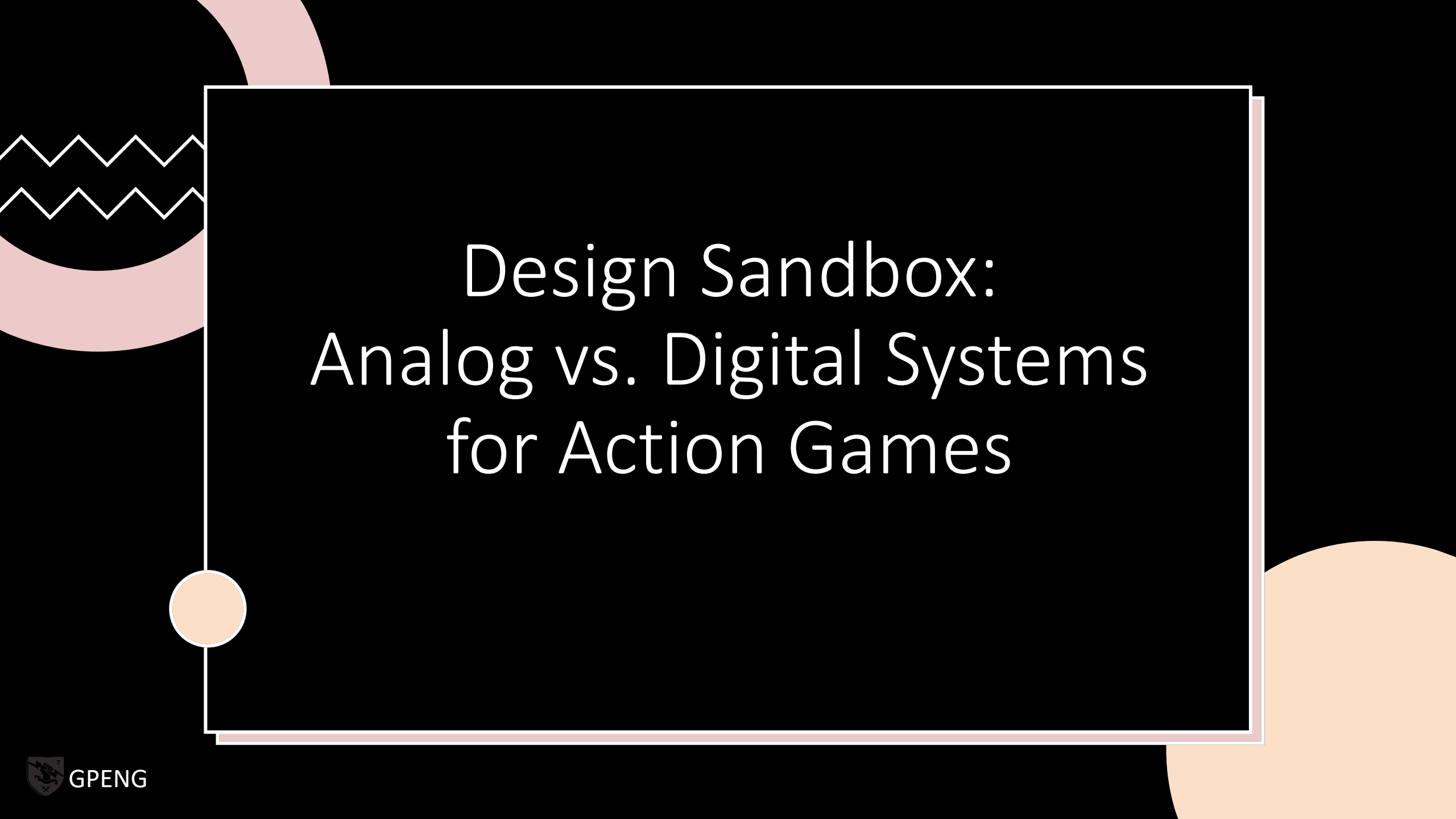




Design Sandbox: Analog vs. Digital Systems for Action Games



OR

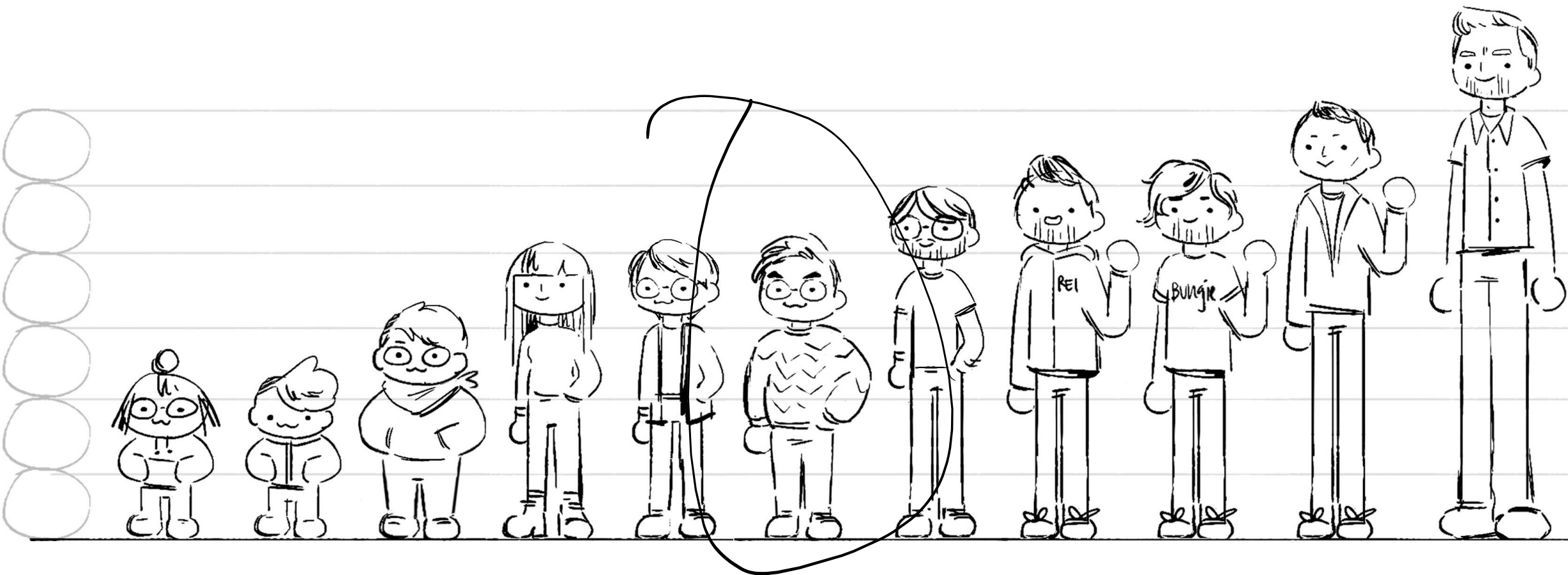


Why a good curve
editor can be a
competitive
advantage

Hello I'm Greg Peng

- Principal Sandbox Designer at Bungie
- Weapons Feature Lead



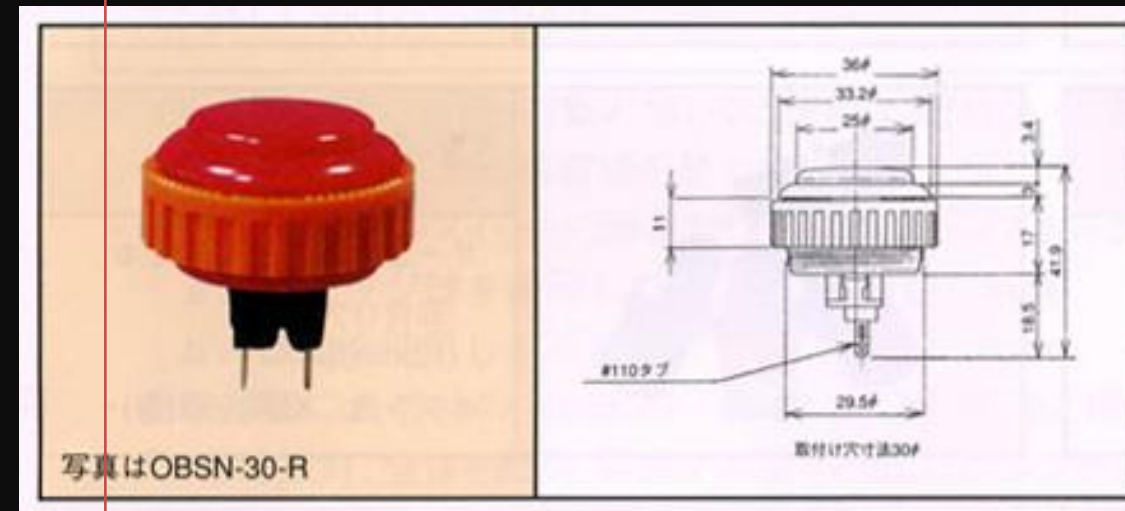


PREAMBLE

Let's start by contextualizing
Analog and Digital for
interaction design

Our Definition of Digital Input & Systems

- Discrete and easy to perceive the differences between
 - Small/medium/large, off/on
- 3 is a magical design number
 - Easy to remember, human brains like 3
- Input: single button press
- Why Digital?
 - Easier to perform
 - Easier to understand
 - Easier to repeat exactly



Our Definition of Analog Input & Systems

- Enough fidelity where human perception can't separate between adjacent steps
 - i.e. 0-255 is basically continuous
 - Floating point numbers
- Input: Analog Sticks, Mouse
 - Lots of data, requires precision
- Why Analog?
 - More data to divine intention, lending to improved player expression
 - Can represent more complex systems / be used to simulate reality



Awkward Middle Ground

- ~6-15 states
 - Harder to feel the difference between each state
 - But also lose the squishiness of analog representation
 - Worst of both worlds?
- Grading Systems (A+ → F-)
 - People argue about grading systems and what they really represent
 - [NIH Study](#) on Patient Survey Accuracy
 - 5 pt scale more accurate than 10



Who really knows what any of this means?

