



The Audio Technology of



Guy Somberg

The Audio Technology of



Guy Somberg

PART 2

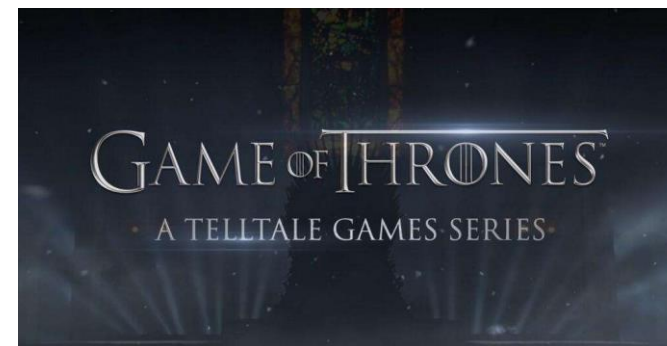
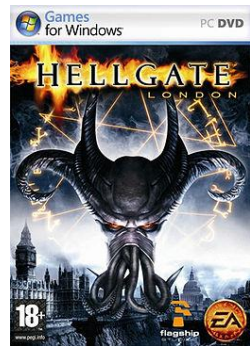
About Guy

- In games since 2002
- Owned the audio engine at (nearly) every company

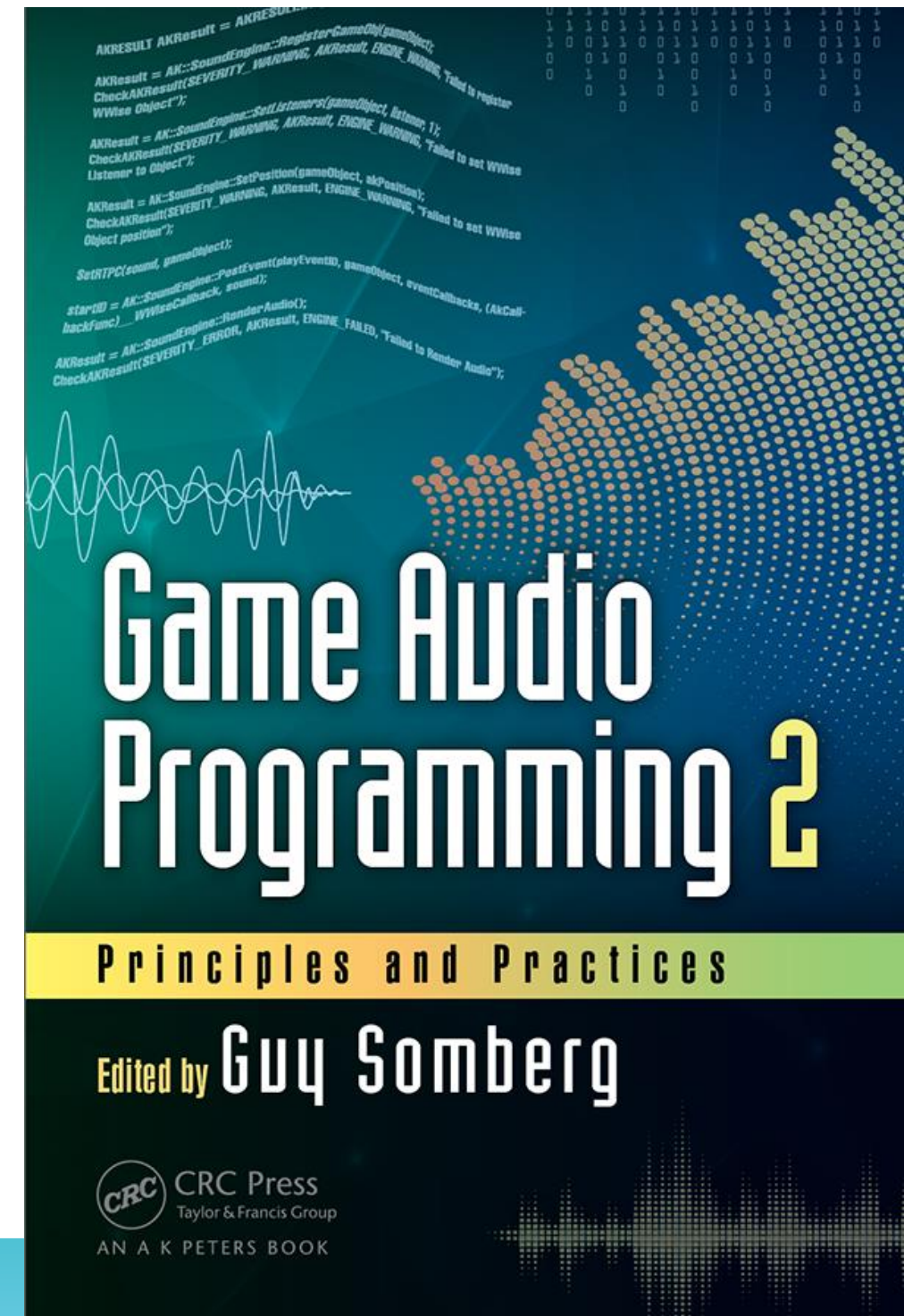
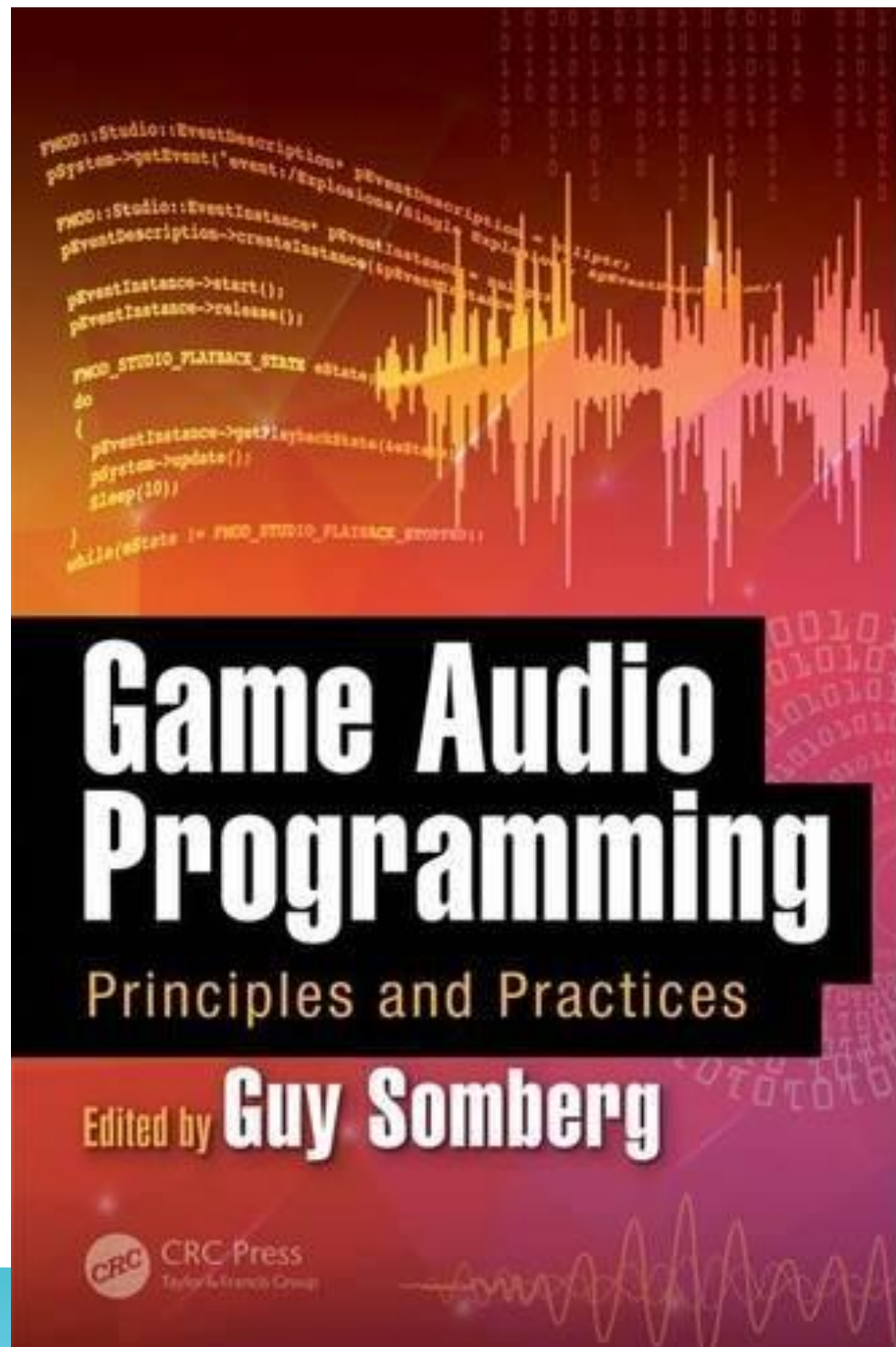


About Guy

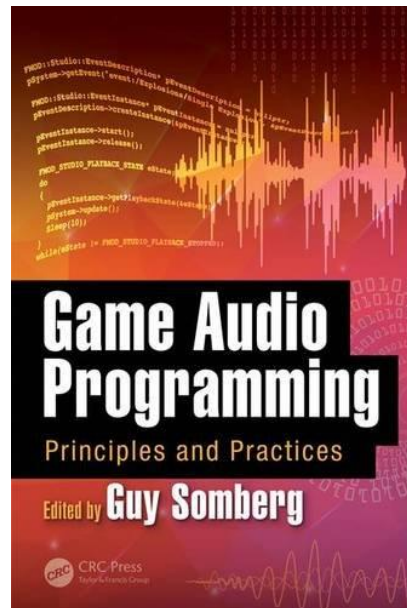
- ...and shipped lots of games



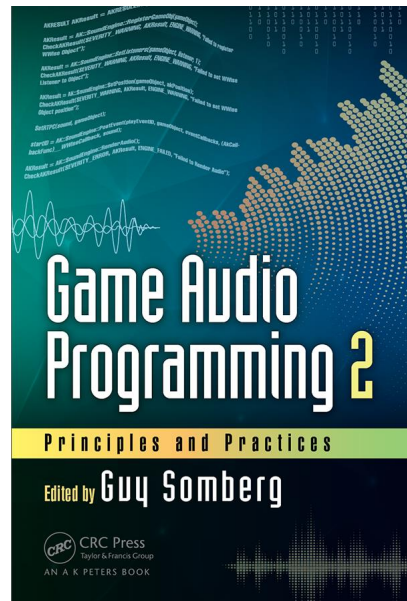
Shameless Plug



Shameless Plug



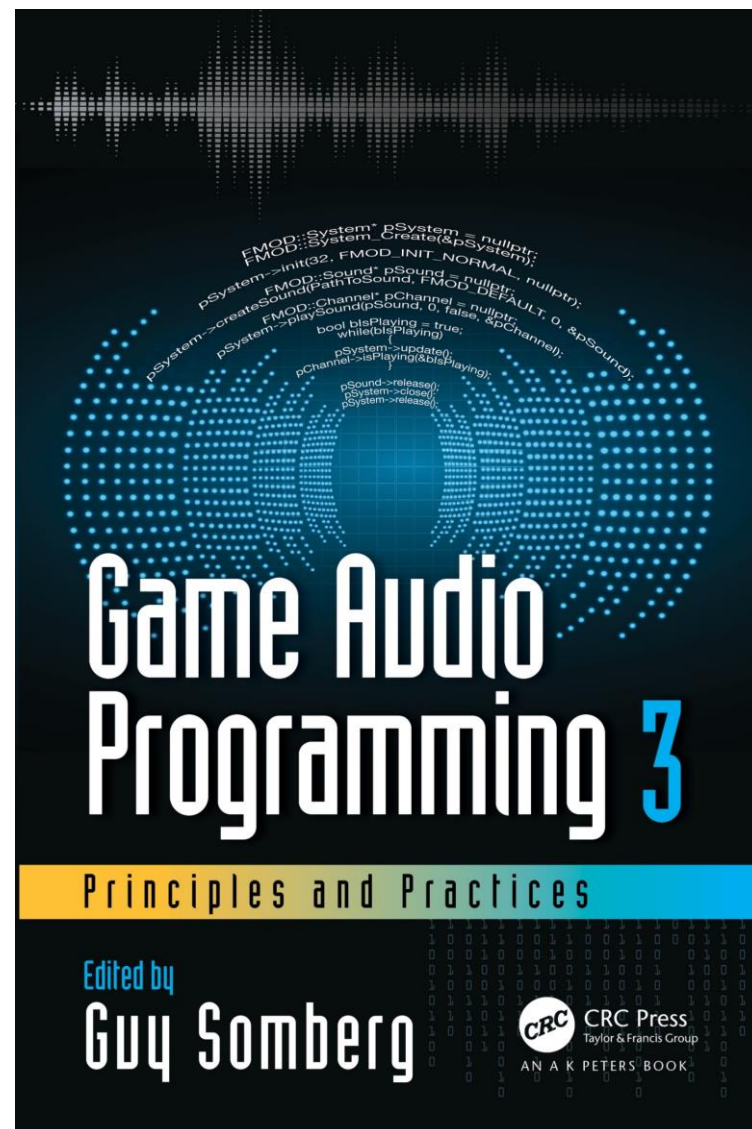
<https://www.crcpress.com/Game-Audio-Programming-Principles-and-Practices/Somberg/p/book/9781498746731>



<https://www.crcpress.com/Game-Audio-Programming-2-Principles-and-Practices/Somberg/p/book/9781138068919>

Shameless Plug

Coming Soon: Volume 3!



