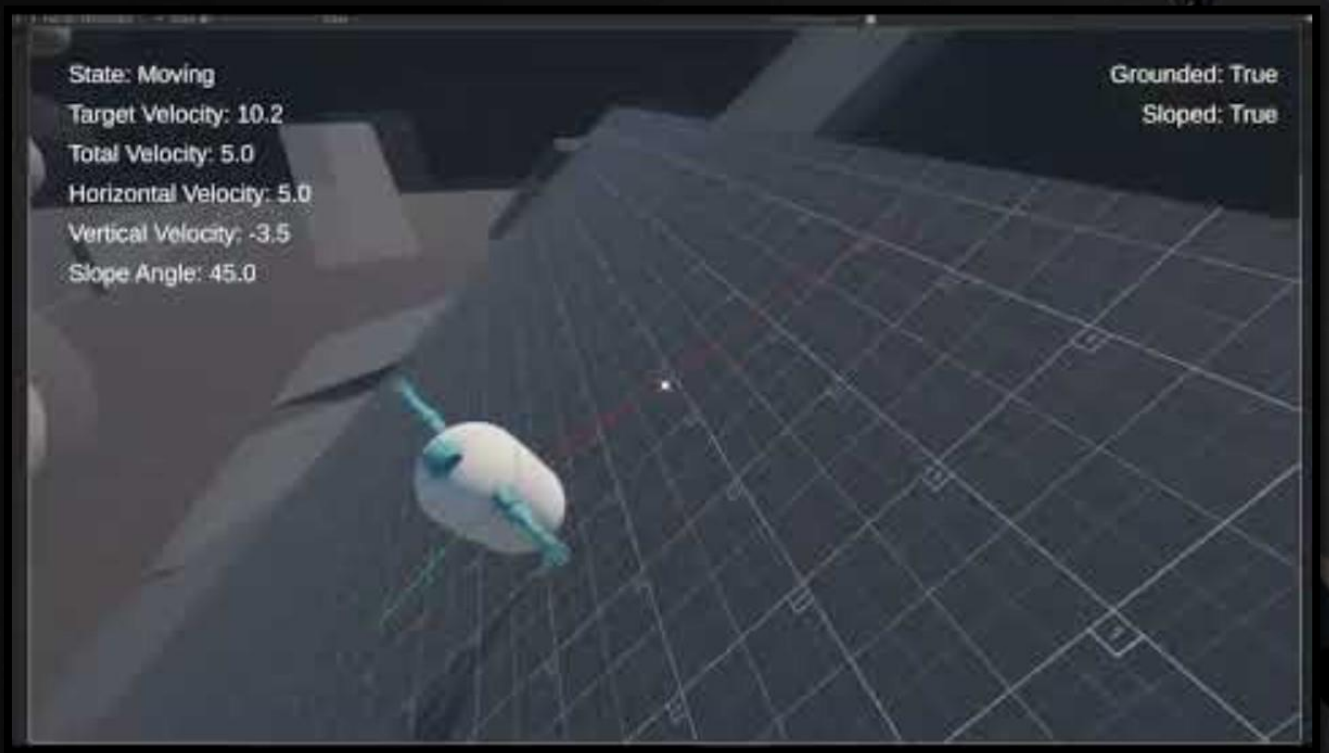


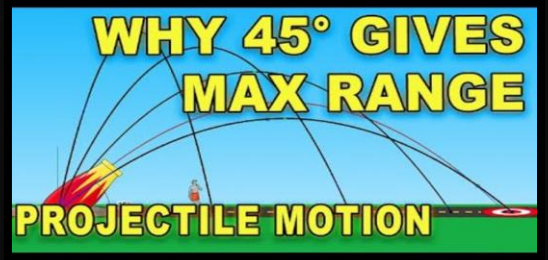




BUNNYHOP

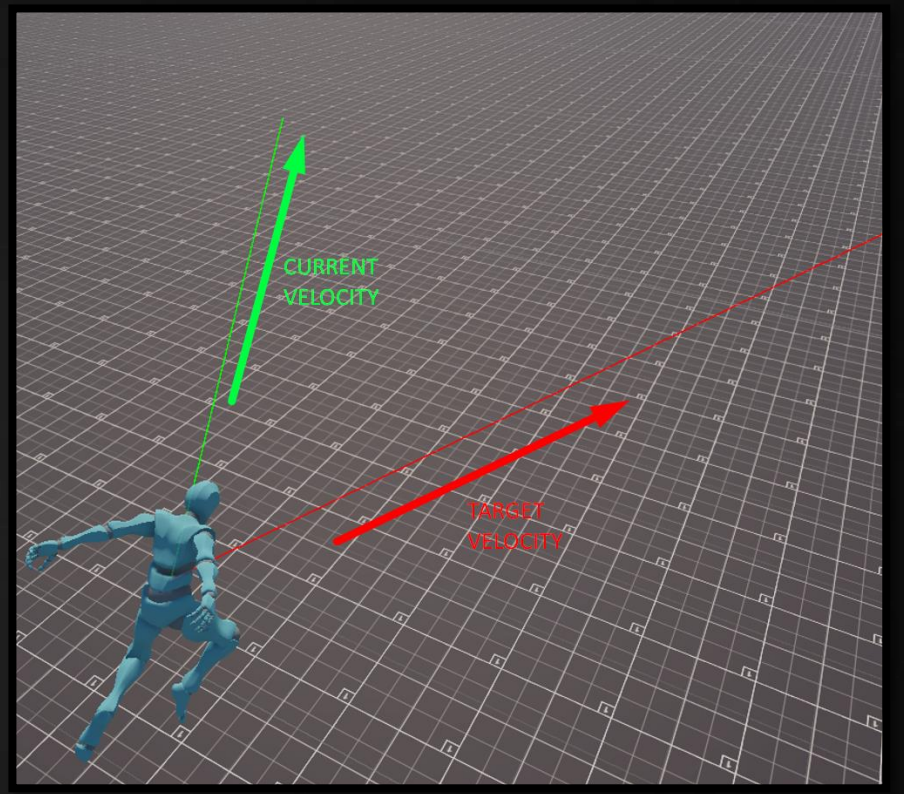
BHOPPING

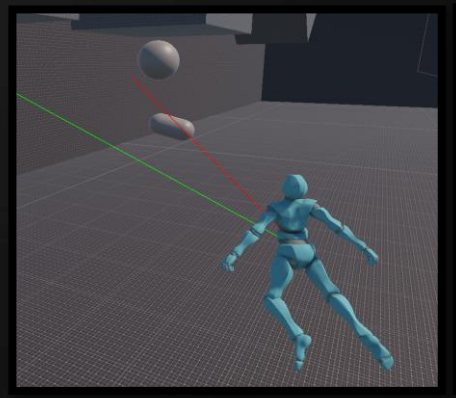


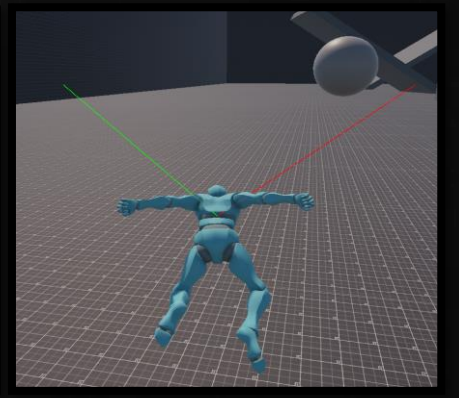


$$\frac{\sin 2\theta \, v^2}{g} \qquad \theta = 45^\circ \qquad R_x \text{ max}$$