Points-Based Classroom Report

Name	Homework 1	Quiz 1	Homework 2	Quiz 2
Zoe	80	80	90	80
Jayden	95	95	90	90
Pierce	85	90	0	80
Juan	90	85	95	95

Standards-Based Classroom Report

Name	Objective: Create a thesis statement	Objective: Create topic sentences	Objective: Organize ideas	Objective: Identify evidence
Zoe	Not proficient	Partially proficient	Partially proficient	Partially proficient
Jayden	Advanced	Advanced	Proficient	Proficient
Pierce	Partially proficient	Proficient	Proficient	Proficient
Juan	Proficient	Partially proficient	Advanced	Advanced

Image Source: The Atlantic

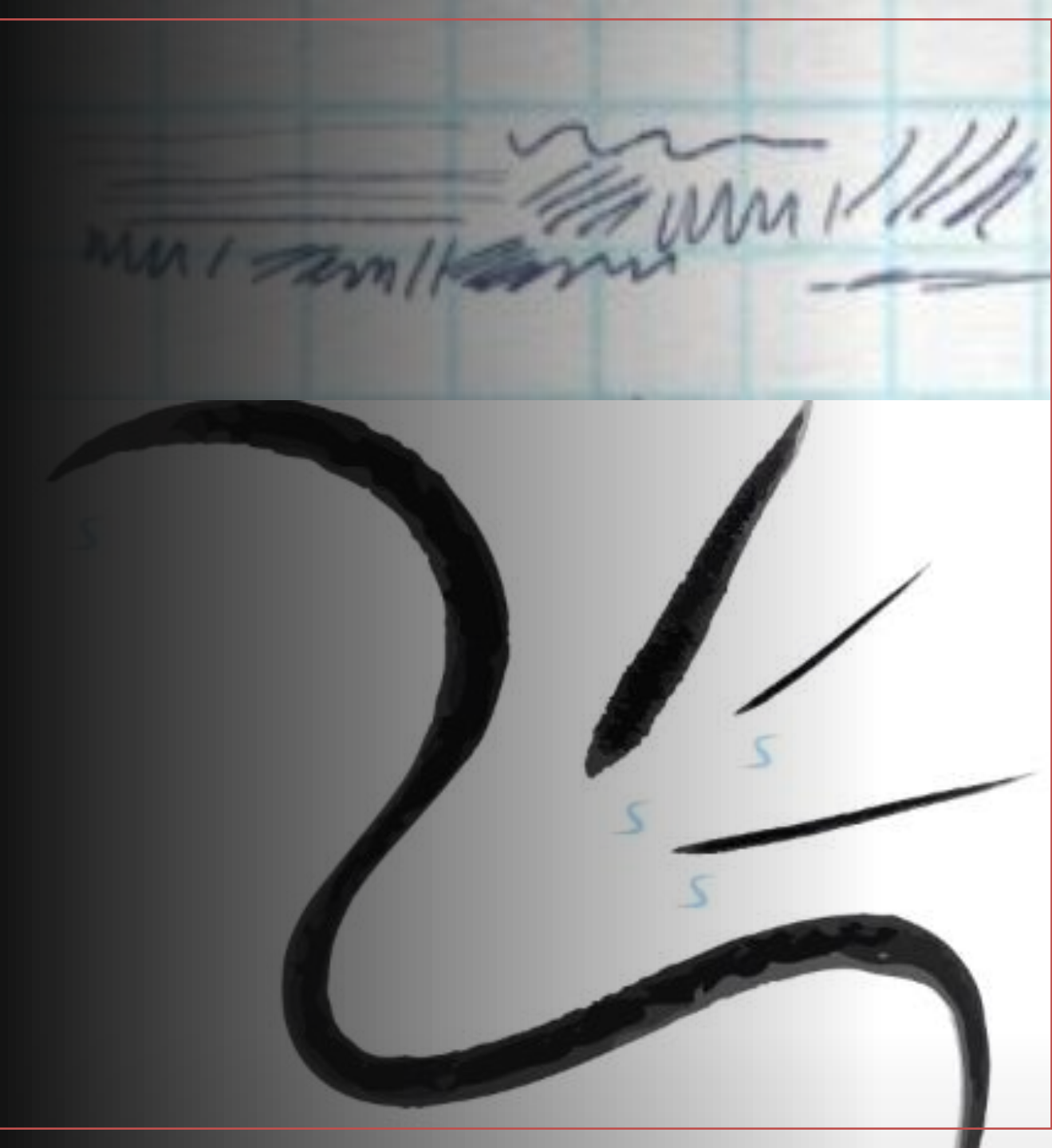
Bungie action games are known for their deep simulation and expressiveness.