<https://www.routledge.com/Game-Audio-Programming-3-Principles-and-Practices/Somberg/p/book/9780367348045>



TORCHLIGHT FRONTIERS

ARPGs are Hard

- Everything is happening on the screen in front of you
 - See also: RTS, Adventure Games, etc.
- Chaotic action
 - See also: FPS, MOBA, etc.
- Randomized level layout
 - See also: Roguelikes, Strategy, etc.

See GDC2020

- Importance-Based Mixing
- Volumetric Sounds
- Screen-Space Distance Attenuation



RedOcto OGIN0

0.88

0.84

0.5

0.65

2.13

2.27

0.64

1.99

1.5

1.7

3.0

1.53

1.58

1.72

1.63

1.54

Importance Bucket Effects

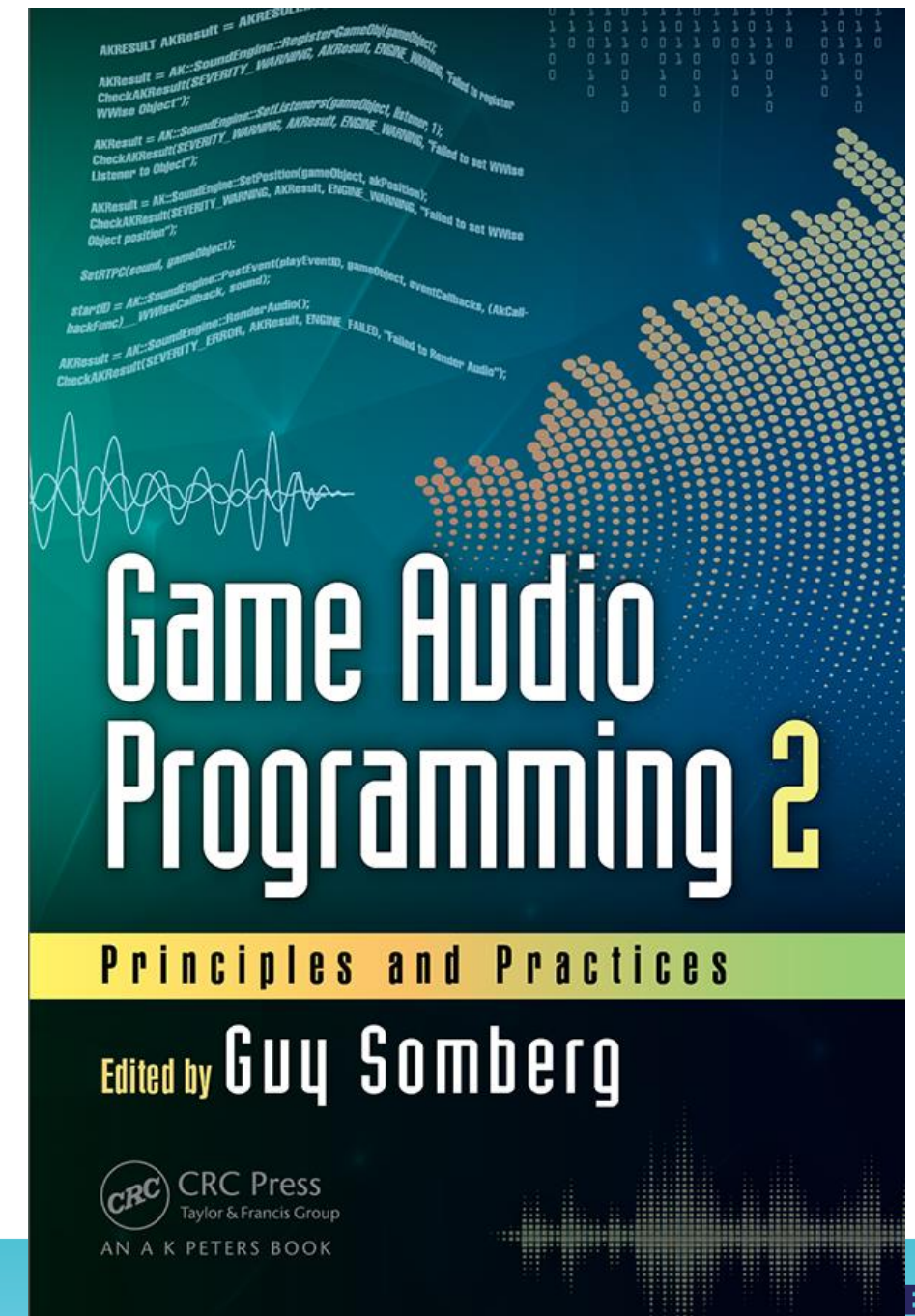
- Volume, Peaking Filter, High Shelf Filter
 - Implementation detail: Peaking and High Shelf Filters implemented using Multiband EQ

Priority	Effect
1	Peaking Filter
2	No change
3	Volume Reduction
4	Volume Reduction, High Shelf
5	Volume Reduction, High Shelf



Working Solution

- Game Audio Programming Principles and Practices Volume 2
- Chapter 12: "Approximate Position of Ambient Sounds of Multiple Sources" by Nic Taylor
- Thanks, Nic!



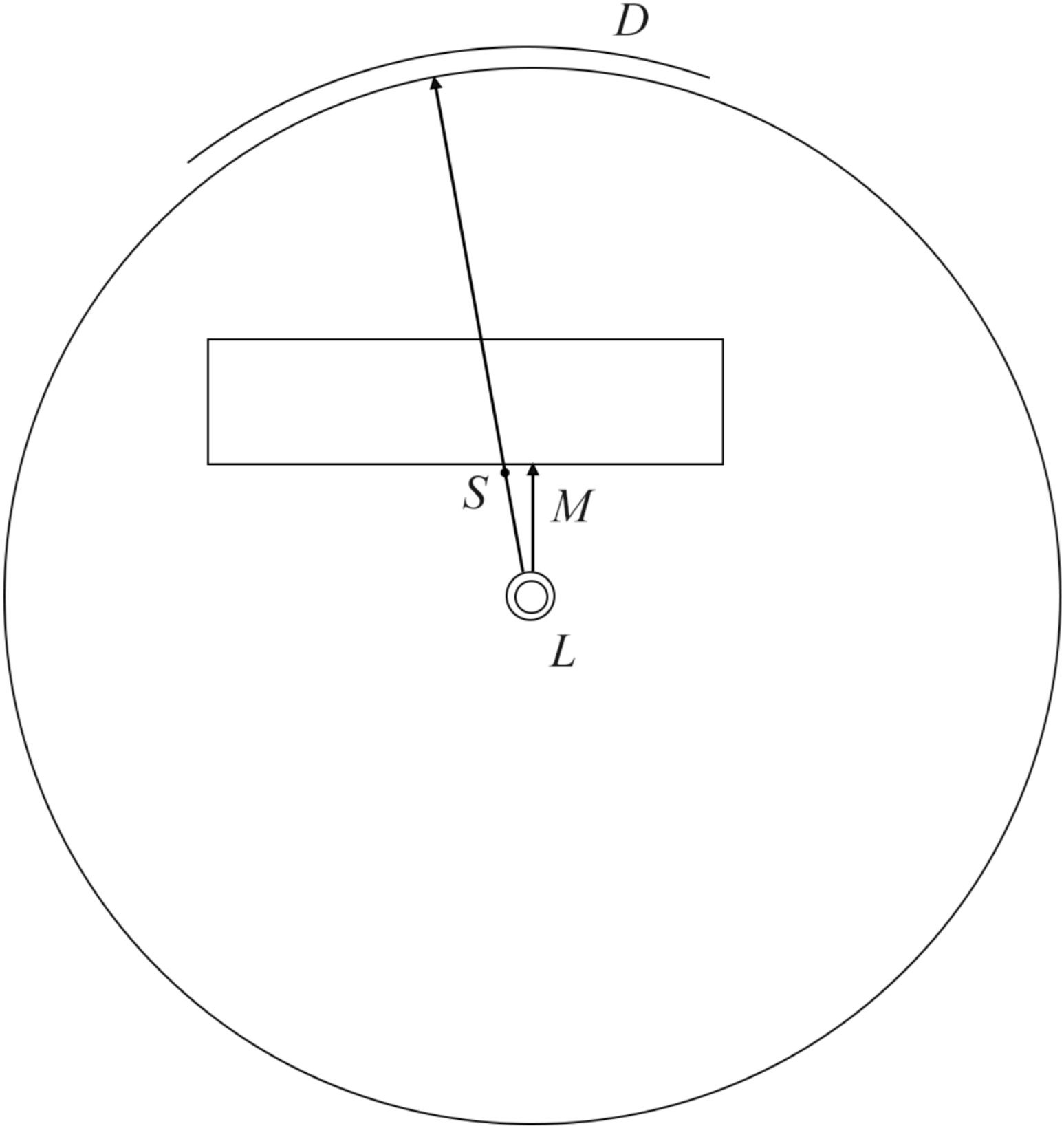
More Math!

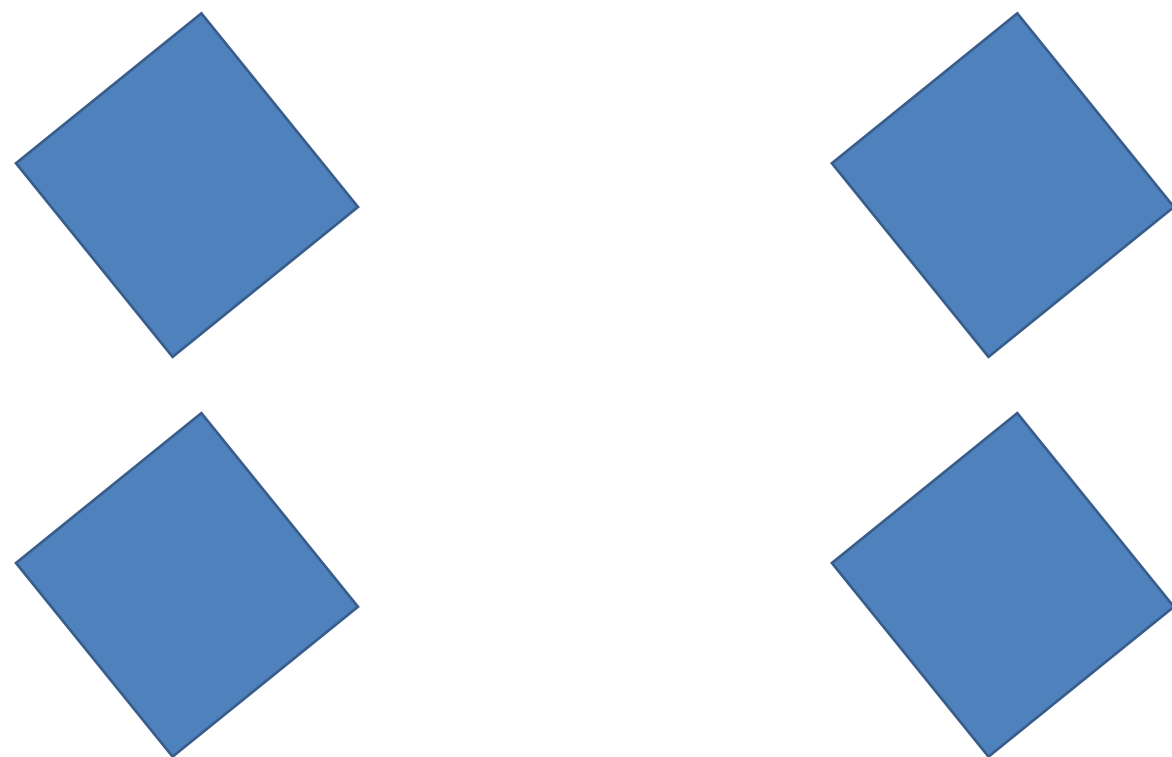
$$\hat{\sigma} = \lim_{\Delta s_i \rightarrow 0} \sum_{i=1}^n \frac{\hat{v}}{\|\hat{v}\|} W(\hat{v}) \Delta s_i = \int_C \frac{\hat{v}}{\|\hat{v}\|} W(\hat{v}) \Delta s$$