$$V_y = V \sin\theta$$





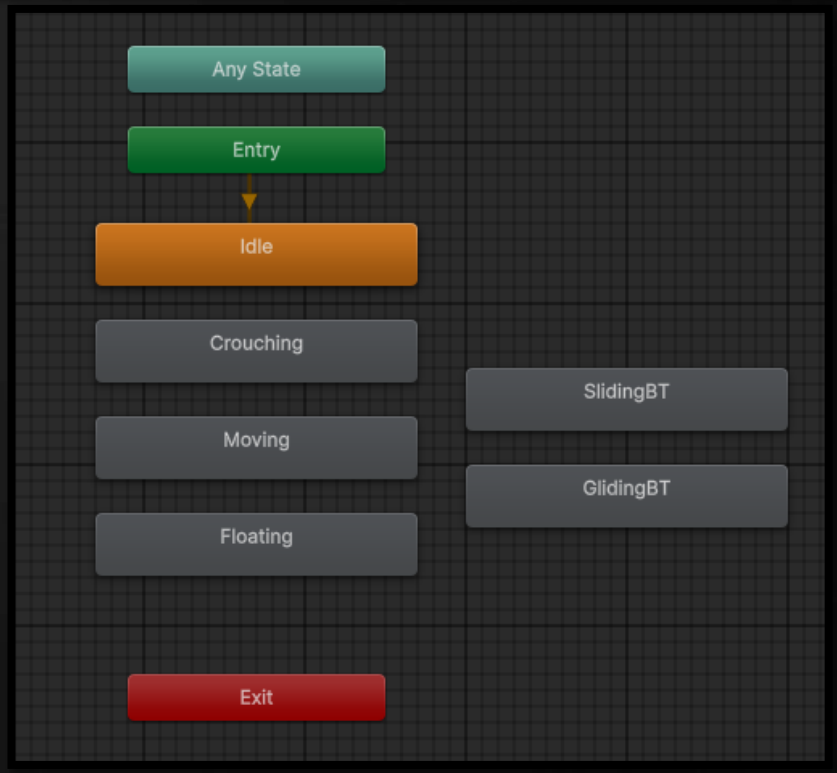


ANIMATION

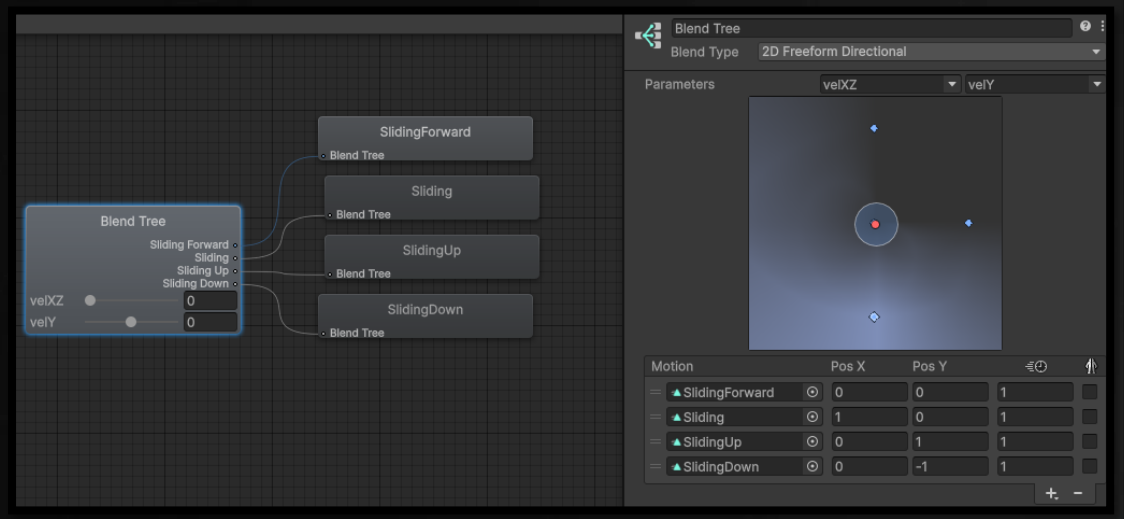
- Code-Based Animation State Machine
- 2D Blend Trees
- Stop-Motion Animation

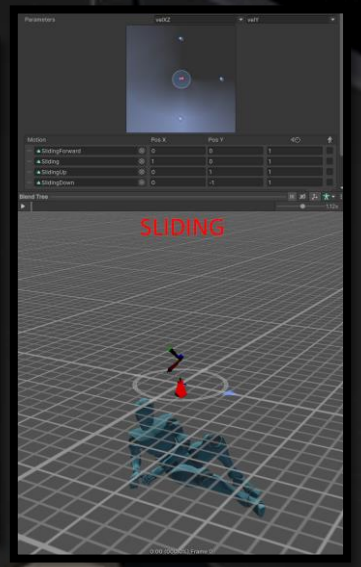


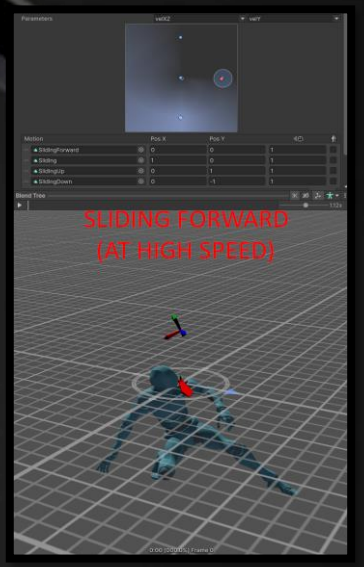

```
switch (PlayerMovementStateMachine.m_CurrentState)
{
    case PlayerMovementStateMachine.State.Idle:
        ChangeAnimation("Idle");
        break;
    case PlayerMovementStateMachine.State.Crouching:
        ChangeAnimation("Crouching");
        break;
    case PlayerMovementStateMachine.State.Floating:
        ChangeAnimation("Floating");
        m_Animator.speed = (m_rb.linearVelocity.magnitude / 8f);
        break;
    case PlayerMovementStateMachine.State.Moving:
        ChangeAnimation("Moving");
        m_Animator.speed = (scr_PlayerChar.currentVel.magnitude / 10f);
        break;
    case PlayerMovementStateMachine.State.Sliding:
        ChangeAnimation("SlidingBT");
        break;
    case PlayerMovementStateMachine.State.Gliding:
        ChangeAnimation("GlidingBT");
        break;
}
```