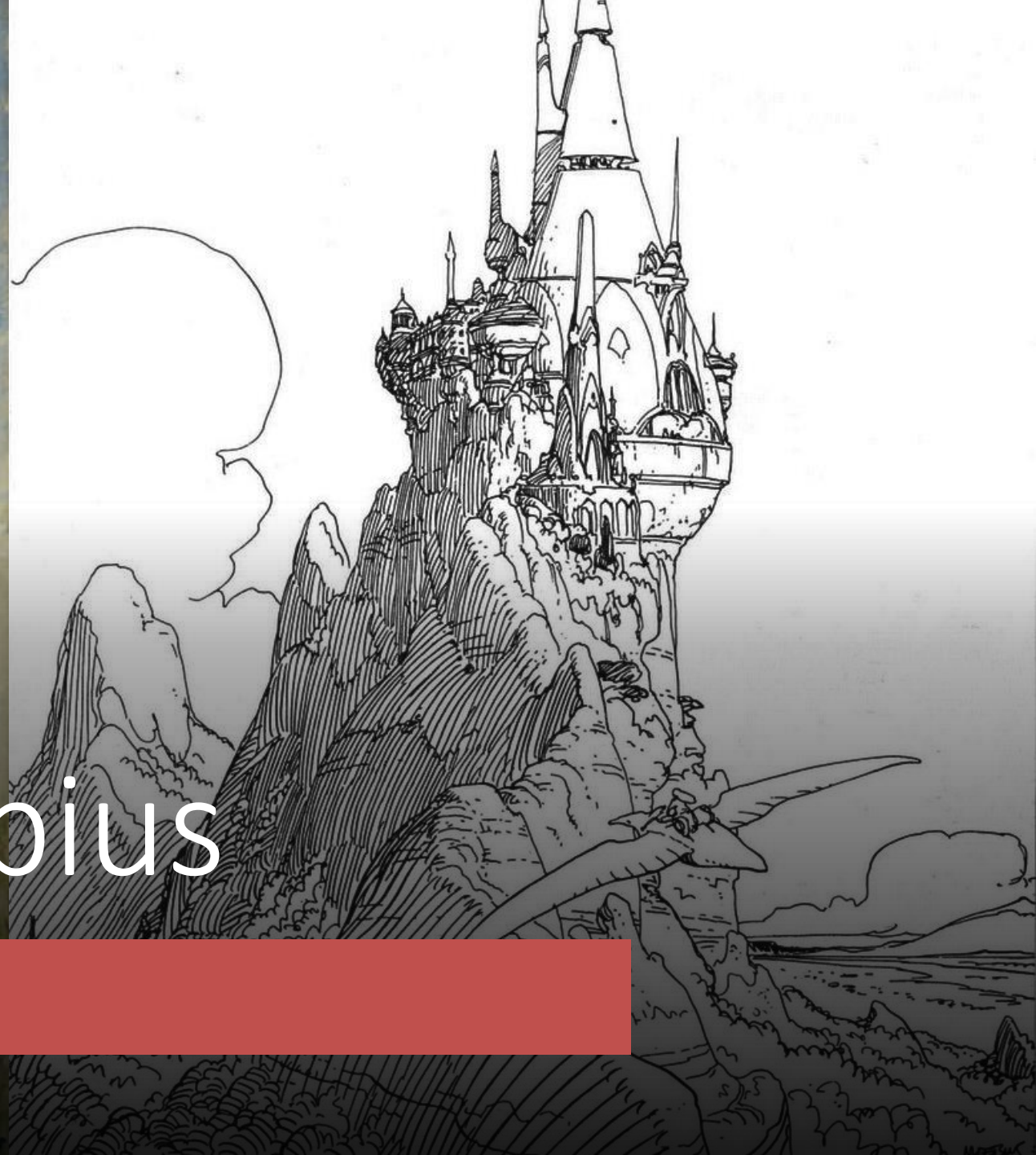
So let's just make every system we can analog, right?

NOPE

Choose tools carefully

- Analogy: Paintbrush vs Pen
 - Inherently, paintbrush allows fine control of line width through modifying pressure, angle, speed
 - Pen is gonna pen, clean fixed line width
- Way more effort on user to learn how to use a paintbrush
 - More work to become proficient
 - More effort to use in the moment