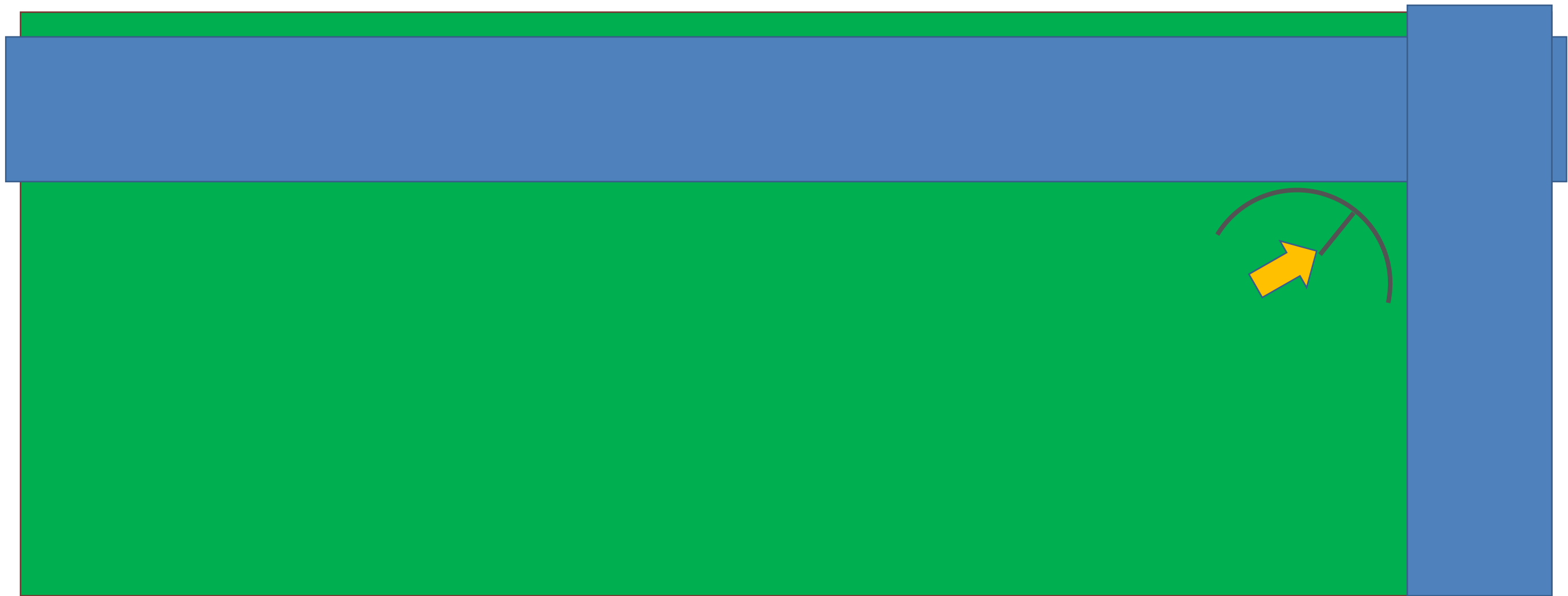
Becomes:

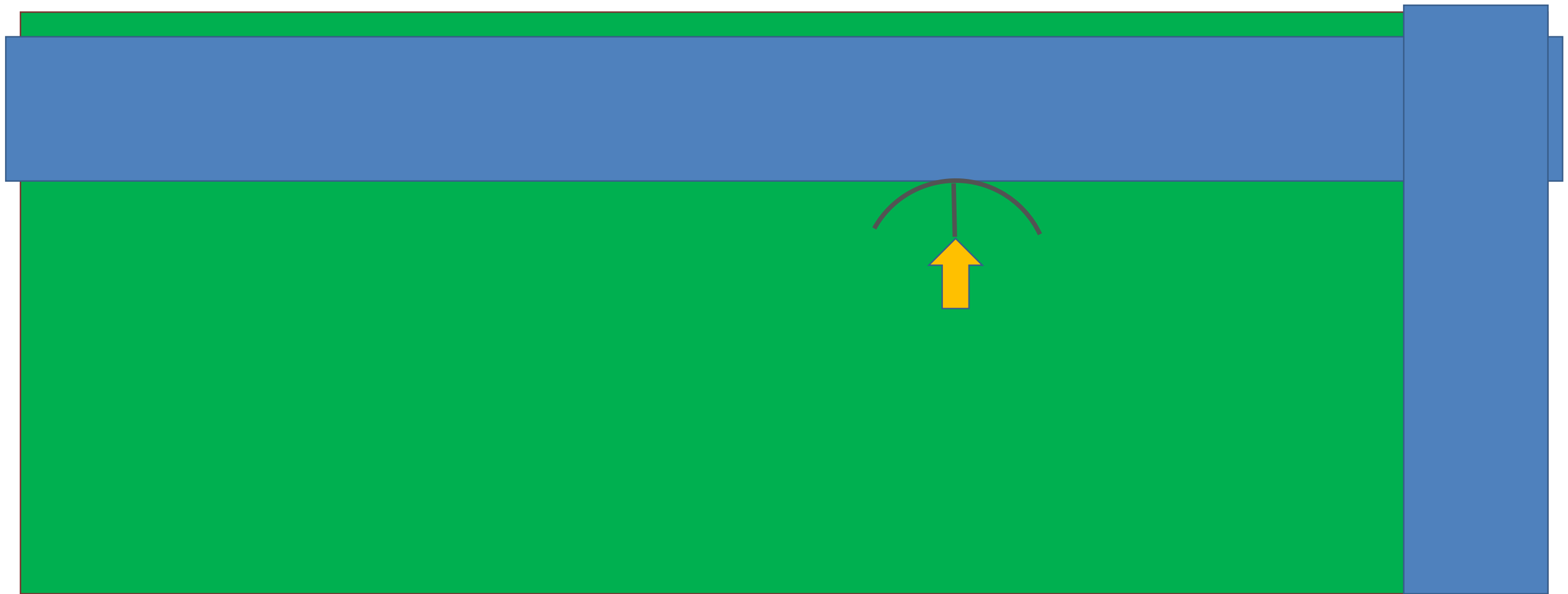
$$\hat{\sigma} = \iint_R \frac{\langle x, y \rangle}{\sqrt{x^2 + y^2}} \left(1 - \sqrt{x^2 + y^2} \right) \Delta x \Delta y$$