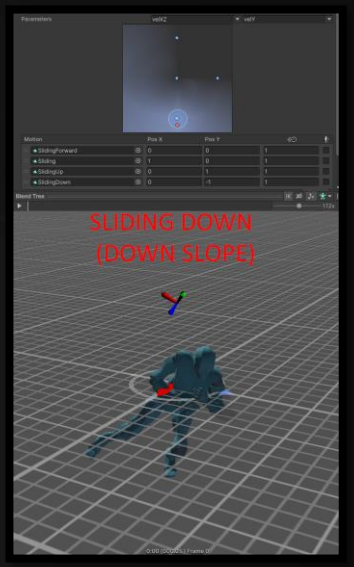


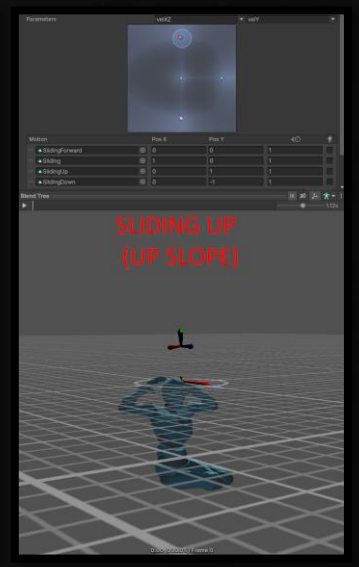
ANIMATION: BLEND TREES

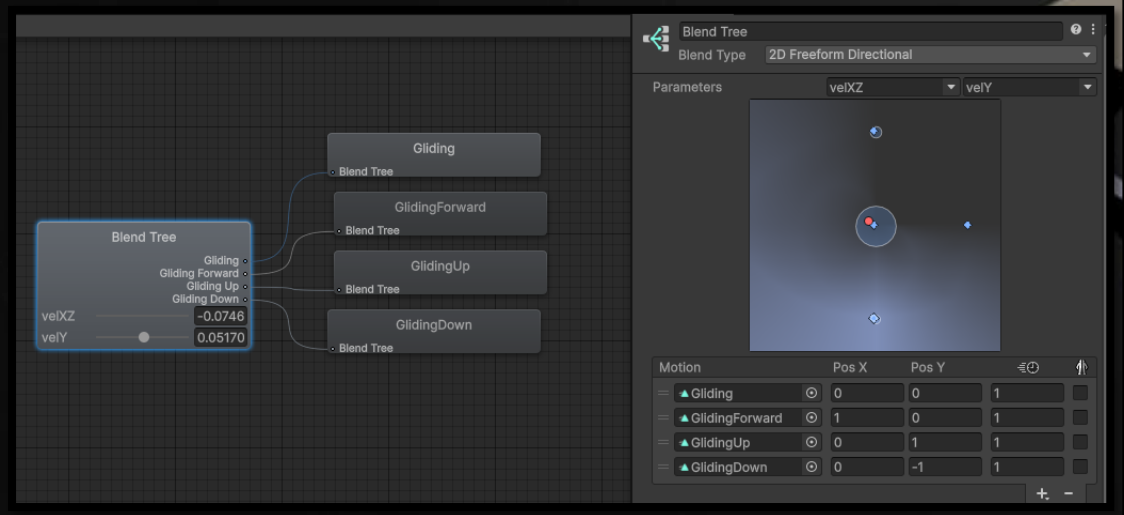






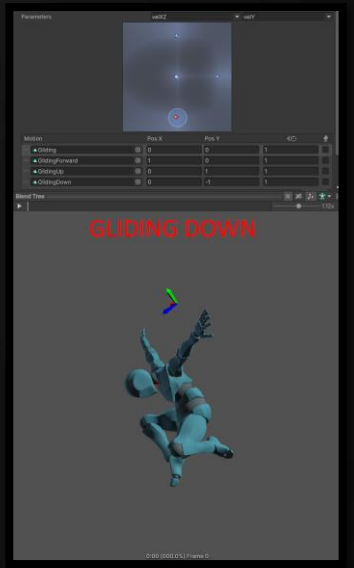


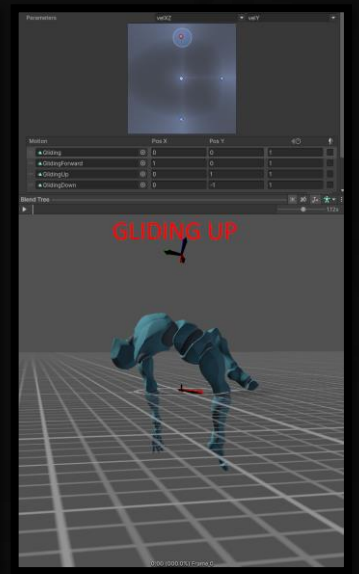






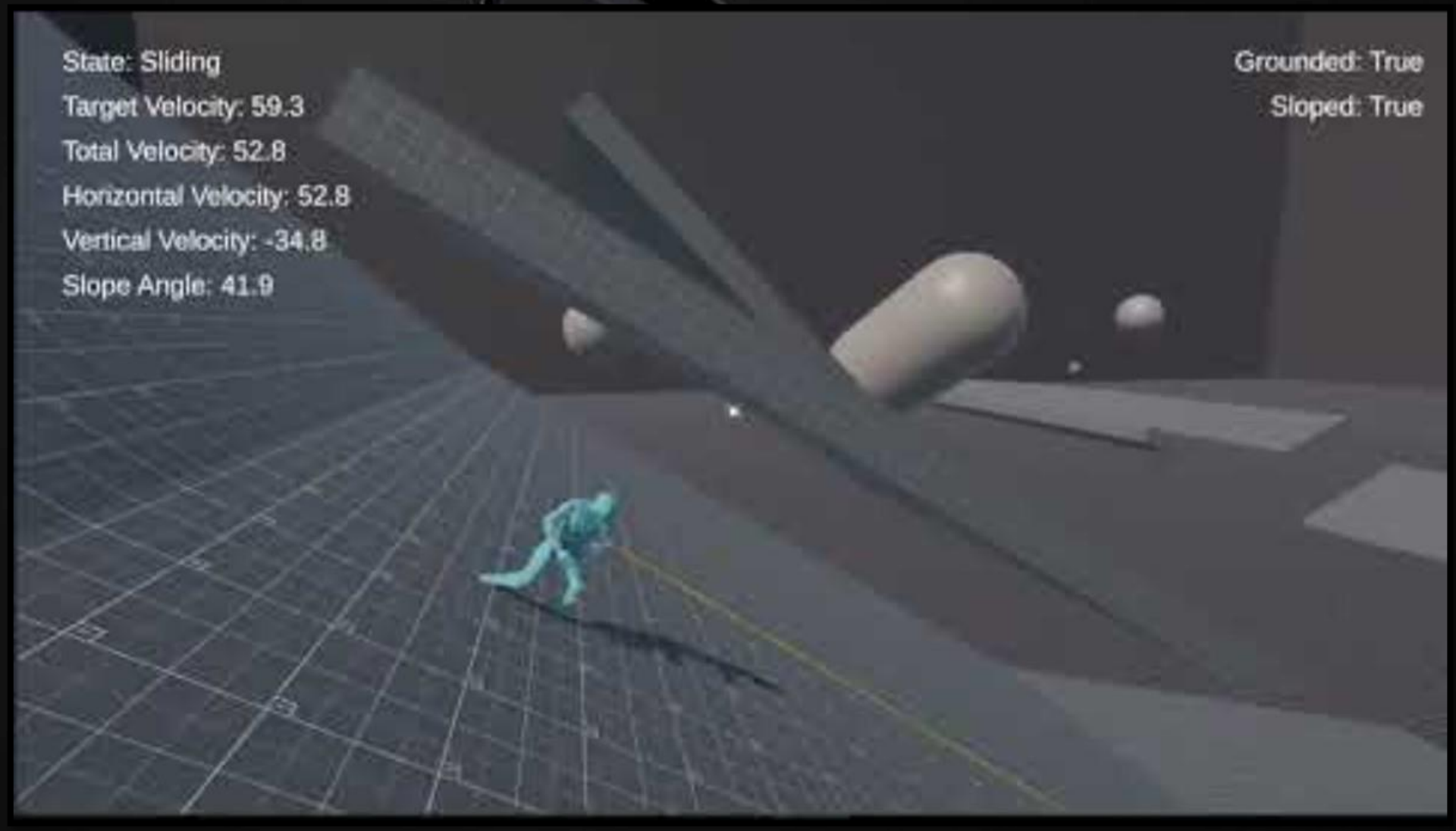






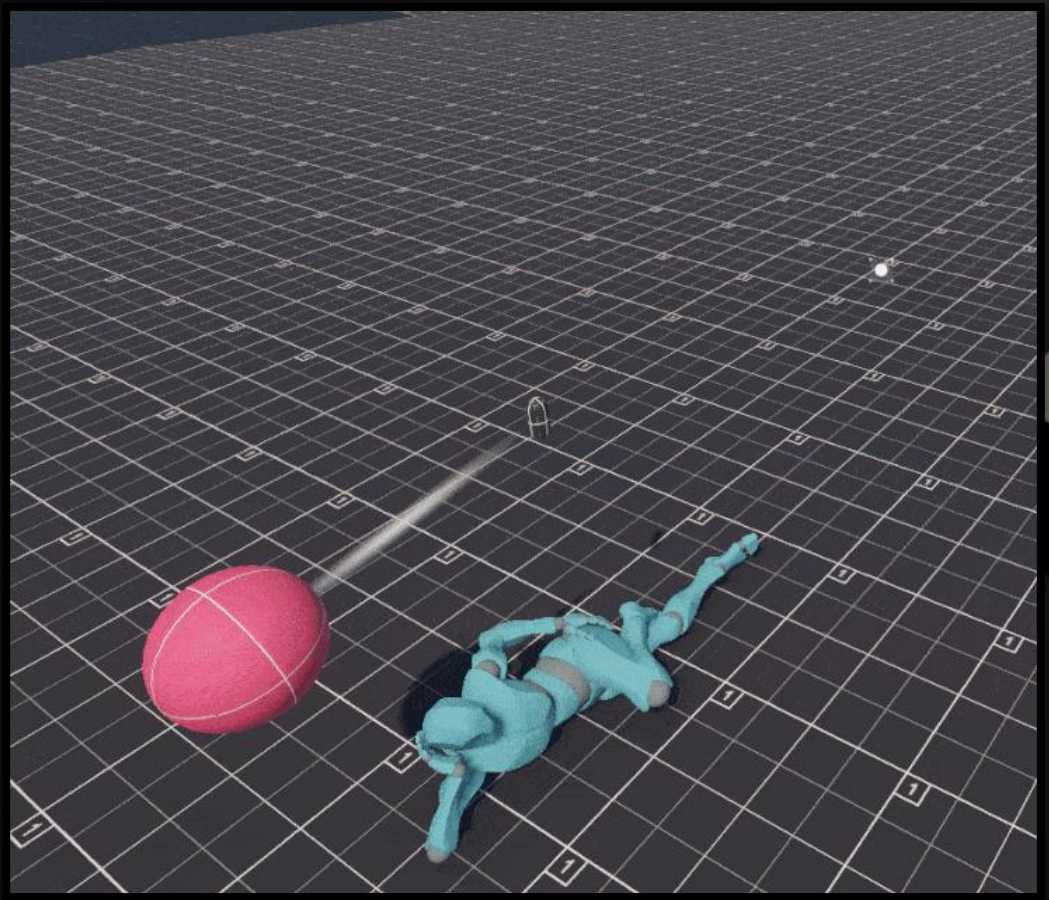
ANIMATION: STOP-MOTION

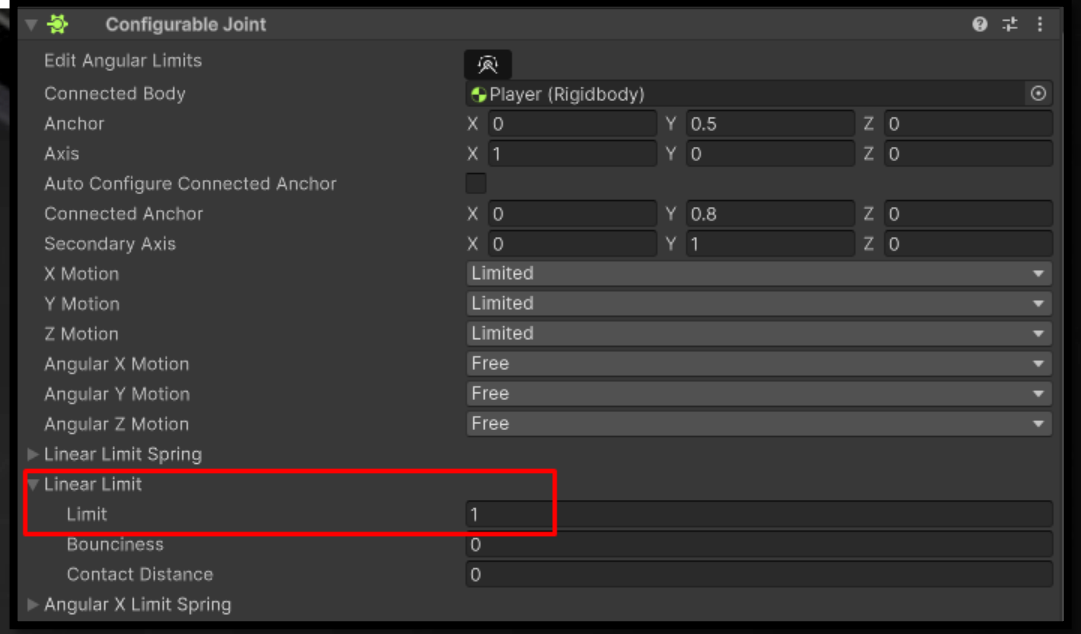
- Avoided Update()
- Used Coroutine: 0.2