Monet or Moebius

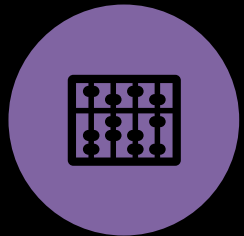
Coverage



**How do you simplify
what is complex?**



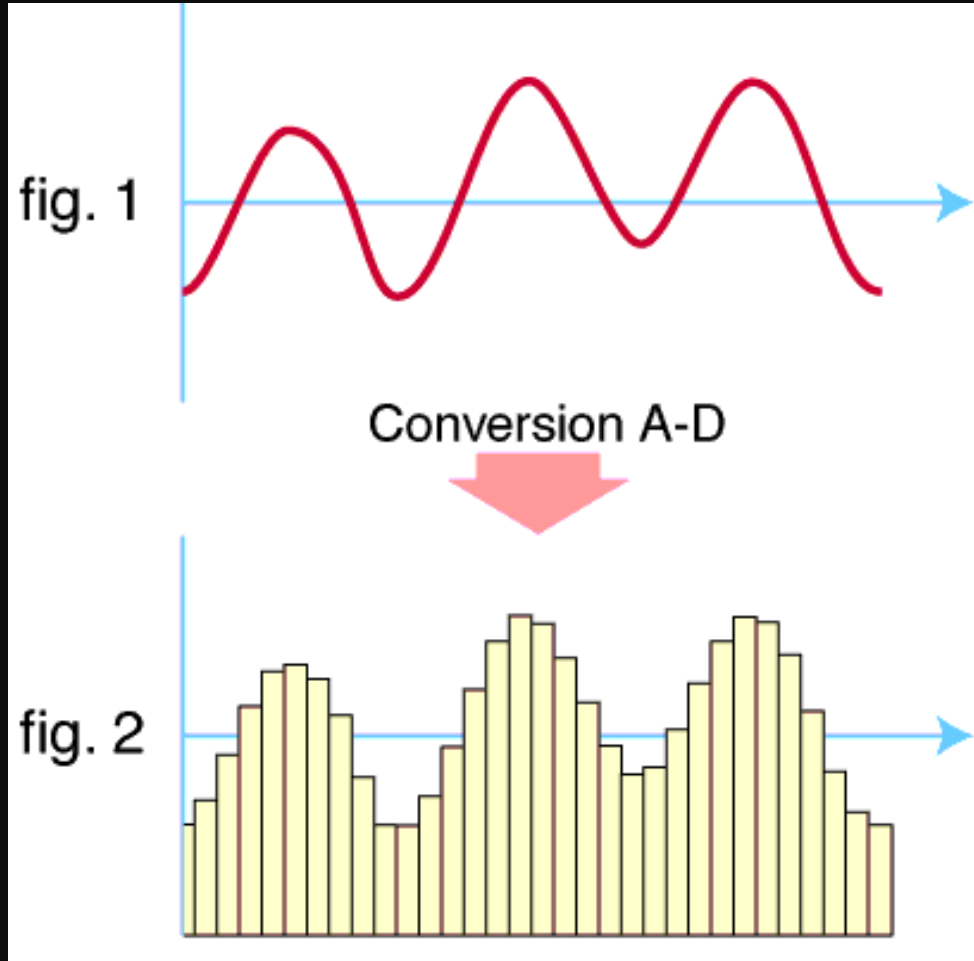
**How do you generate
complexity from
what is simple?**



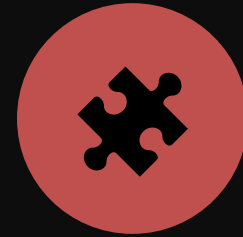
**How do you evaluate
what you've
created?**