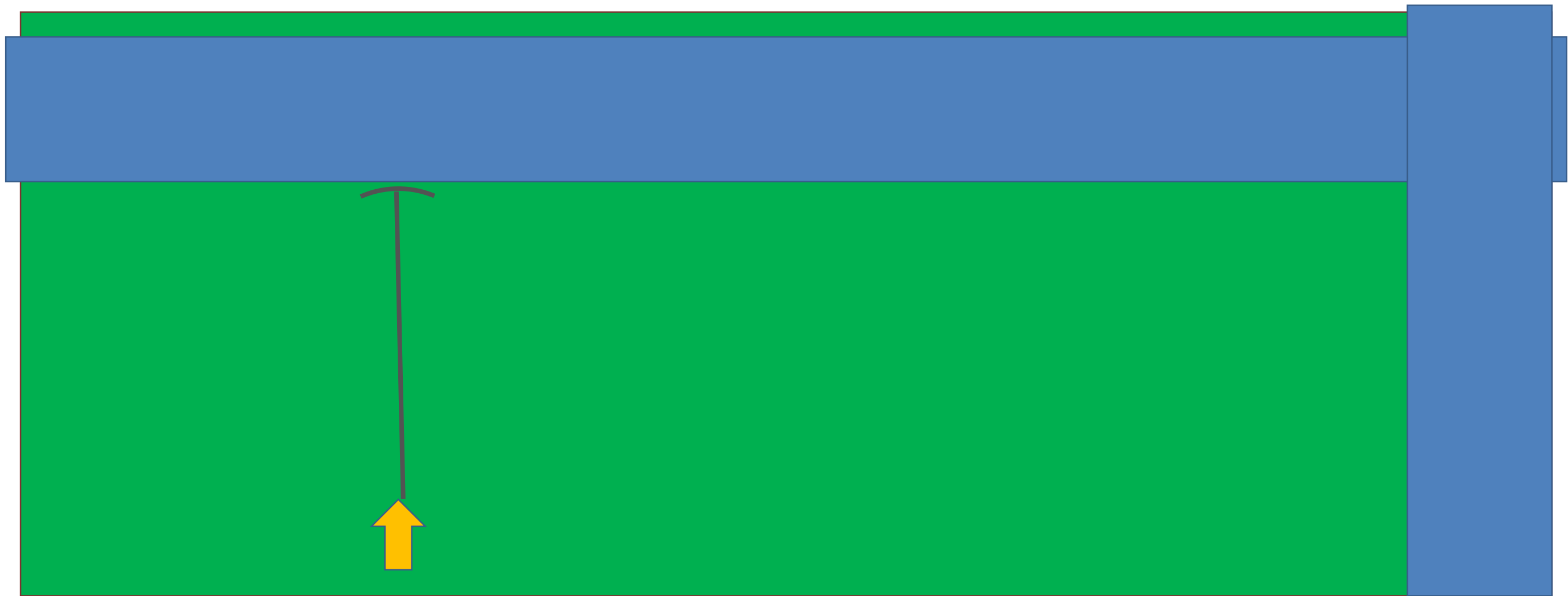
Image Credit: Nic Taylor







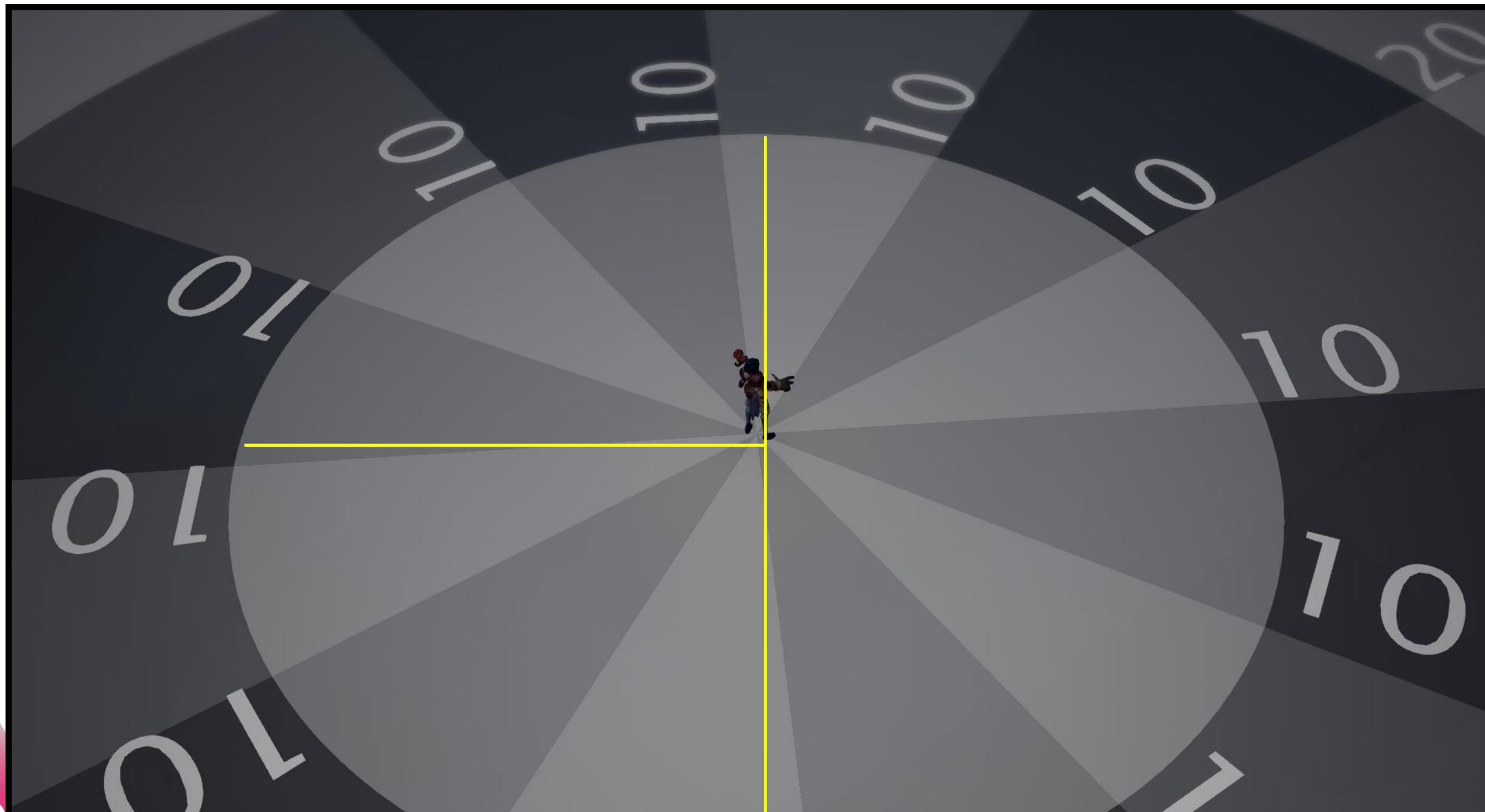


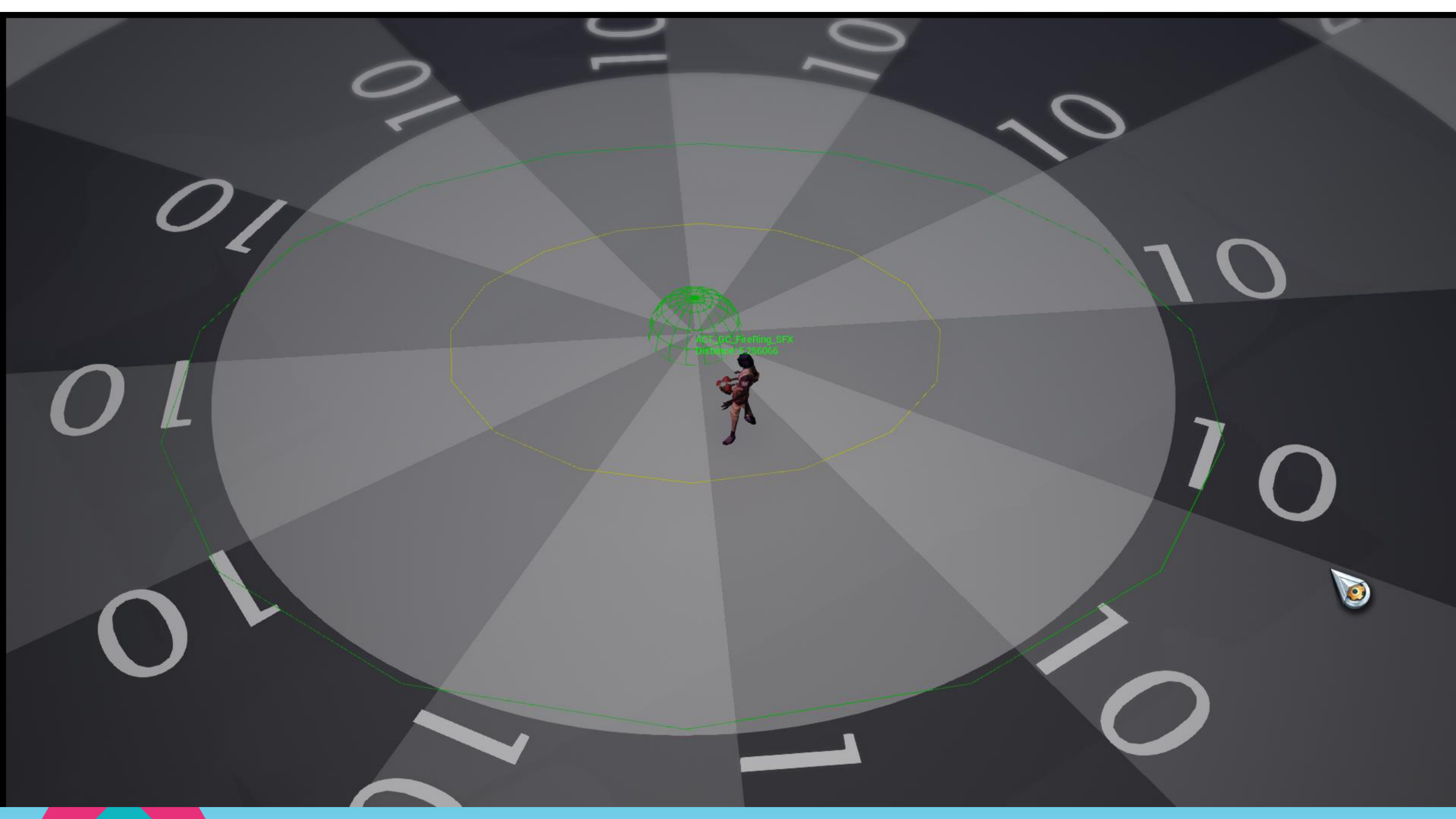


The Problem

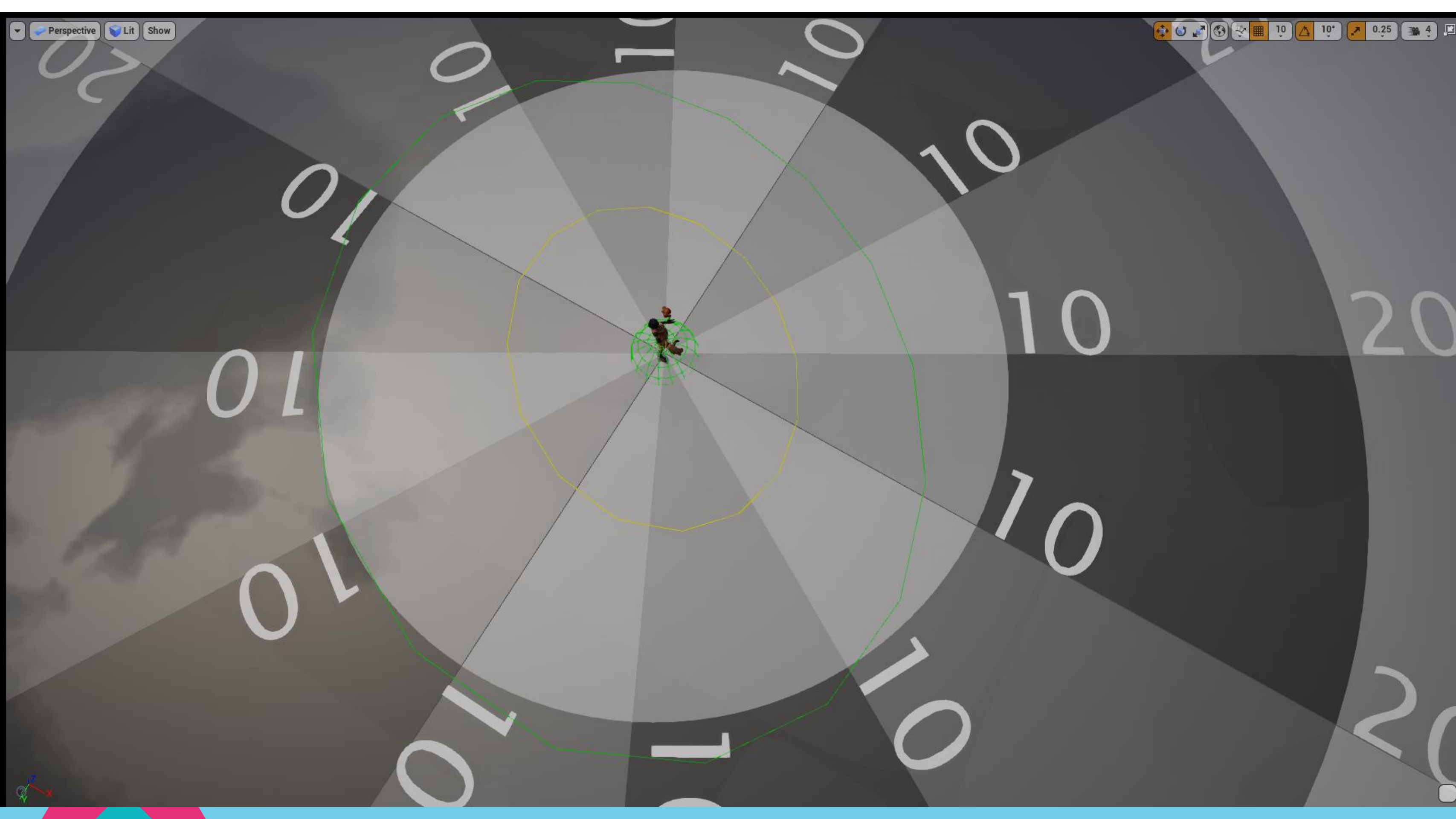


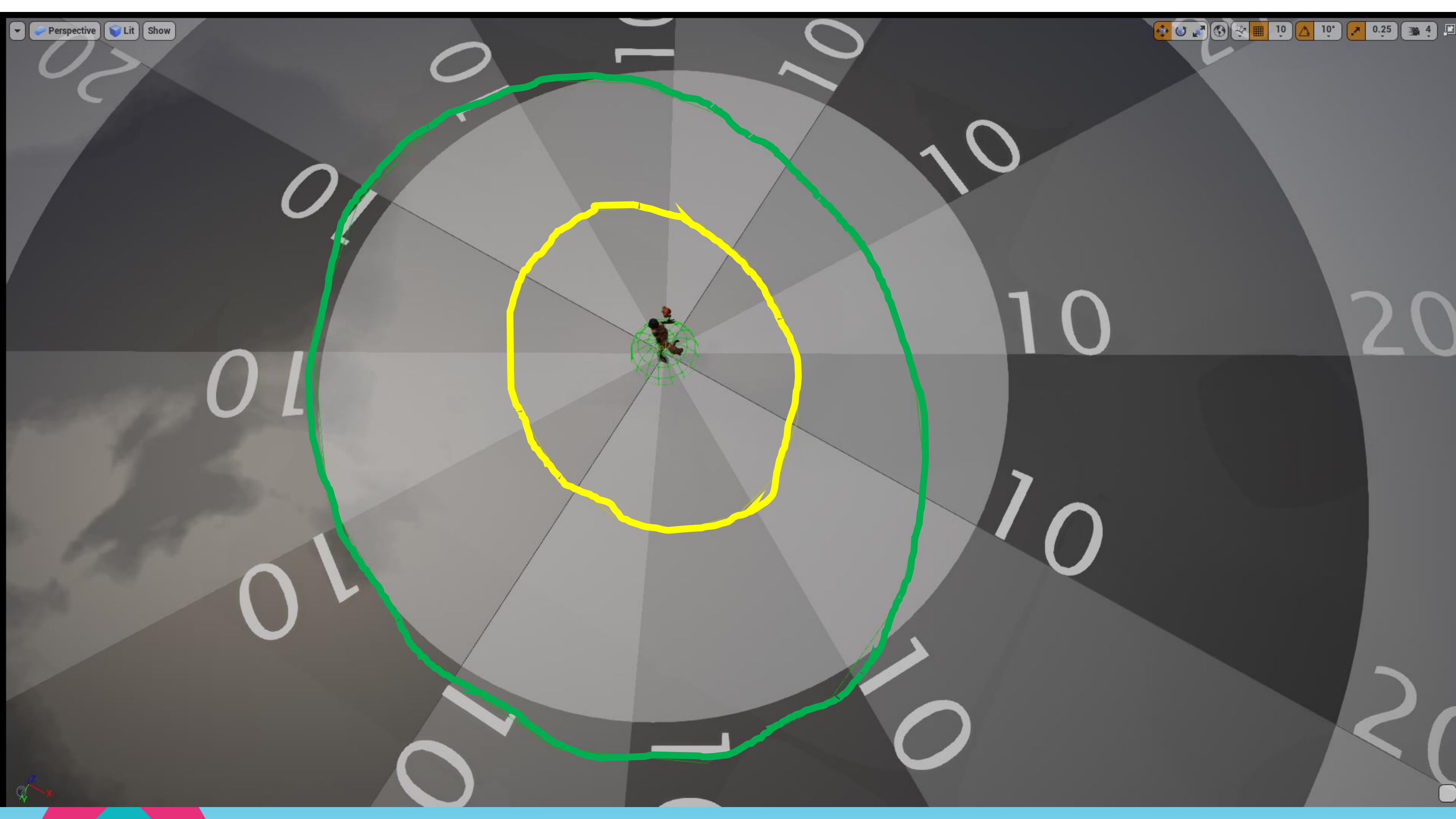
The Problem

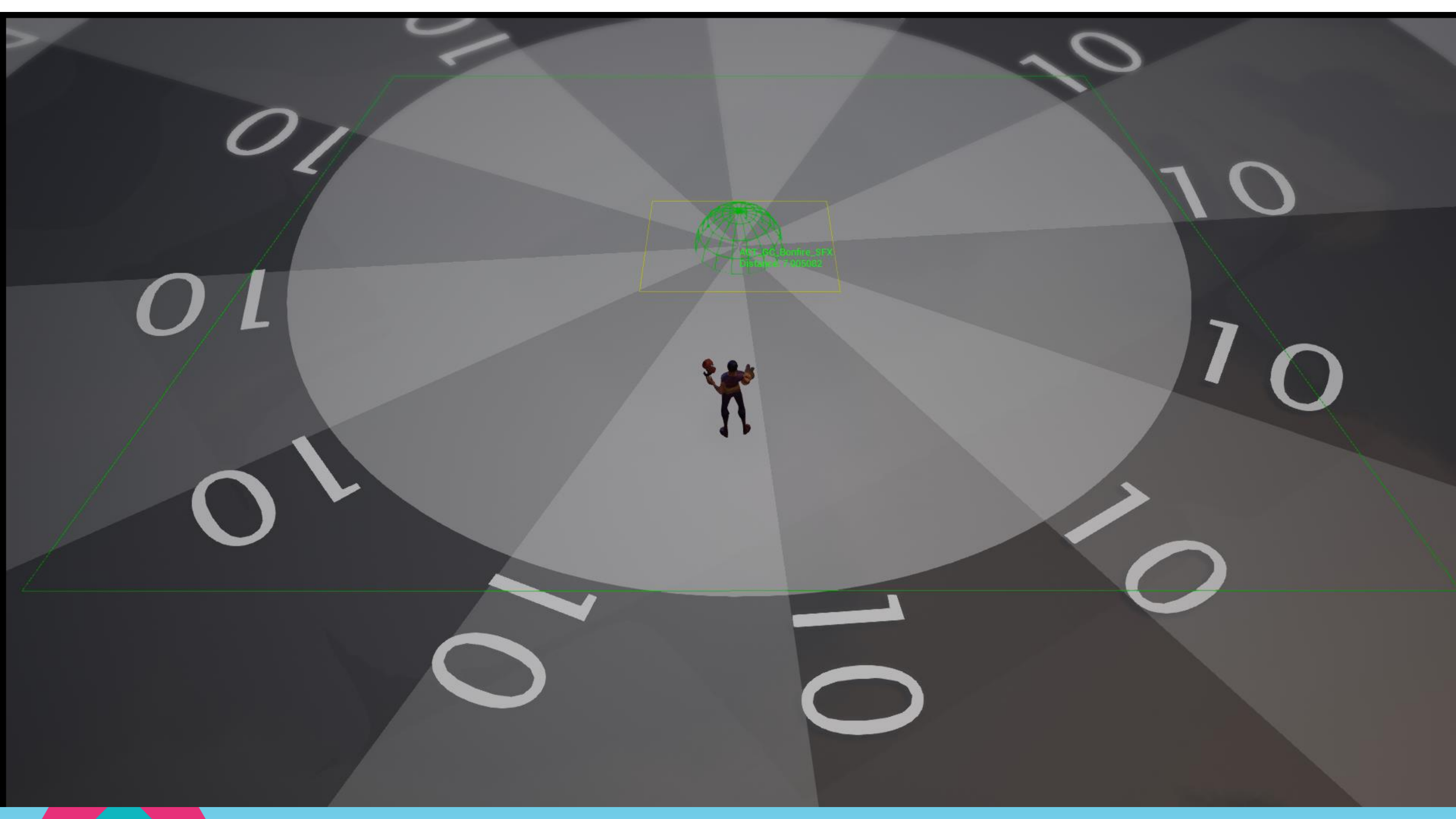


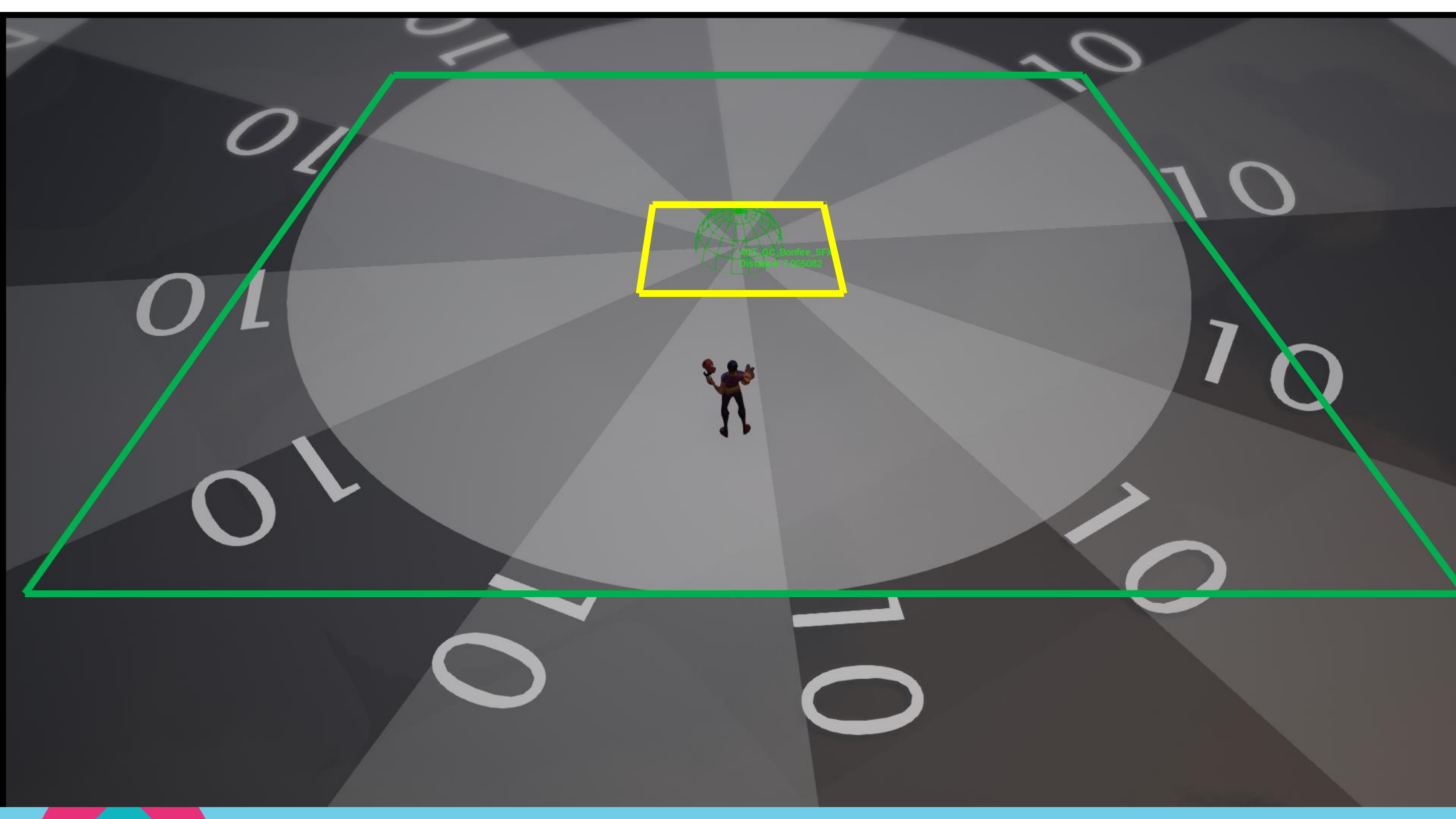




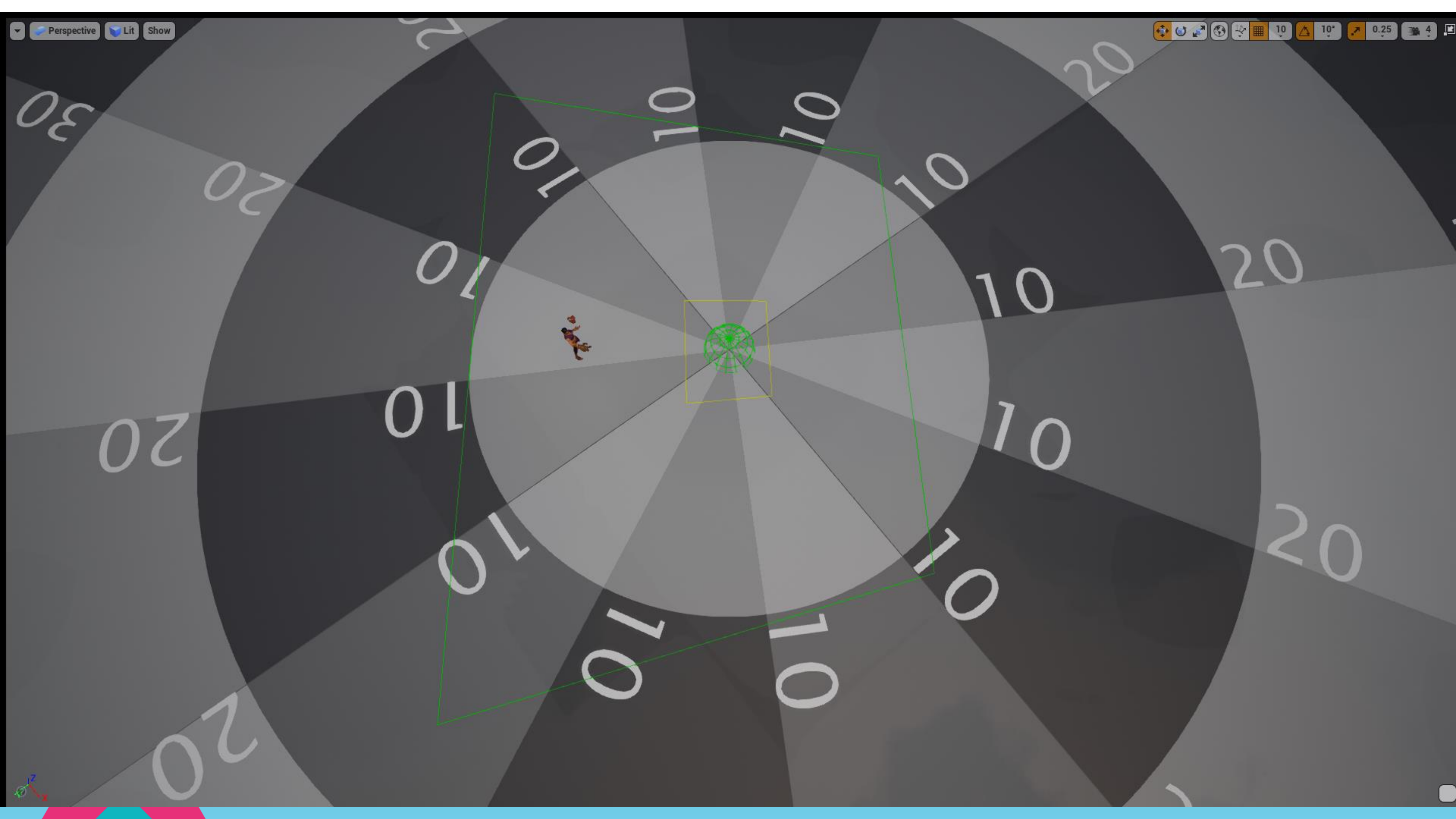


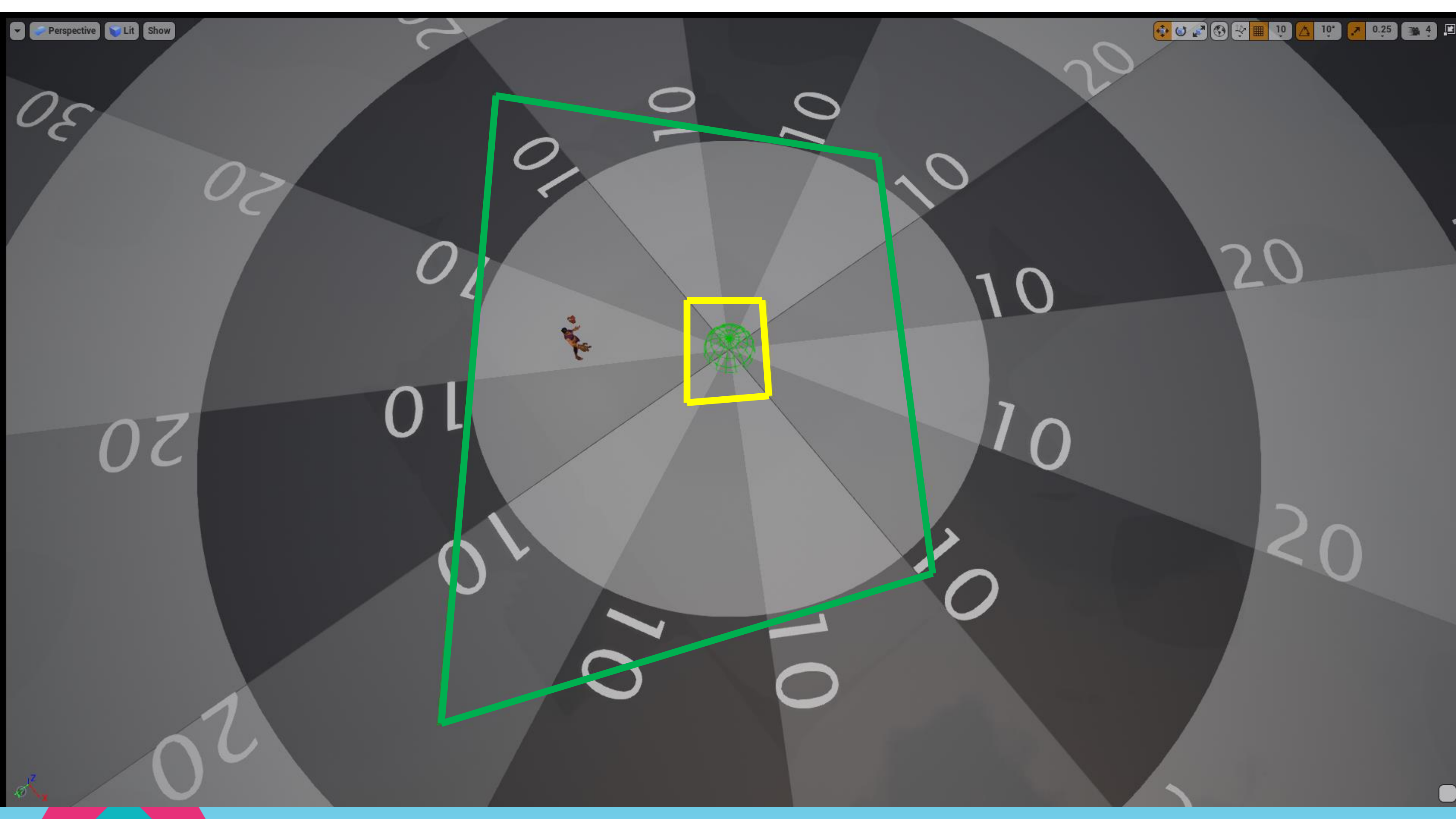






Box: SSC_Bonfire_SFV
Distance: 7.406182





Today's Topics

- Music
- Timed ADSRs

Topics

- Music
- Timed ADSRs

Music

- Composer from Day 1
 - Matt Uelmen – credits include Diablo, Diablo 2, Torchlight, Torchlight 2, and Hob.

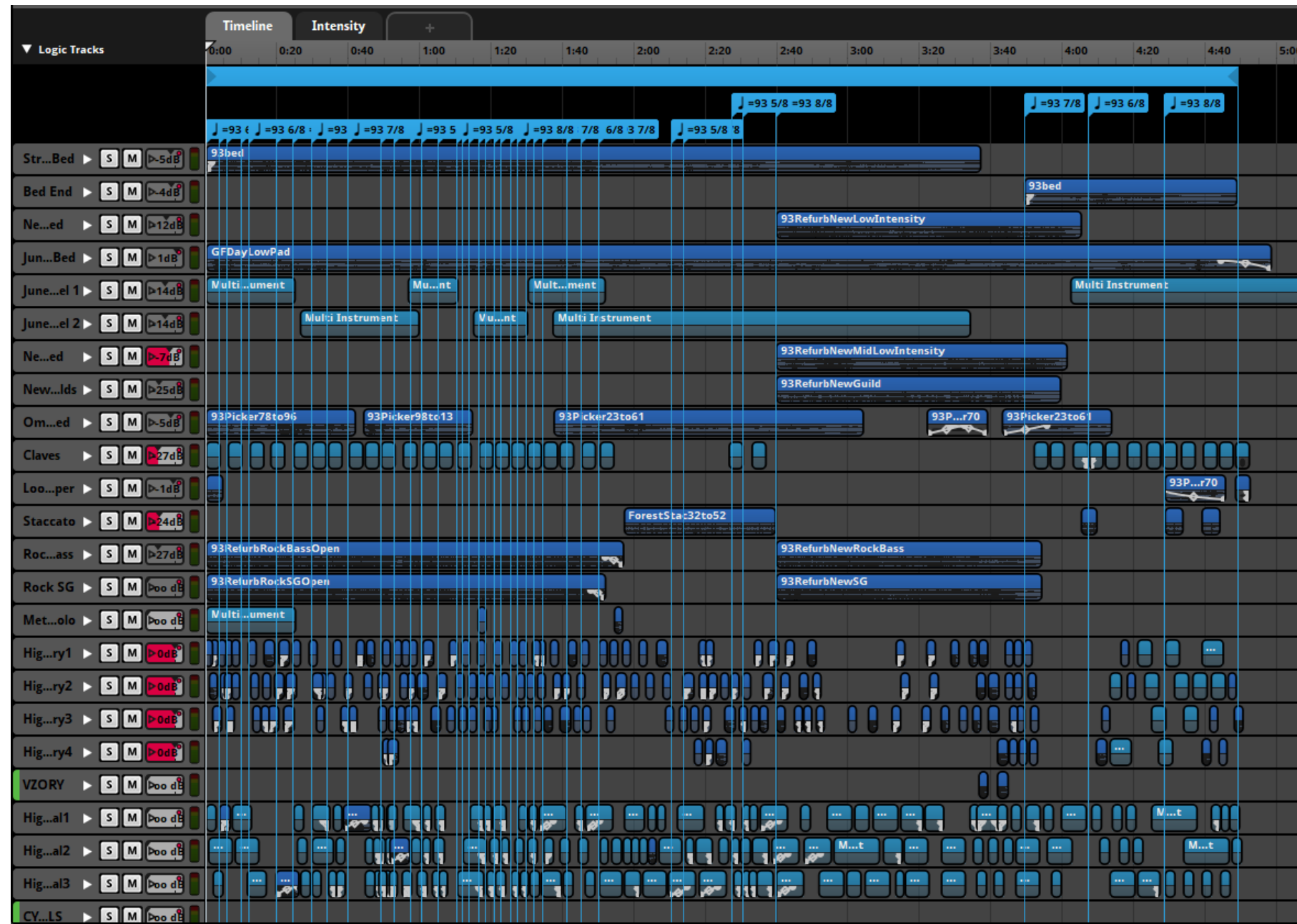
Initial Proposal

Action

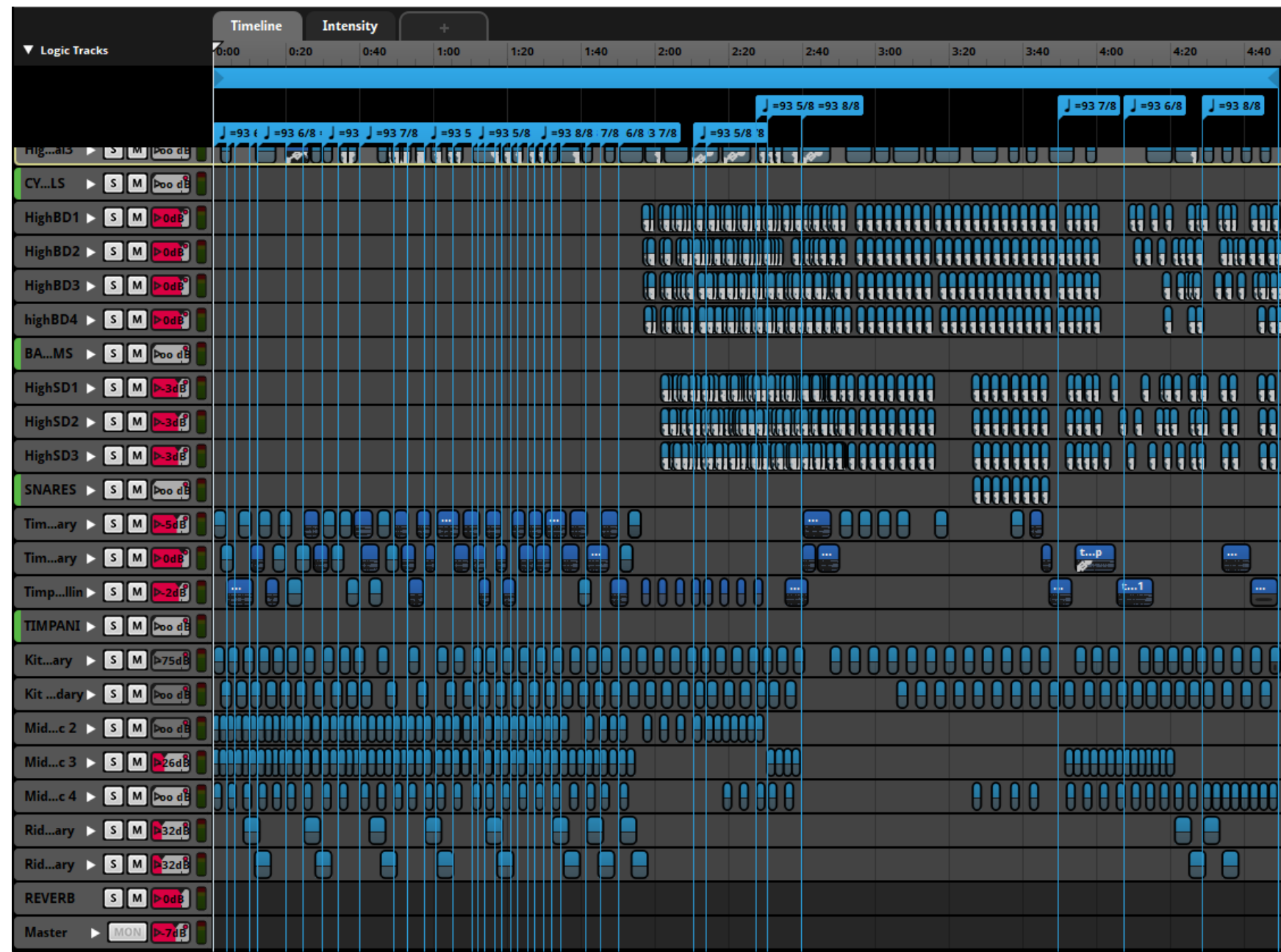
Medium