WEAPON: FLAIL

- Configurable Joint – Chain Length Adjustment
- Flail State Machine





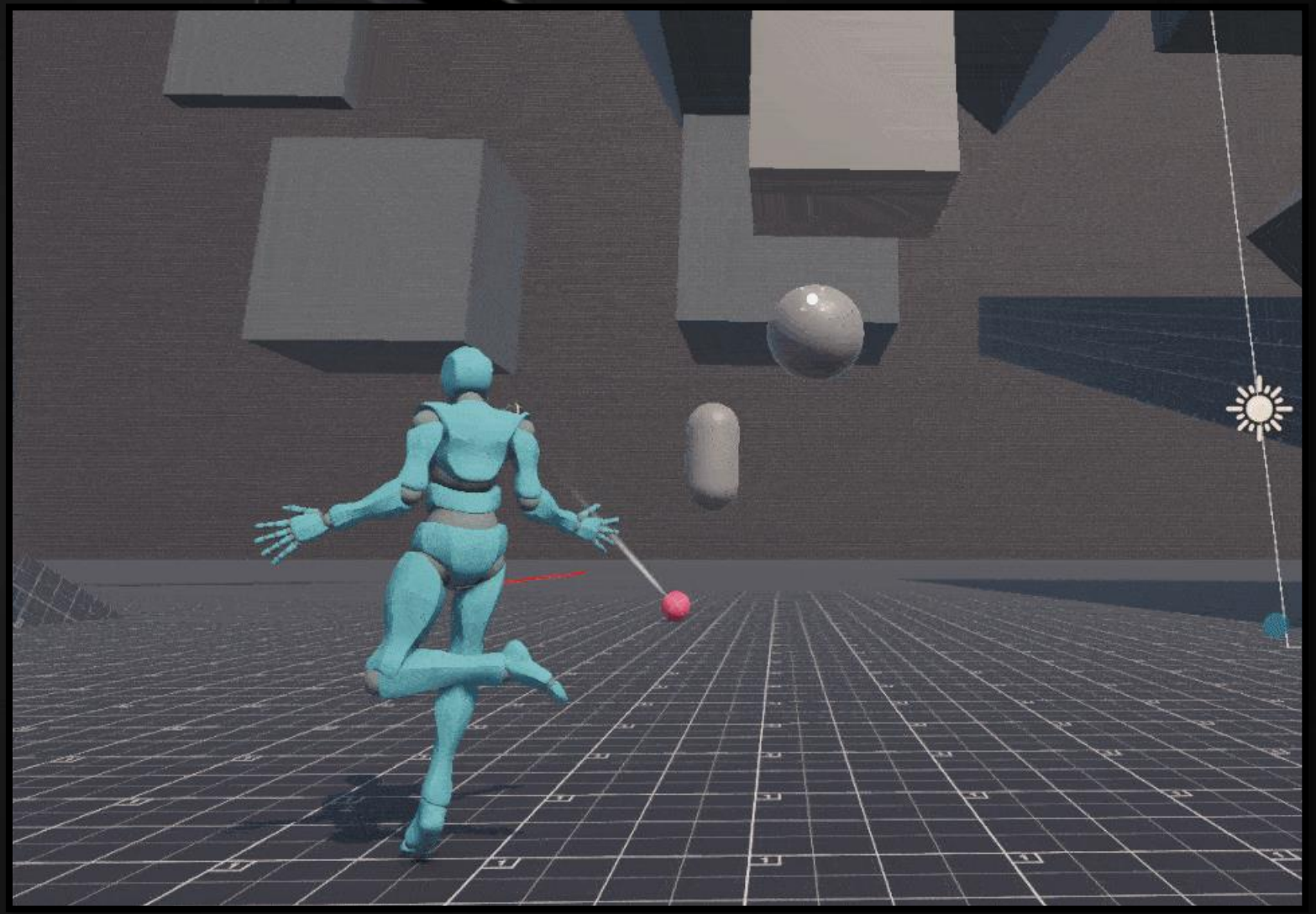

```
switch (m_currentState)
{
    case FlailState.Swinging:...

    case FlailState.Attacking:...

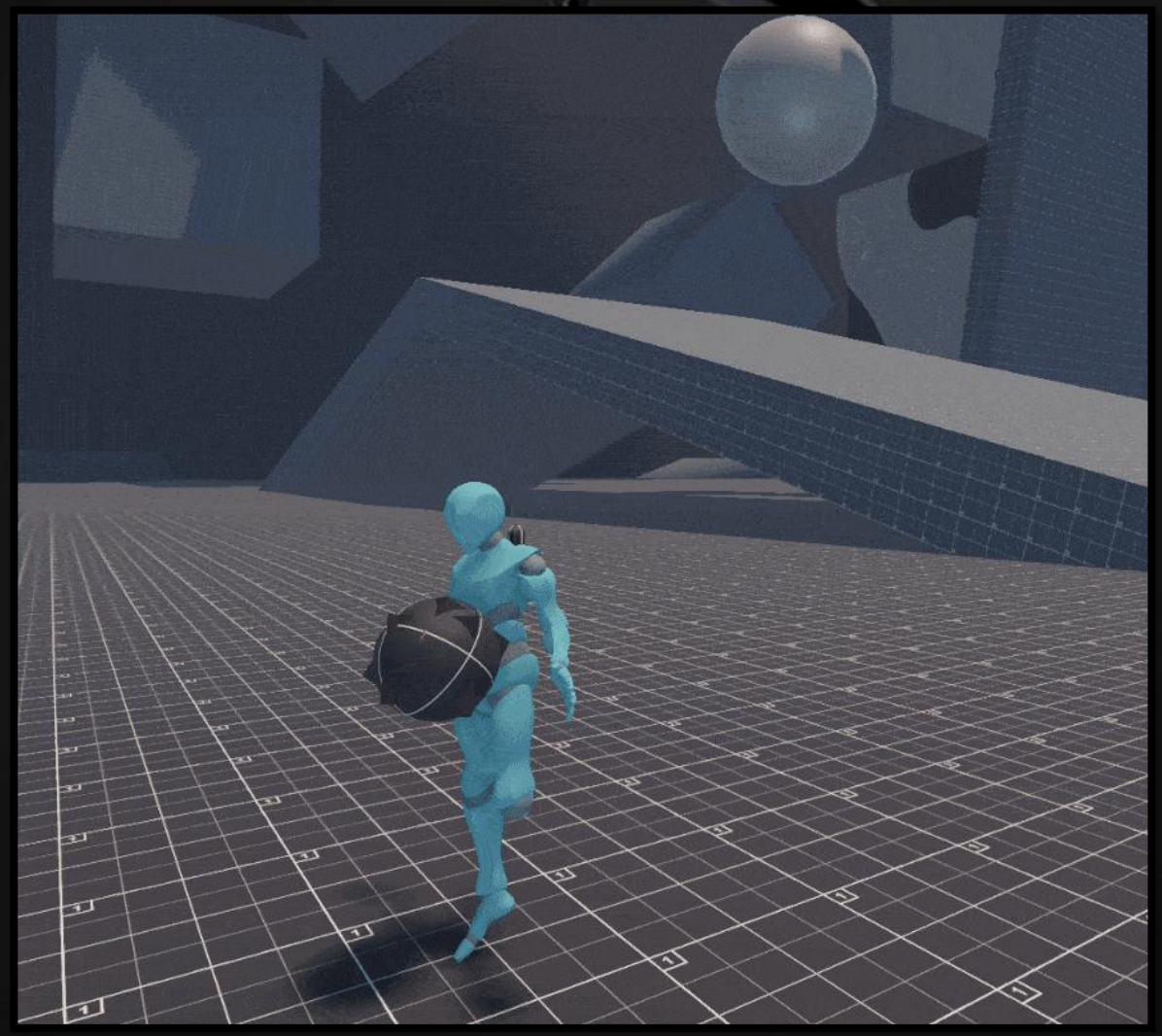
    case FlailState.Retracting:...
}
```

WEAPON: FLAIL STATES

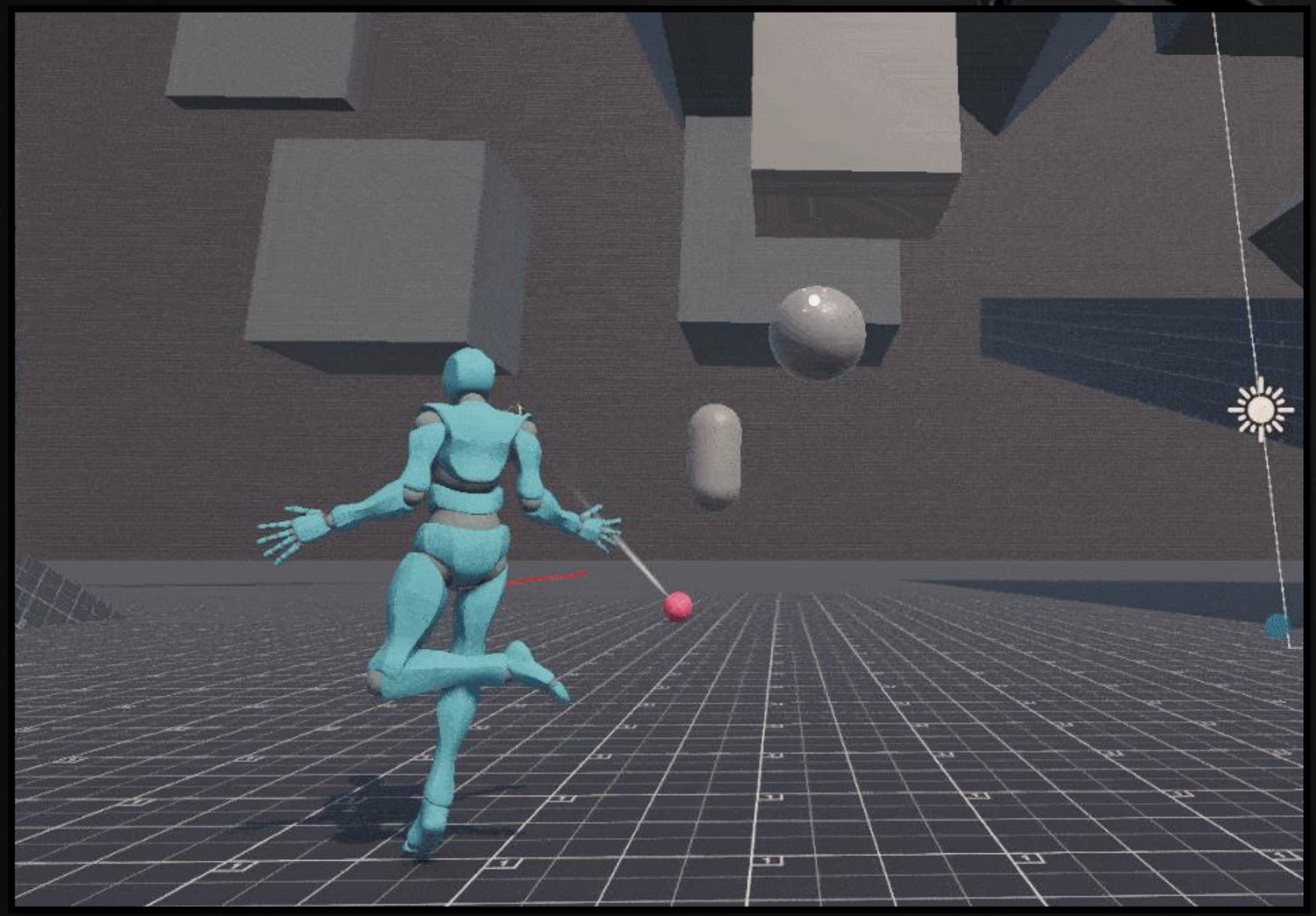
- Idle
- Swinging – Critical On/Off
- Attack/Launch – Critical On/Off
- Retract