**Why does
expressiveness
matter?**



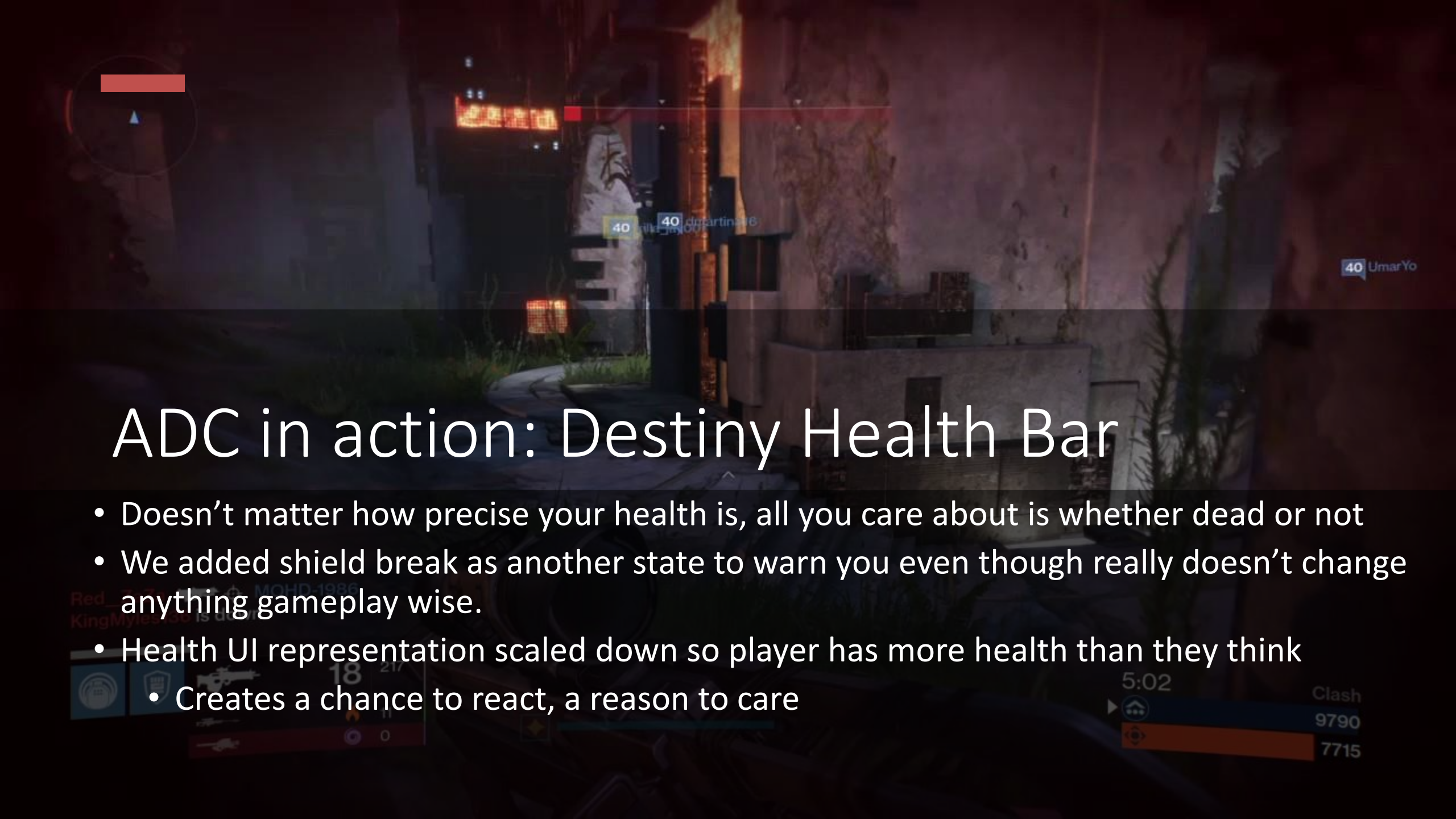
Analog to Digital Conversion (ADC)



How do you simplify
what is complex?

Concepts at a
mechanical/computational scale that
we are applying to a human perception
& cognition scale

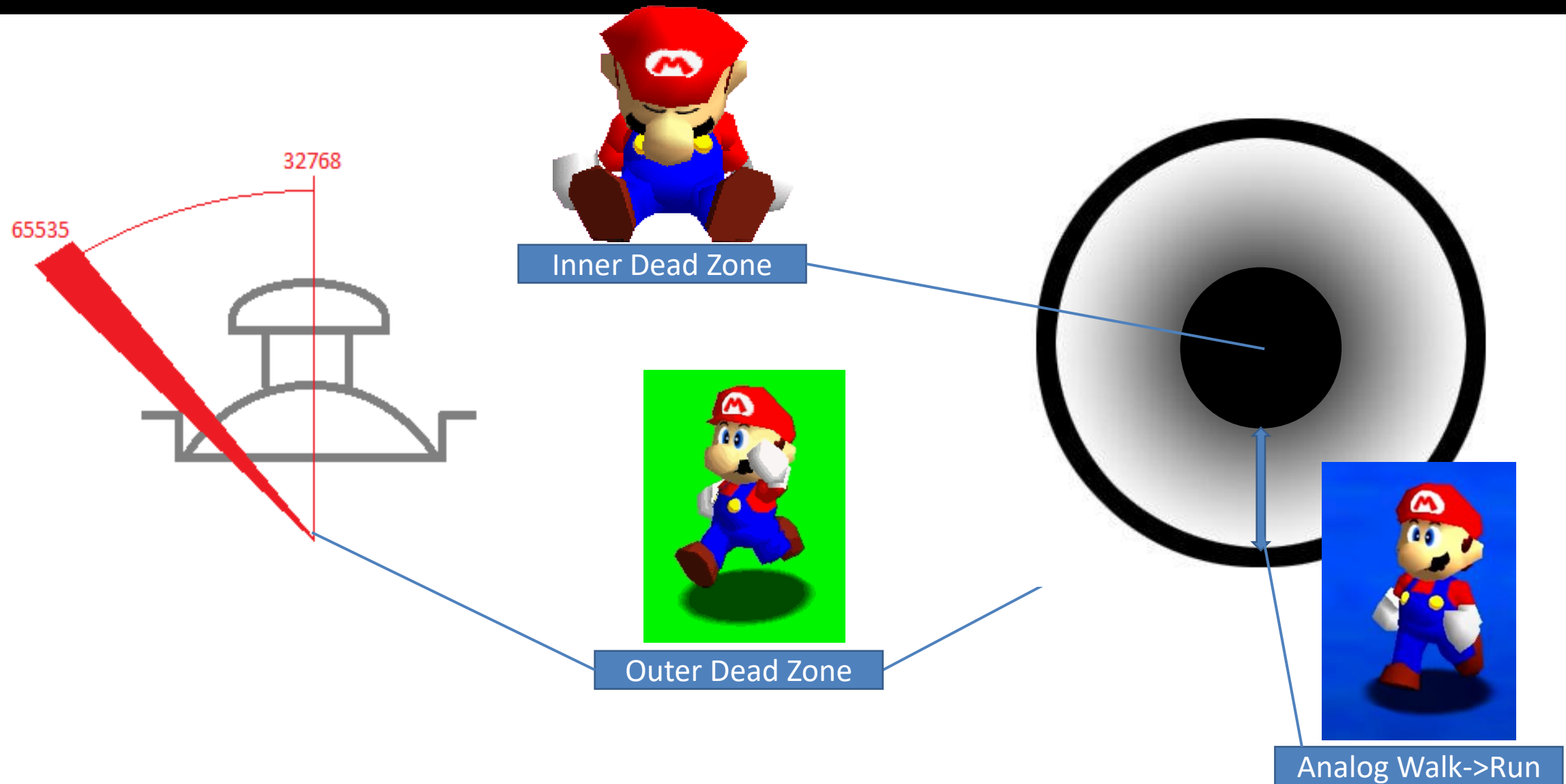
Let's look at an analog continuous
system that is simplified for the player