Idle

Where We Ended Up



Where We Ended Up



Events Banks Assets

Q-

- Ambience
- Breakables
- Character Sekt Events
- Characters
- Cinenatics
- Clickables
- Footsteps
- For Guy
- Generic Weapon Sounds
- Global Class sounds
- GUYTESTS
- Impacts
- Items
- Monsters
- Music
 - Boss Music
 - DNU
 - Echonok
 - Goblin Caves
 - Goblin Forest
 - GoblinEstablish_Cin_Music
 - Goblinforest Day #referenced
 - Goblinforest Music Bosses
 - Goblinforest Music**
 - Goblinforest Night #referenced
 - Goblin Lake
 - Imperial Outpost
 - Infested Forest
 - Menu
 - Player Fort
 - Workshop
- OLD Events
- Pets
- Skills
- Snapshot Events

New Event New Folder Flatten

Goblinforest Music

TIME BEATS STOPPED 00:00.000

Time Of Day MX Intensity

0.00 0.00

Logic Tracks

Timeline Time Of Day MX Intensity

music SOLO MUTE 0.0 dB

snapshot SOLO MUTE 0.0 dB

Master MON 0.0 dB

Fader

Pre Volume Post

0.00 dB

Click to add a pre-fader effect

Click to add a post-fader effect

Overview

Parameters

Local

Time Of Day MX 0.00

Intensity 0.00

Properties

Tags

User Properties

Notes

Audio Track

Pan 0.00

In Out

Music Track Features

- 10-44 Individual Tracks
- Day/Night cycle
- Three intensity levels

Calculating Intensity, First Pass

- Find monsters in radius
- Calculate “near drama” and “far drama”
- Low Drama – If near drama is above this, upgrade to medium intensity.
- Medium Drama – If far drama is above this, upgrade intensity (low->med or med->high)
- Min/Max time at each intensity
- Hold time at Medium intensity
- Player Health Percent Low->Med and Med->High

It worked!