WEAPON: FLAIL IDLE



WEAPON: FLAIL ATTACK



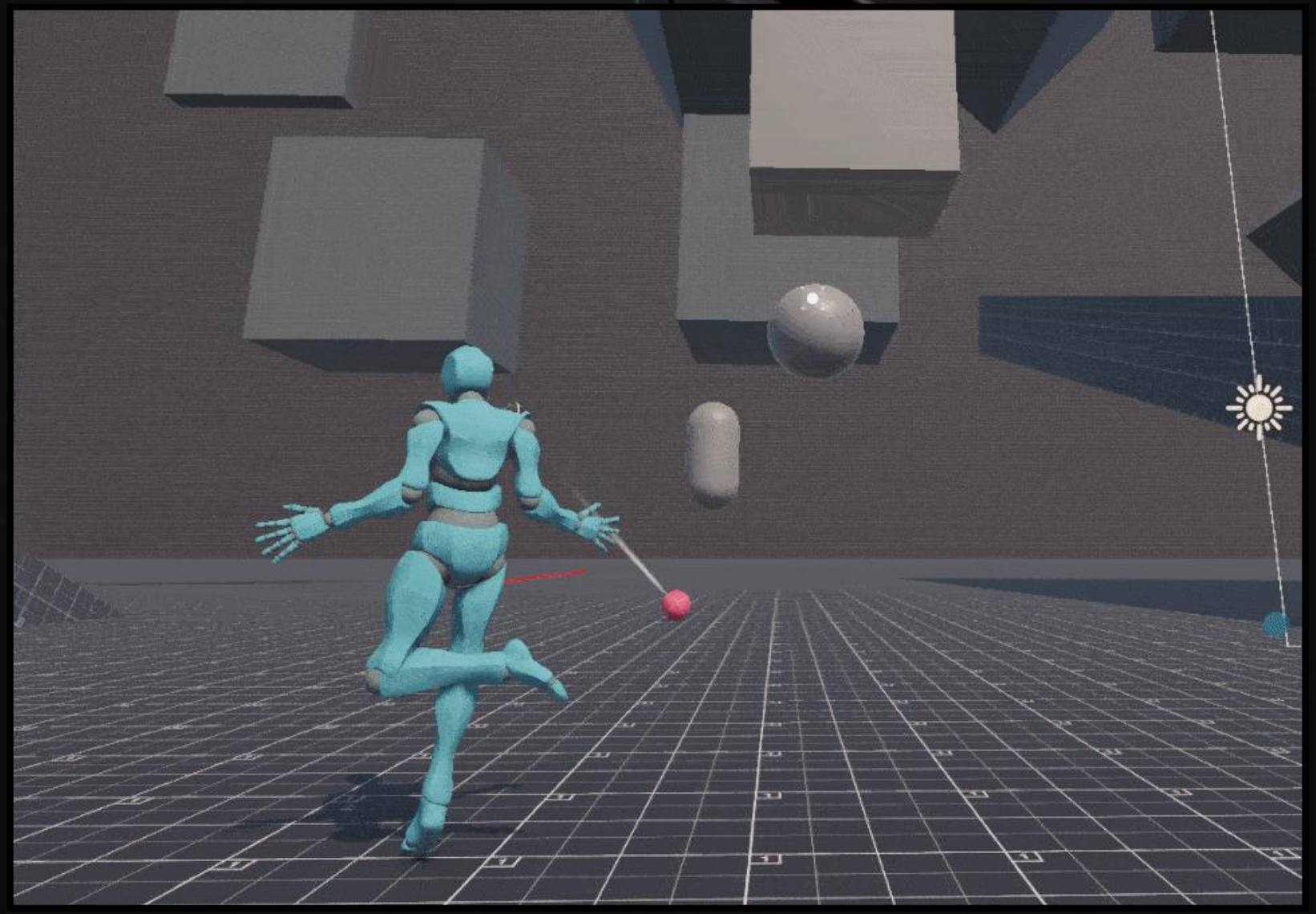
Linear Limit
- Limit: 5
- Bounciness: 0
- Contact Distance: 0

MAX CHAIN LENGTH (IDLE _SWINGING)

Linear Limit
- Limit: 15
- Bounciness: 0
- Contact Distance: 0

MAX CHAIN LENGTH (ATTACKING)