A screenshot from the video game Destiny. The background shows a dark, industrial environment with a large, glowing orange health bar in the upper center. The health bar is partially filled with red. In the foreground, there are some dark, metallic structures and a small, glowing orange light. The overall atmosphere is dark and moody.

ADC in action: Destiny Health Bar

- Doesn't matter how precise your health is, all you care about is whether dead or not
- We added shield break as another state to warn you even though really doesn't change anything gameplay wise.
- Health UI representation scaled down so player has more health than they think
 - Creates a chance to react, a reason to care

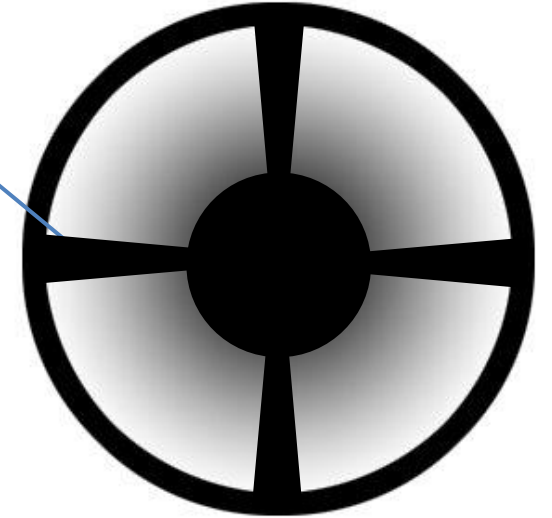
ADC: Analog Input That Uses Digital Thresholds to Aid



Analog Input That Uses Digital States to Aid



Directional
Dead Zone



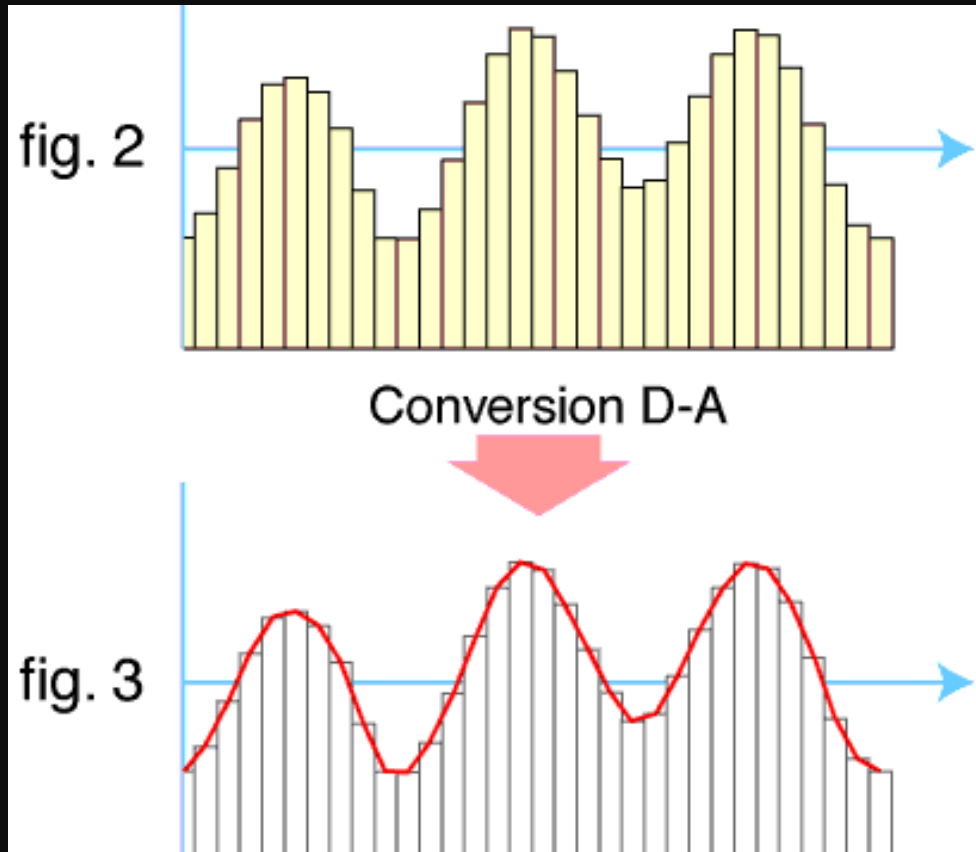
Strafe Movement (for Shooters)