- Music changed intensity
- Felt pretty good
- But nobody ever tweaked the numbers
- Why?
 - Too complex
 - Too many dials

Also, it didn't work

- Noise that didn't matter (i.e. player health)
- Confusing why it went up and down in intensity
- Game balance changes could affect the music inadvertently

New Hotness

- Each monster assigned a category:
 - Popcorn – 1 point
 - Standard – 3 points
 - Brute – 5 points
 - Champion – Treated as though it were a champion
- Popcorn, Medium, and Brute values for Medium and High intensities

New Hotness

- Find all monsters within radius
- Count the number for each category, but also convert to all other categories
- If any of these conversions is over the limits, then upgrade the music

Example

- Goblin Warrior – Popcorn
- Goblin Gunner – Standard
- Goblin Brute – Brute

- Medium:
 - 8 Popcorn, 3 Standard, 1 Brute
- High:
 - 15 Popcorn, 5 Standard, 2 Brute

Example

P=1, S=3, B=5 Med: 8P 3S 1B High: 15P 5S 2B

Warriors	Gunners	Brutes	Total Counts	As Popcorn	As Standard	As Brute	Intensity
3	0	0	3P 0S 0B	3	1	0	Low
3	2	0	3P 2S 0B	9	3	1	Medium
3	0	1	3P 0S 1B	8	2	1	Medium
15	0	0	15P 0S 0B	15	5	3	High
0	4	1	0P 4S 1B	17	5	3	High
				$P + 3S + 5B$	$\left\lceil \frac{P + 3S + 5B}{3} \right\rceil$	$\left\lceil \frac{P + 3S + 5B}{5} \right\rceil$	

Extra Rules

- Champions:
 - 1 Champion => Medium
 - 2+ Champions => High
- Player is Dead => Low
- Loading Screen => Low
- Bosses => Override music

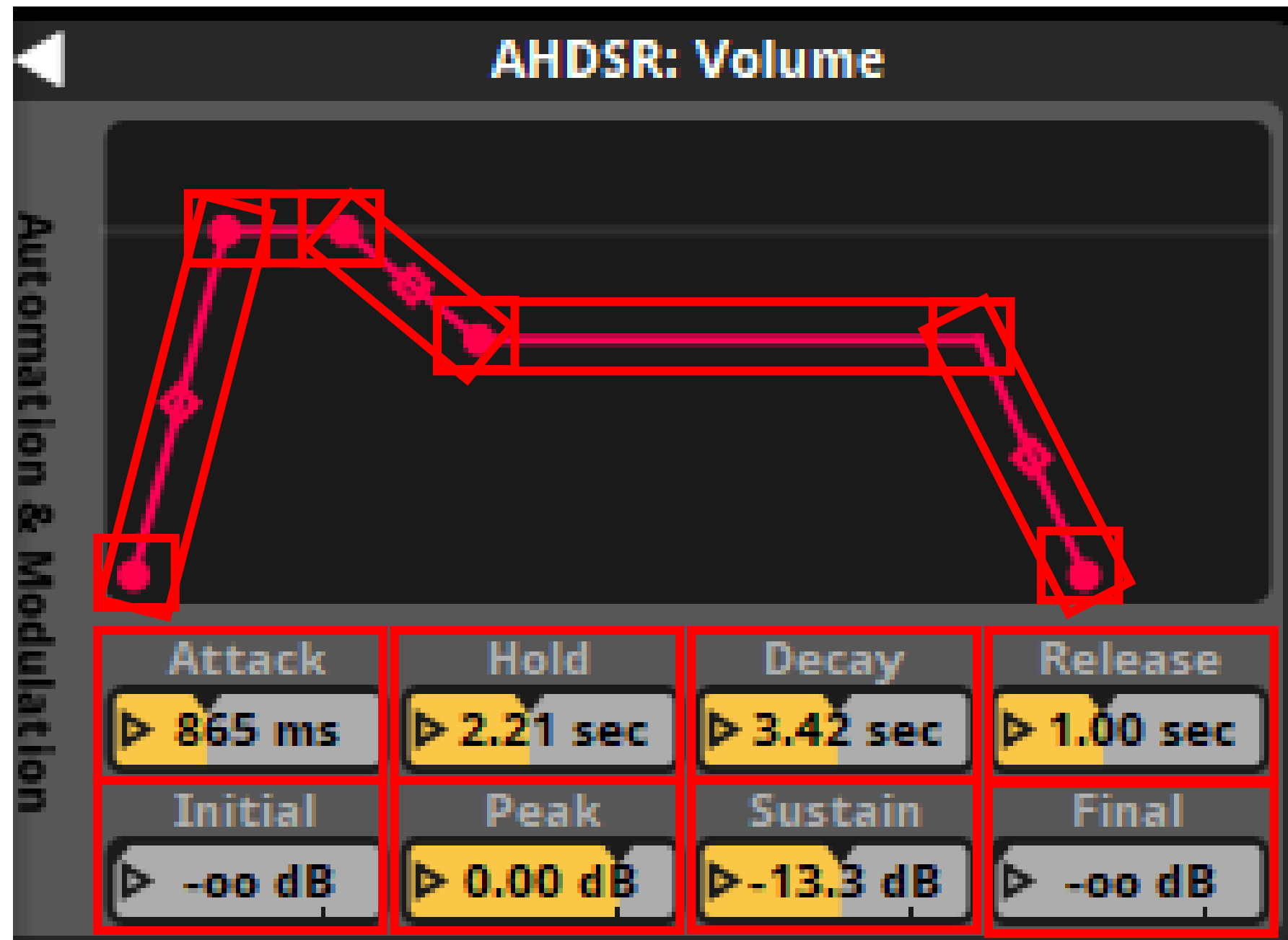
Time Limits

- Low intensity silence timer
- Min time at Low intensity
- Min time at Medium intensity
- Hold time at Medium intensity
 - Can upgrade to High, but not downgrade to Low
- Min/Max time at High intensity

Topics

- Music
- Timed ADSRs

What is an A(H)DSR?



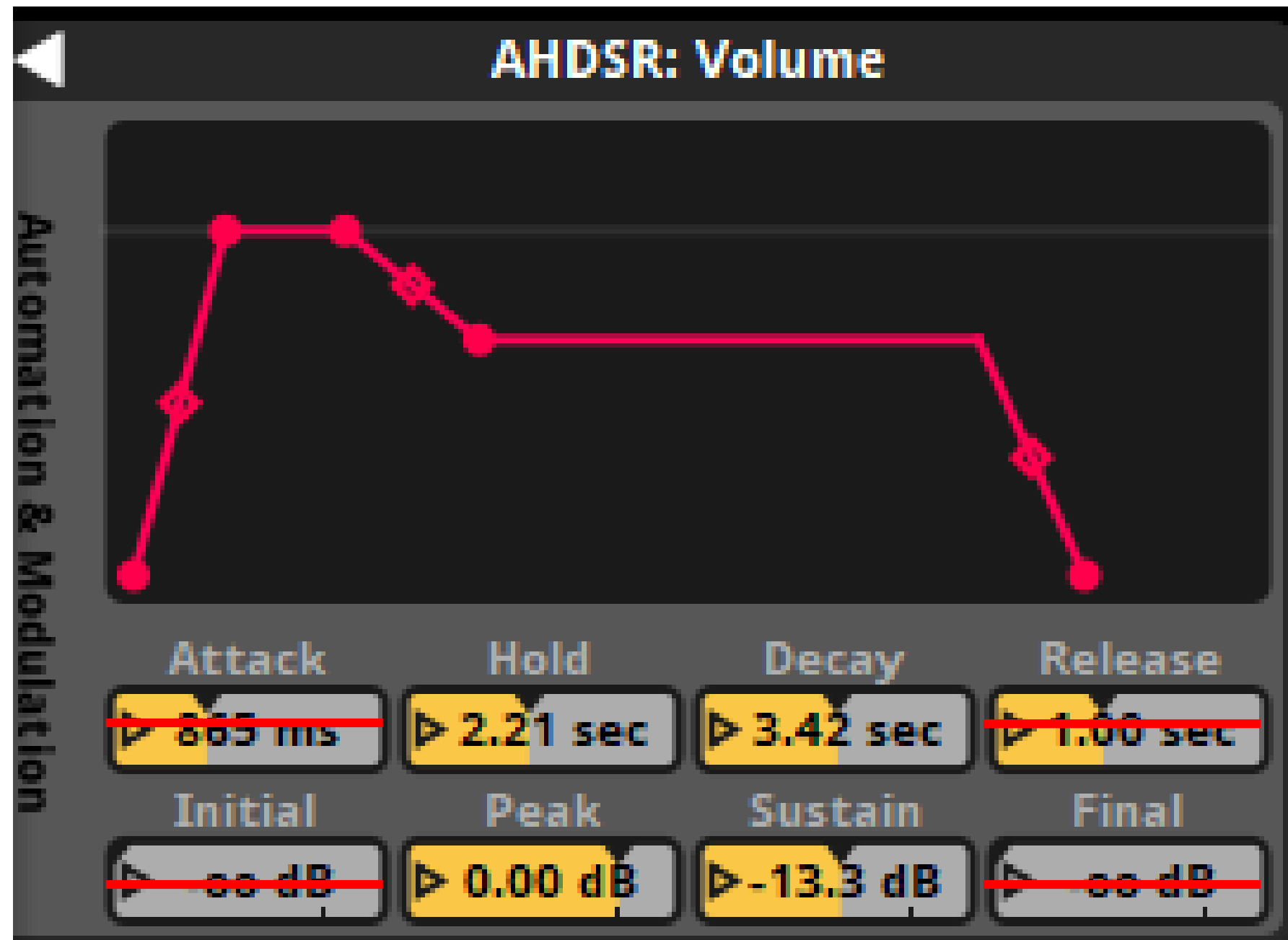
ADSRs

- Fade in/Fade out
 - Hold + Fade
 - Automatic
-
- But:
 - Only apply to a single event

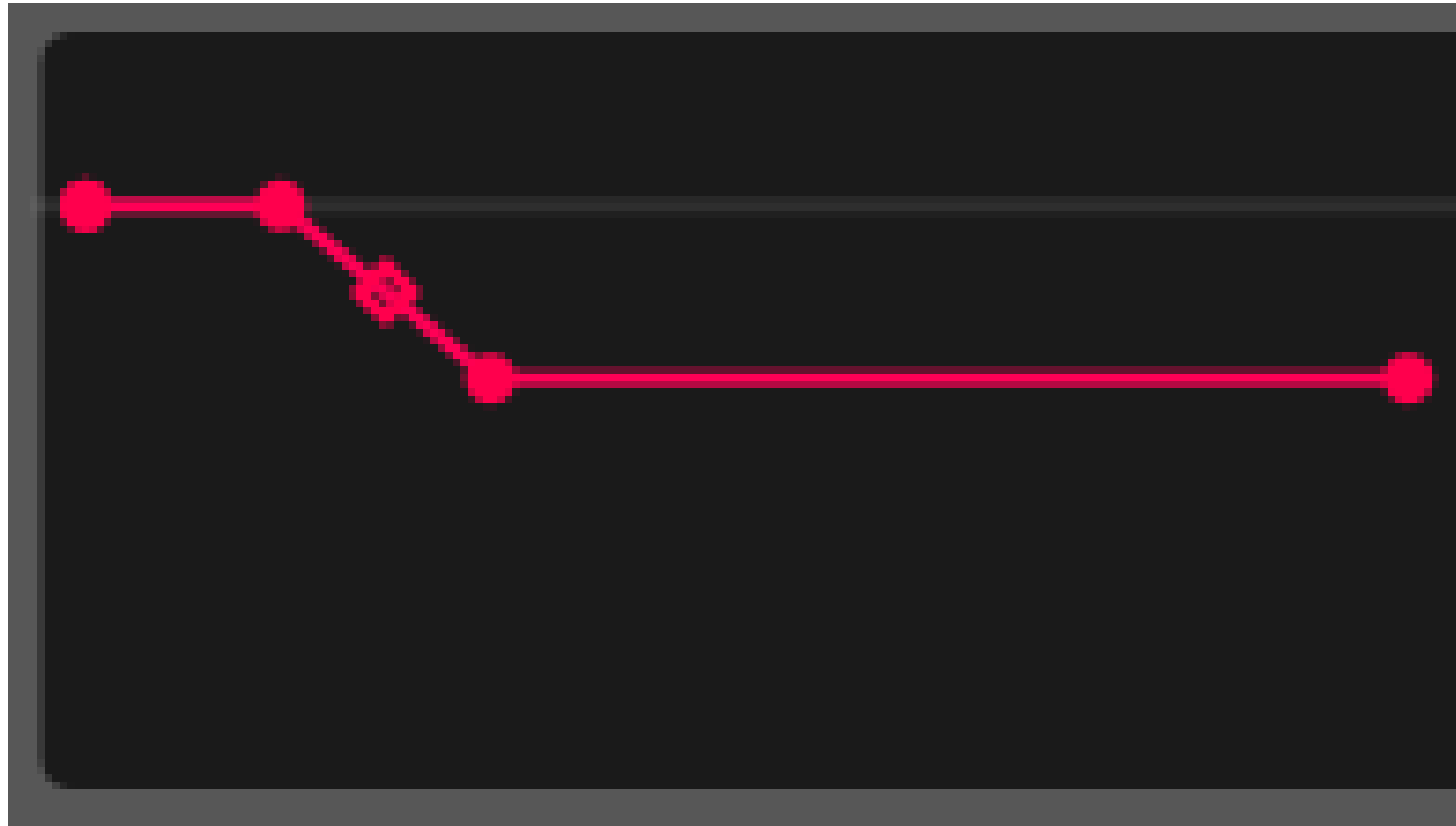
The Problem

- AHDSR works fine for Holy Fury
 - Beam skill, looped sound
- AHDSR doesn't work for Holy Bolt
- Need a solution that works for repeated one-shot sounds