WEAPON: FLAIL RETRACT

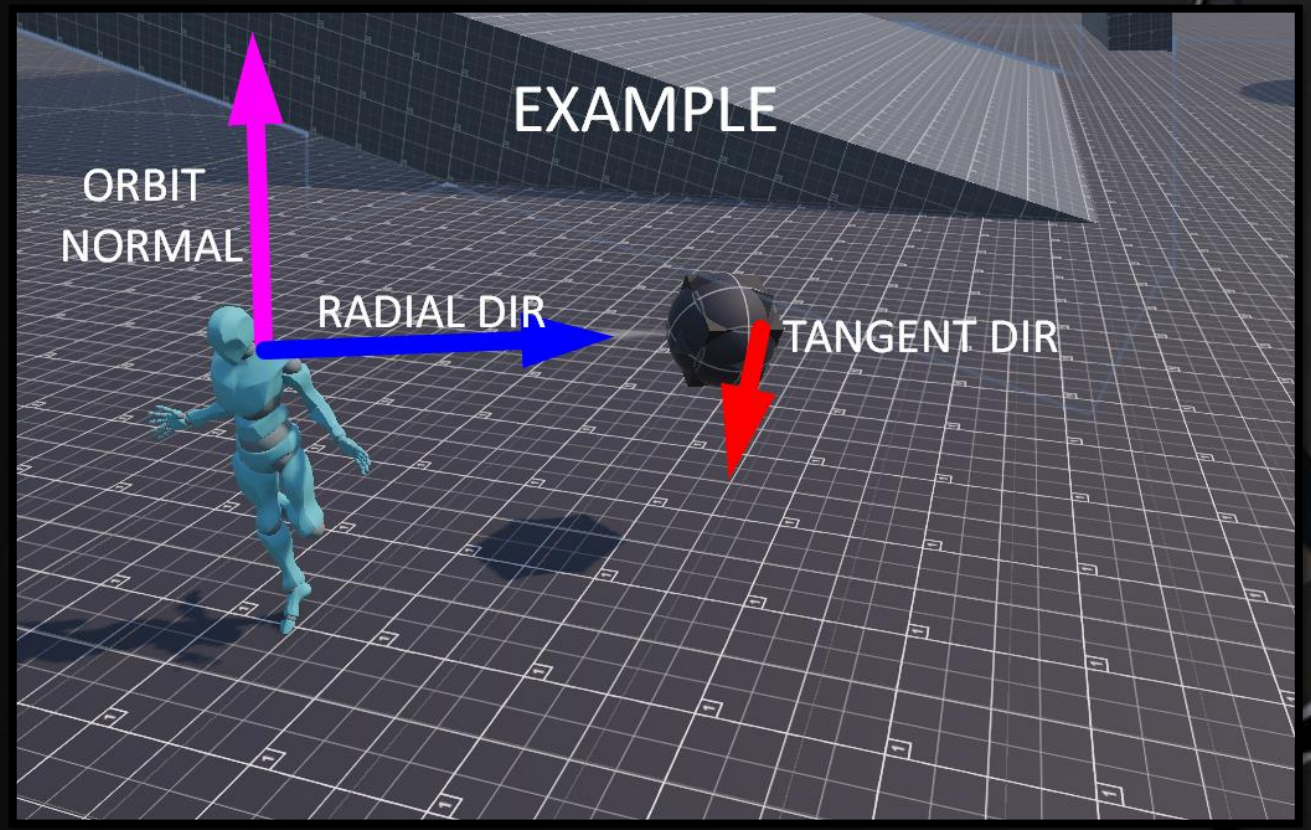


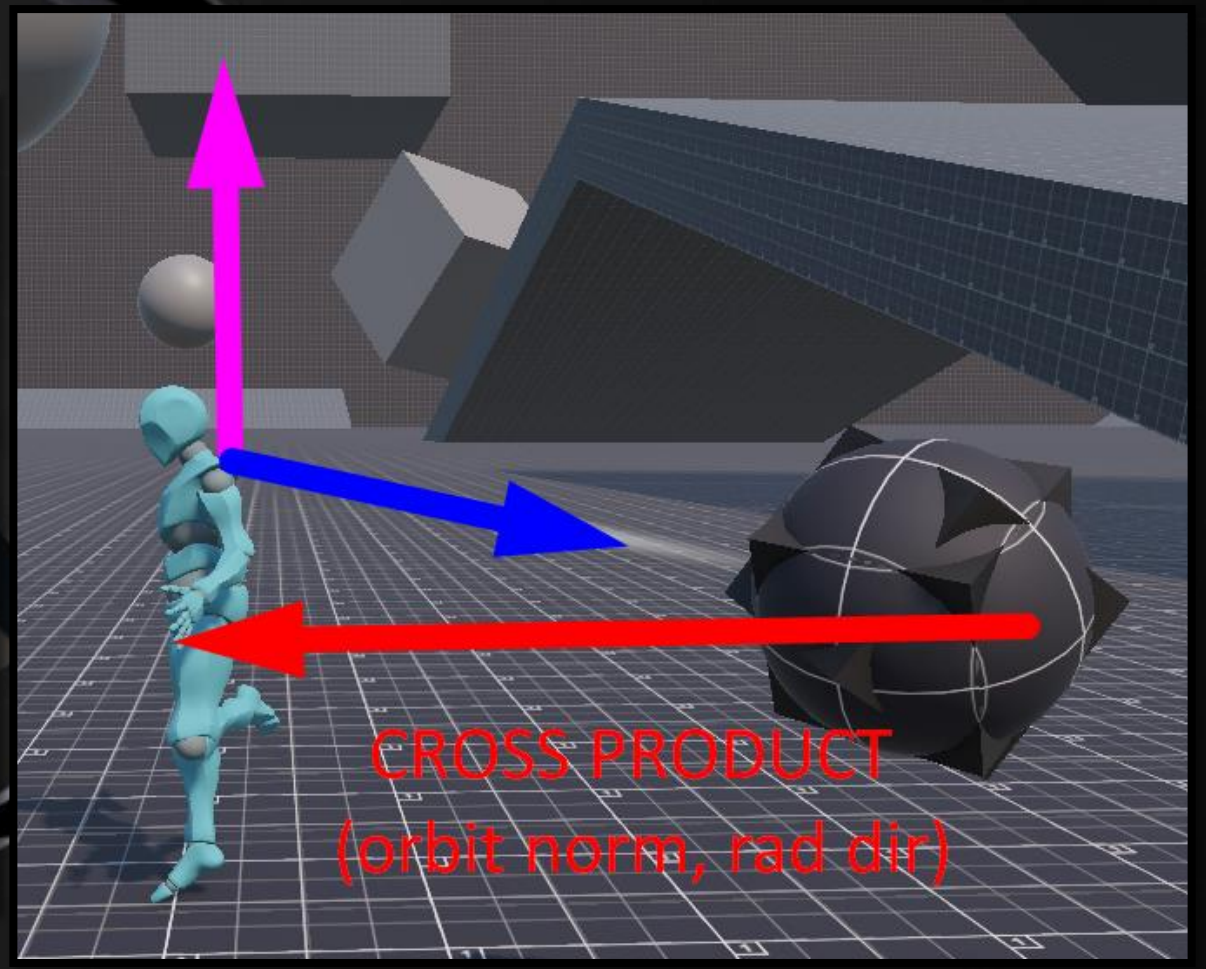
WEAPON: FLAIL SWINGING



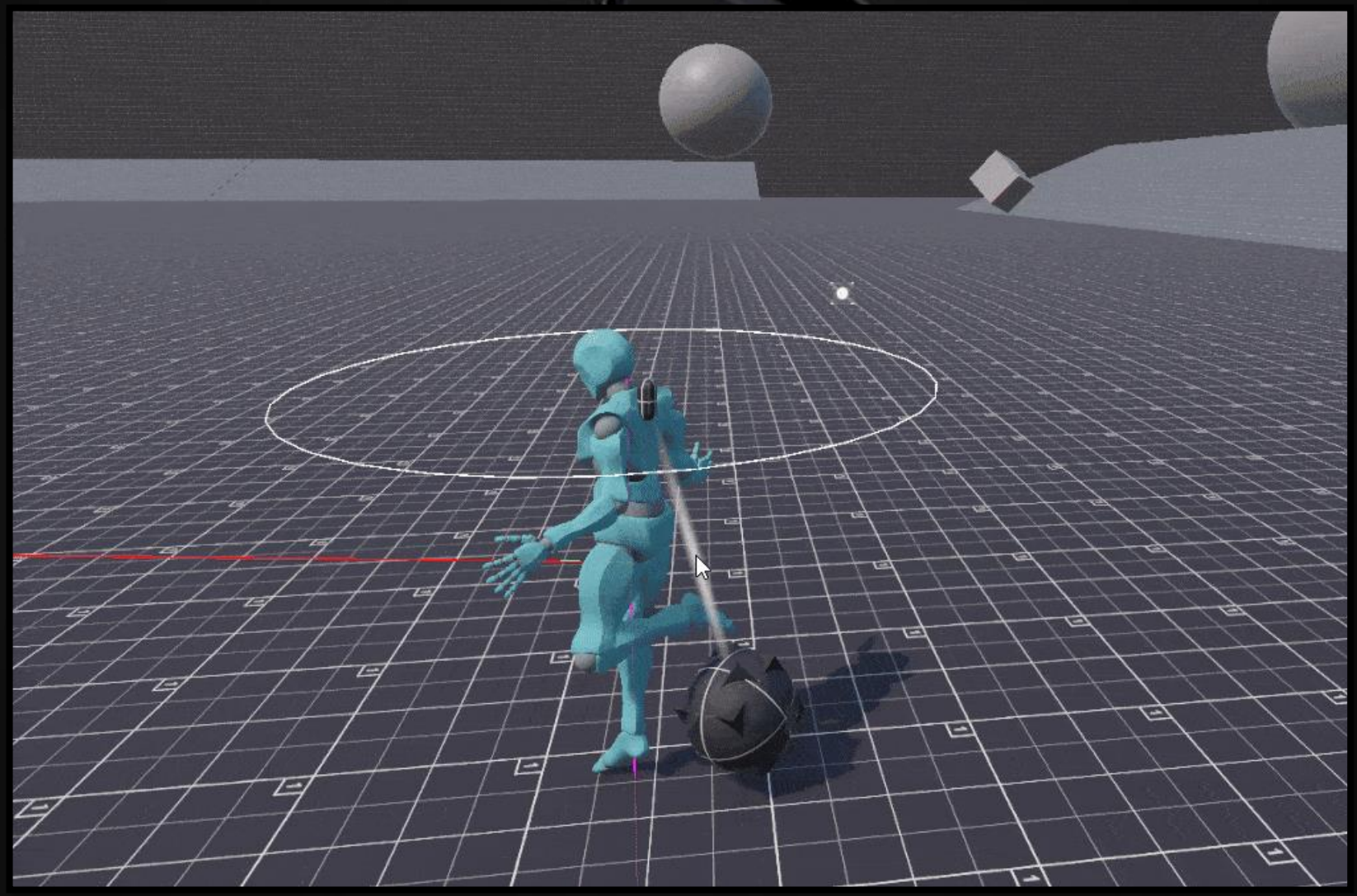


WEAPON: FLAIL SWINGING

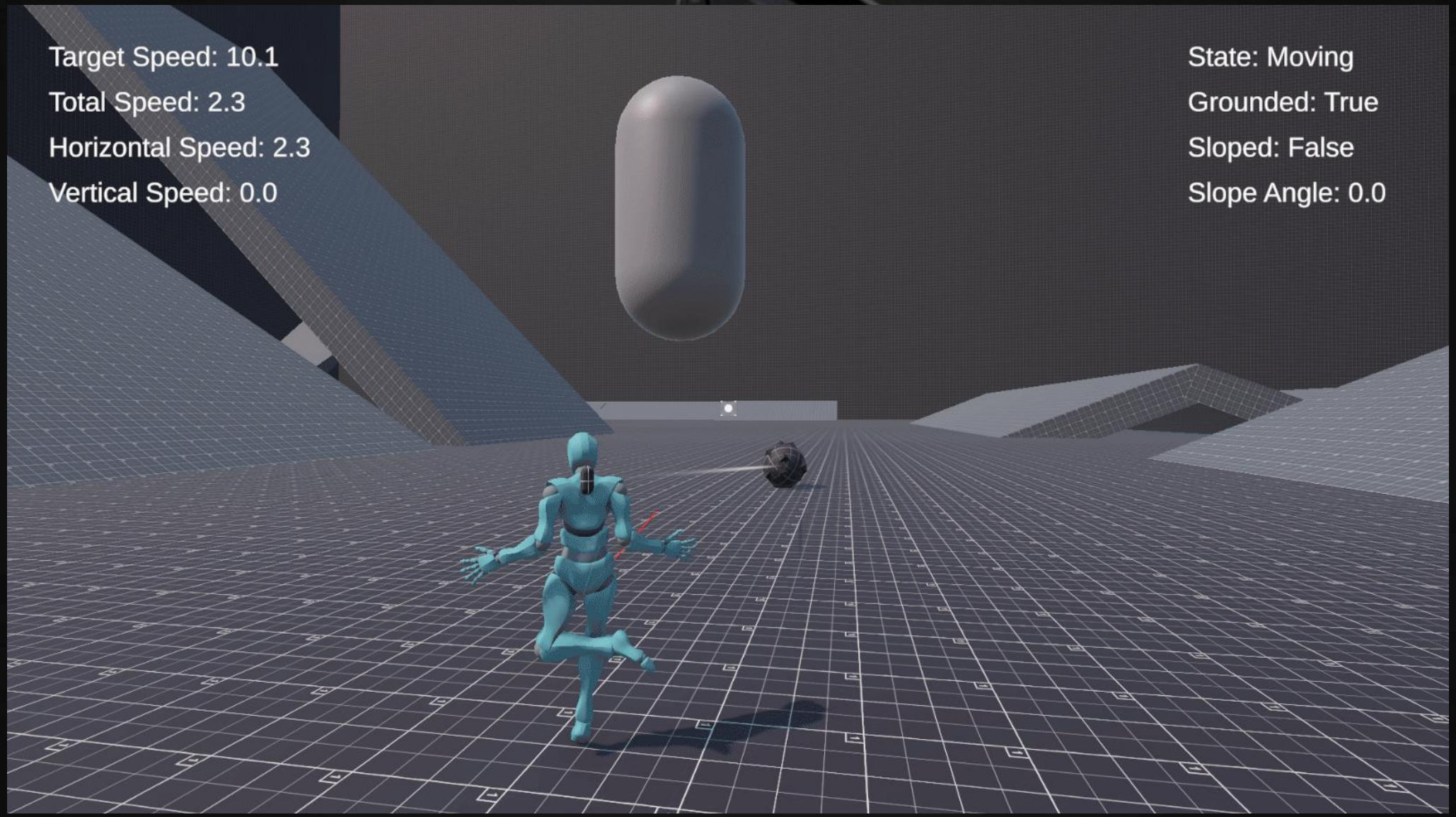




WEAPON: FLAIL SWINGING