Solution: Timed ADSRs



Timed ADSR Curve



Timed ADSRs

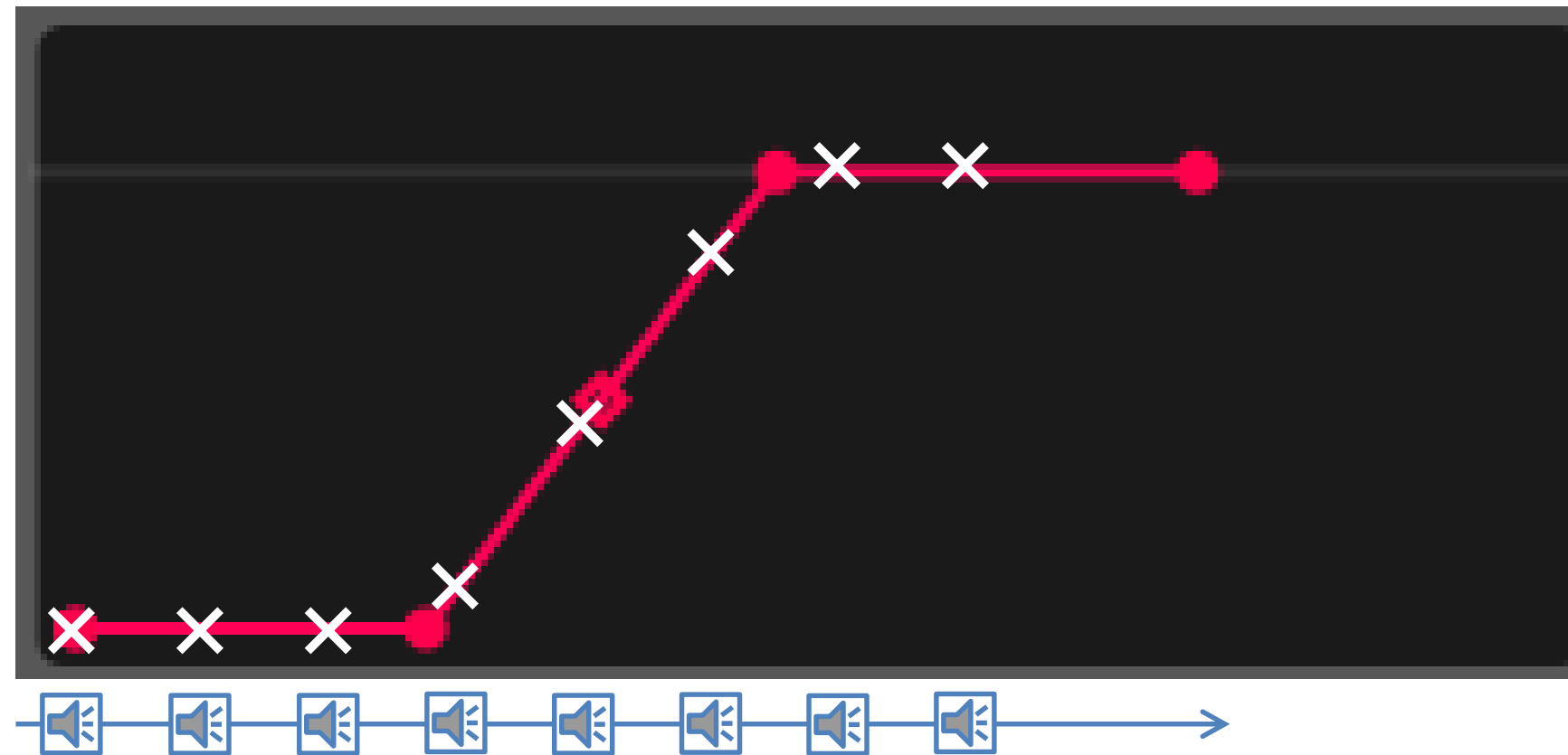


TimedADSR

Timed ADSR Parameters

- Parameter
 - **TimedADSR**
- Properties:
 - **Attack Time** – How long to leave TimedADSR parameter at value 0.0
 - **Fade Time** – Time taken to lerp TimedADSR parameter to value 1.0
 - **Reset Time** – How long to wait since the last instance was played before resetting Timed ADSR back to 0.0

Timed ADSR Curve





Trevail Passage

Quests

- Fungus Forager**
Go to **Trevail Point**
Talk to **General Graye**
- Tutorial: Props & Property**
Go to **Trevail Point**
- Tutorial: From a Humble Seed**
Go to **Trevail Point**
Talk to **Zaya**
- Tutorial: Gambler**
Go to **Trevail Point**

Online Mode
PlayerFort -1110956014
gob_FP_fortent_20x60_01
TeleportPlayer 9035.00, 12615.00, 150.00, yaw: 0deg
Build:92460 Content:latest Env:dev Release:none
AID: RedOctober_NOLOGIN_ACCOUNT/Offline-Account
CID: RedOctober_NOLOGIN_CHAR_ID ZID:unknown_zone
Platform: Arc (1)

CTRL+D to copy
en
66s world time
103fps

Details

- Timed ADSRs must be per-actor, per event
- Most events don't have TimedADSRs

Implementation Details

- Audio Engine:
 - Stores a mapping between events and TimedADSRContexts
- TimedADSRContext:
 - Caches attack, fade, and reset times
 - Stores an array of TimedADSRs
- TimedADSR:
 - Stores instigator, start time, and last trigger time

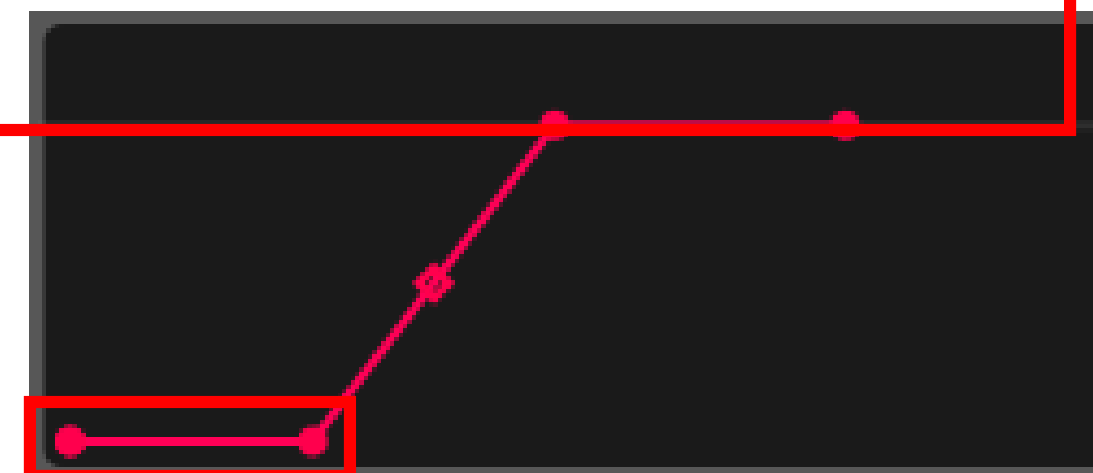
TimedADSR Implementation

```
struct TimedADSR {  
    TweakObjectPtr<const AActor> Instigator;  
    float StartTime;  
    float LastPlayTime;  
    float GetParameterValue(  
        const FTimedADSRContext& Context,  
        float CurrentTime) const;  
    bool IsExpired(const FTimedADSRContext& Context,  
        float CurrentTime) const;  
}
```

TimedADSR Implementation

```
float GetParameterValue(  
    const FTimedADSRContext& Context,  
    float CurrentTime) const {  
    auto TimeSinceStart = CurrentTime - StartTime;  
    if (TimeSinceStart < Context.AttackTimeSeconds)  
        return 0.0f;
```

...



TimedADSR Implementation

...

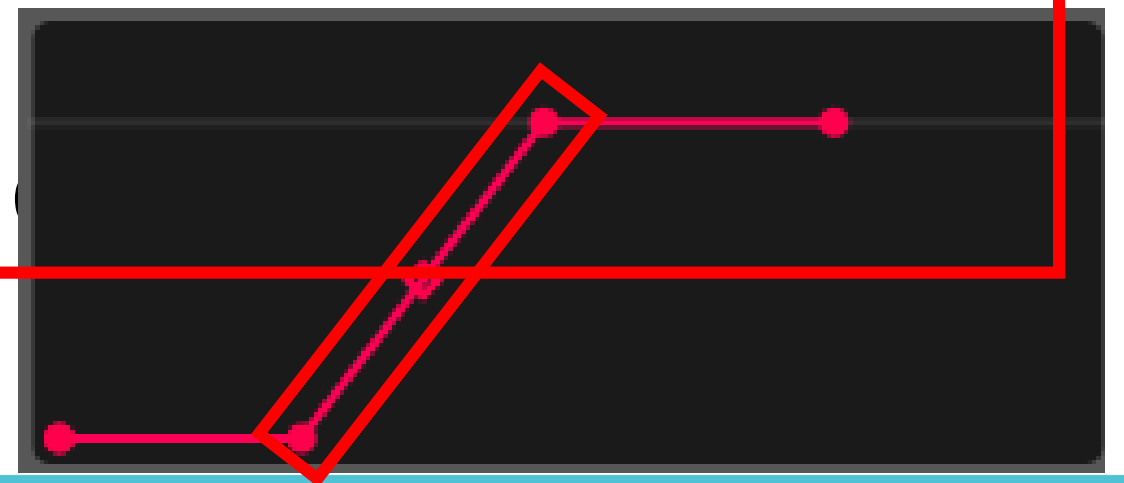
```
TimeSinceStart -= Context.AttackTimeSeconds;
```

```
if(TimeSinceStart < Context.FadeTimeSeconds)
```

```
return
```

```
    Lerp(0.0f, 1.0f,  
    TimeSinceStart / Context.Fa
```

...

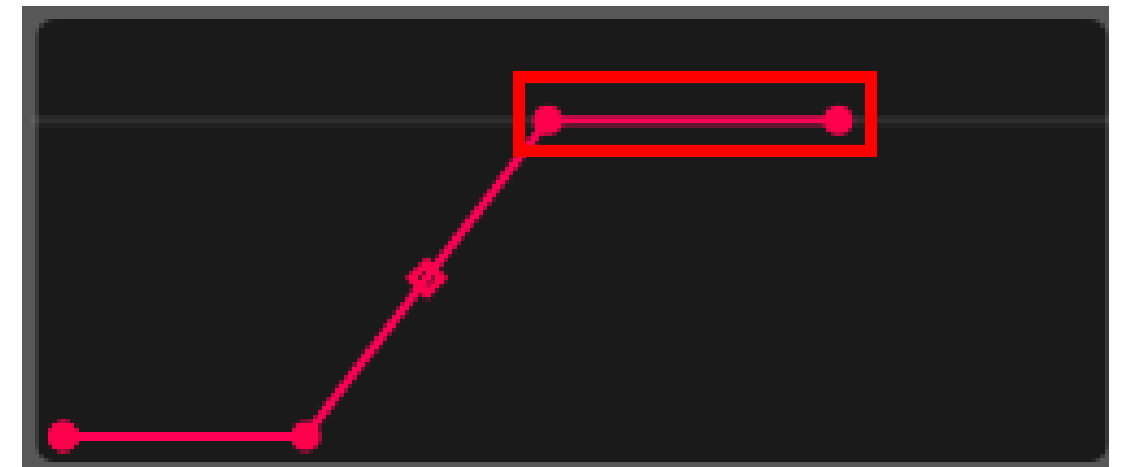


TimedADSR Implementation

...

```
return 1.0f;
```

```
}
```



TimedADSR Implementation

```
bool IsExpired(const FTimedADSRContext& Context,  
               float CurrentTime)  
{  
    auto TimeSinceLastPlay =  
        CurrentTime - LastPlayTime;  
    return  
        TimeSinceLastPlay > Context.ResetTimeSeconds;  
}
```

Conclusion

- ARPGs have distinctive (but not unique) challenges
- Dynamic Music is hard to get right
 - Occam's Razor applies: simpler solutions are usually better
 - Even the simple solution can have complexities
 - Be eager to trigger action music
 - But be mindful of ear fatigue – set limits!
- Timed ADSRs are a good solution for ADSRs for one-shot sounds
 - Track per-event, per actor
 - No ticks!