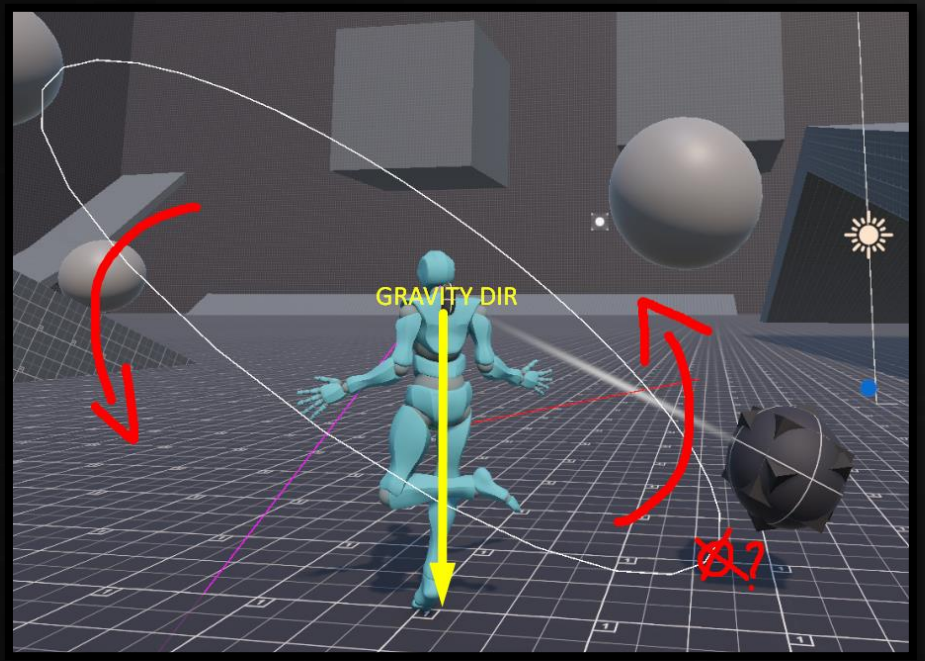


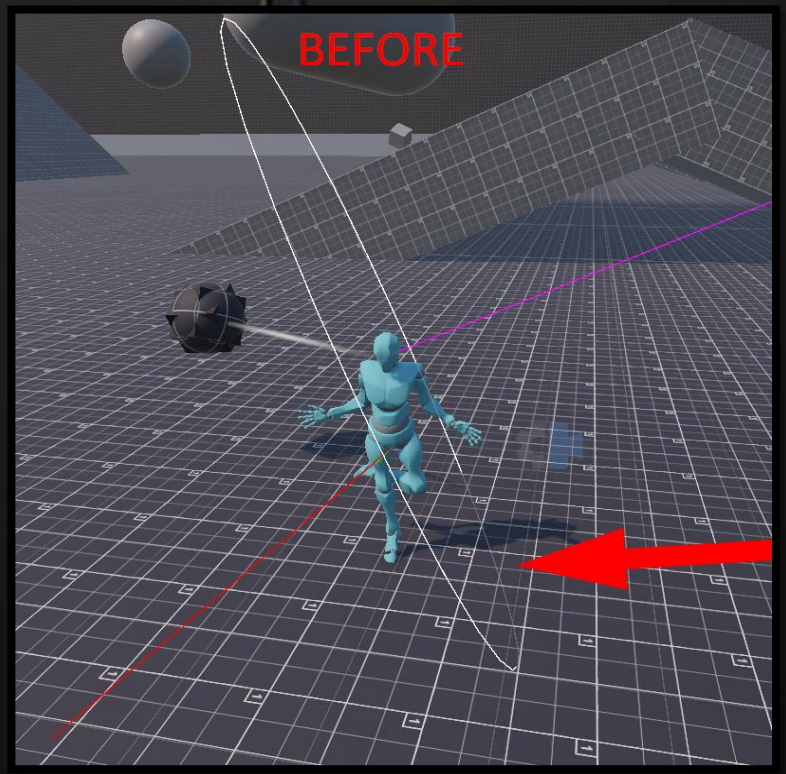
WEAPON: FLAIL CRITICAL

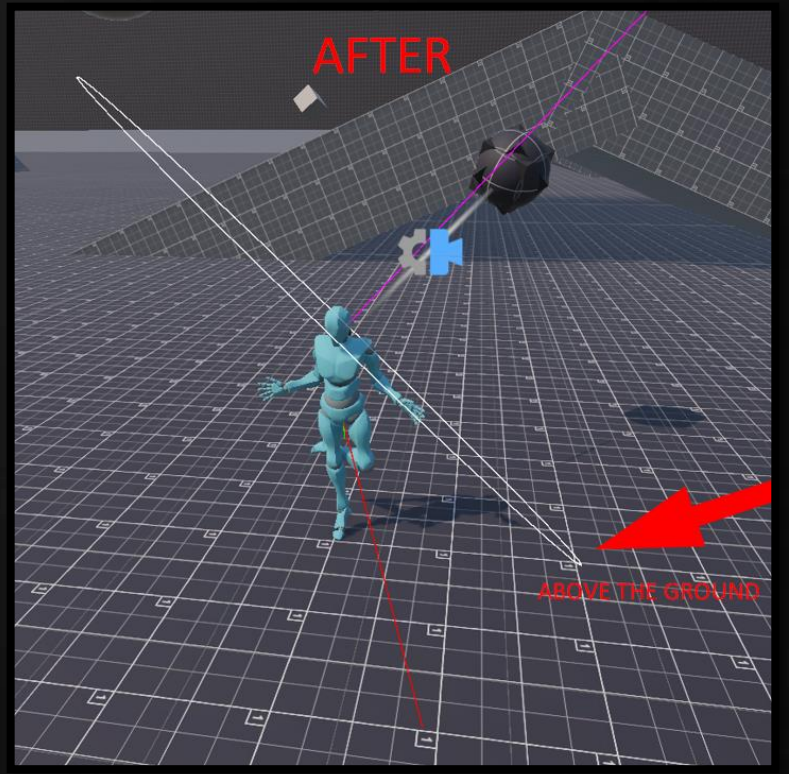


FORGIVENESS MECHANICS

- Swing Orbit Forgiveness
- Swing Power Forgiveness
- Critical Hit AOE
- Camera Zoom Out

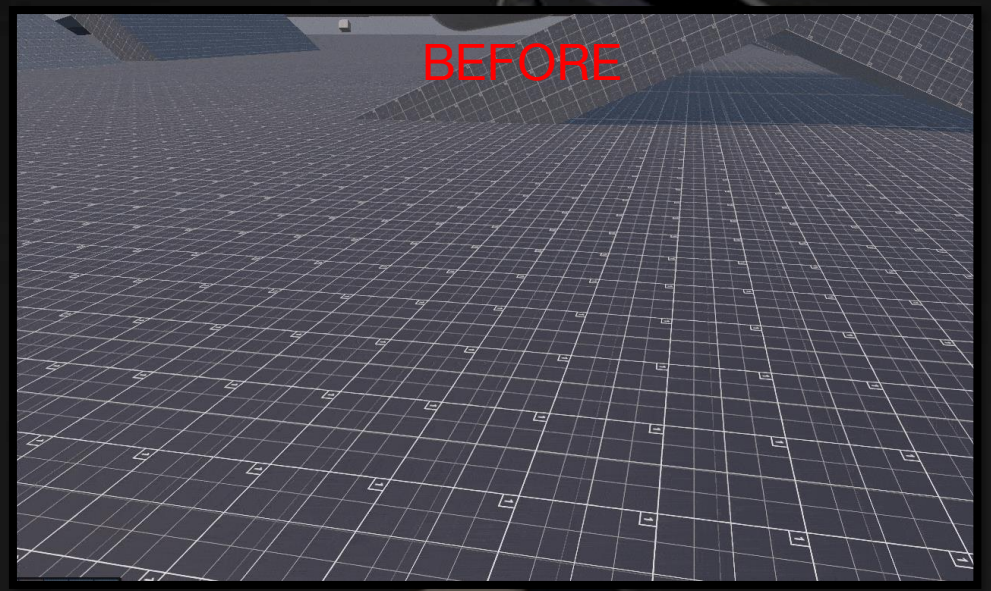


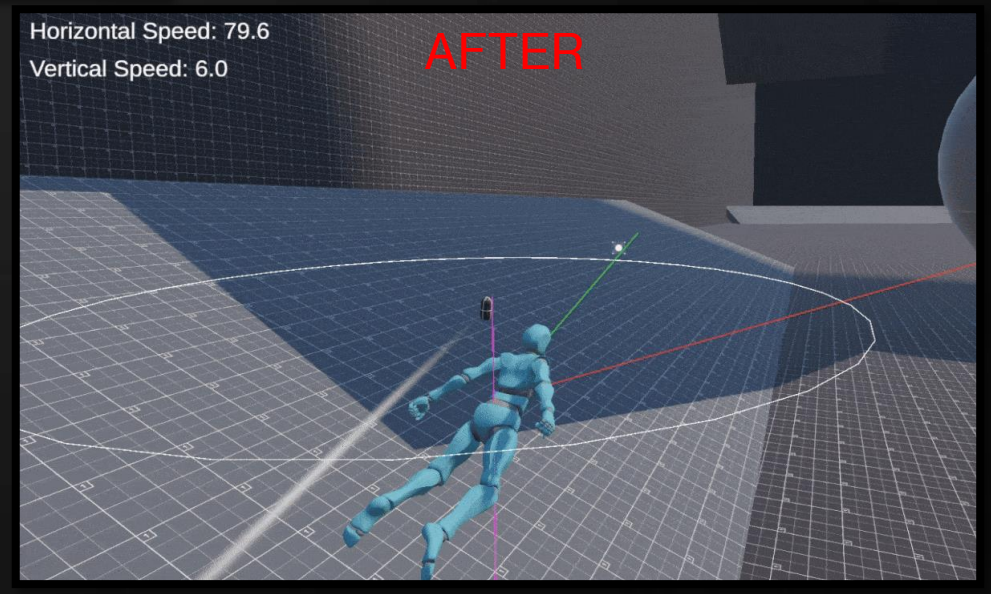




FORGIVENESS MECHANICS

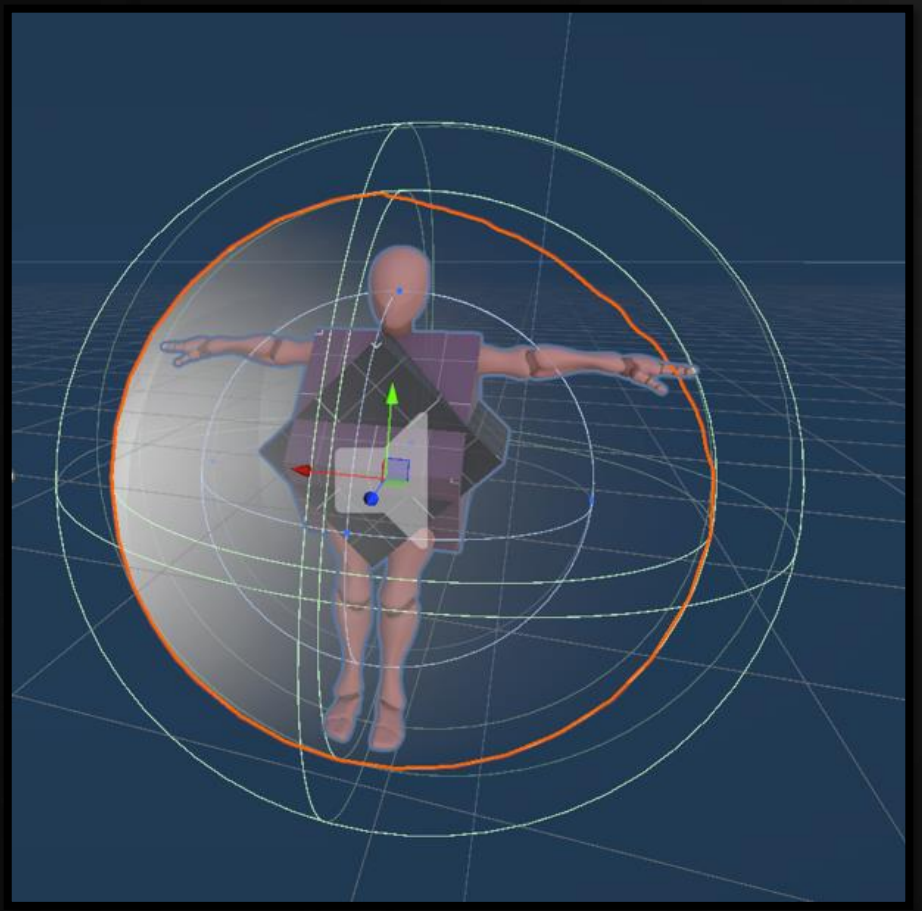
- Swing Orbit Forgiveness
- Swing Power Forgiveness
- Critical Hit AOE
- Camera Zoom Out

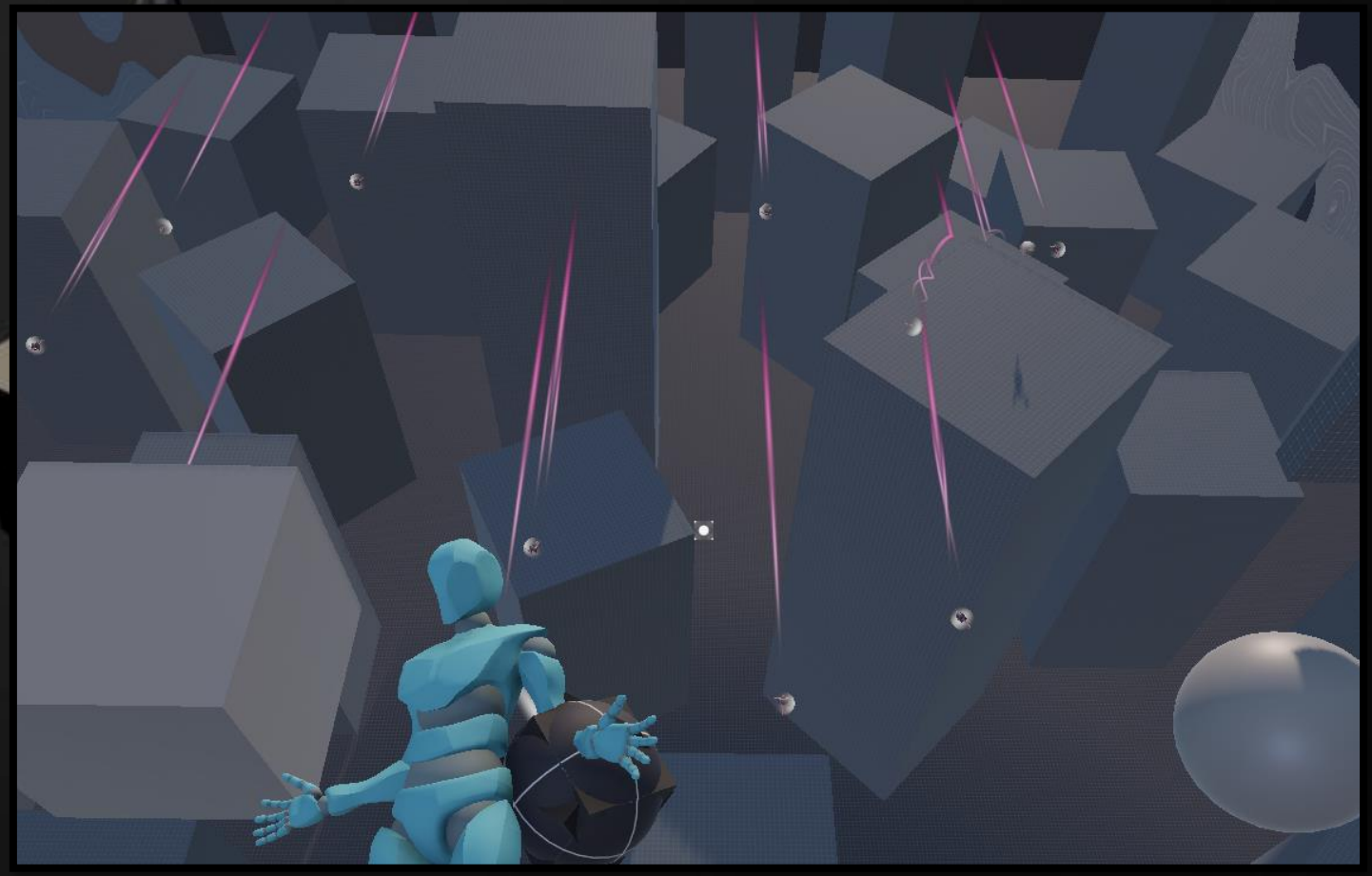




ENEMY

- Swing Orbit Forgiveness
- Swing Power Forgiveness





FUTURE WORKS

- Some ideas for GradEx



Flail retracts points?

Player body tilts towards flail

Flail: Launch based on DotProduct of aim on orbit

Cues

Props: Hit prop to turn it red and attack enemies if not players

Player: Wall Run & Wall Launches

Player: Slide on props

Player: Trail/Tron

Arena: Breakout of it and enter new cube

THE END


Target Speed: 13.5
Total Speed: 14.8
Horizontal Speed: 13.5
Vertical Speed: 6.0
Gravity Shift in: 22
State: Gliding
Grounded: False
Sloped: False
Slope Angle: 0.0
Extra Swing Power: 75.8
Flail State: Swinging