- Very often a player wants to move precisely in relation to the current camera view: WASD
- For this movement model we have dead zones in the cardinal directions
- Not the best for 3d platformers
- Physical implementation: 8-way gate





How do you generate complexity from what is simple?

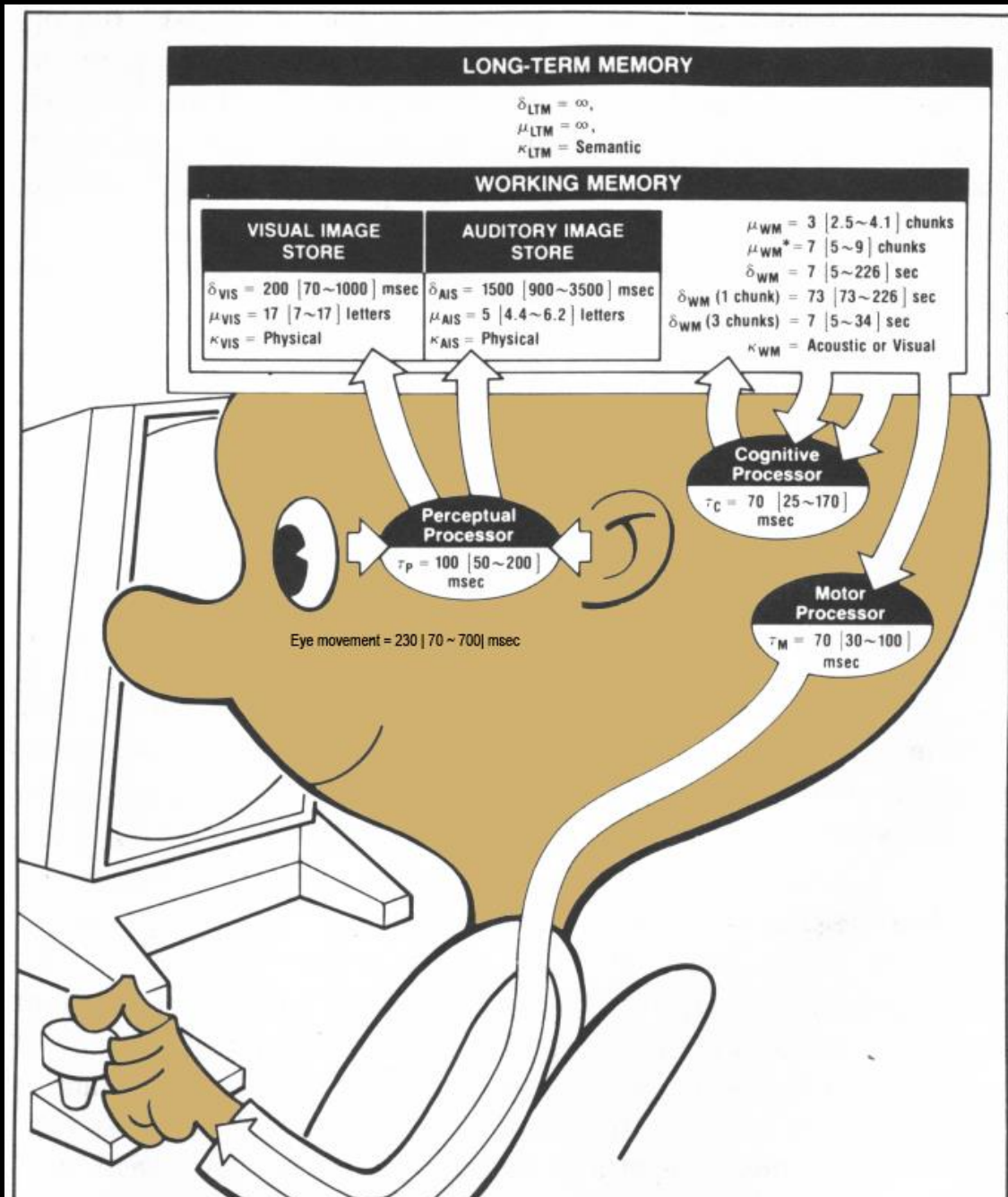


Digital to Analog Conversion (**DAC**)

Let's take a digital input and create analog simulation from it



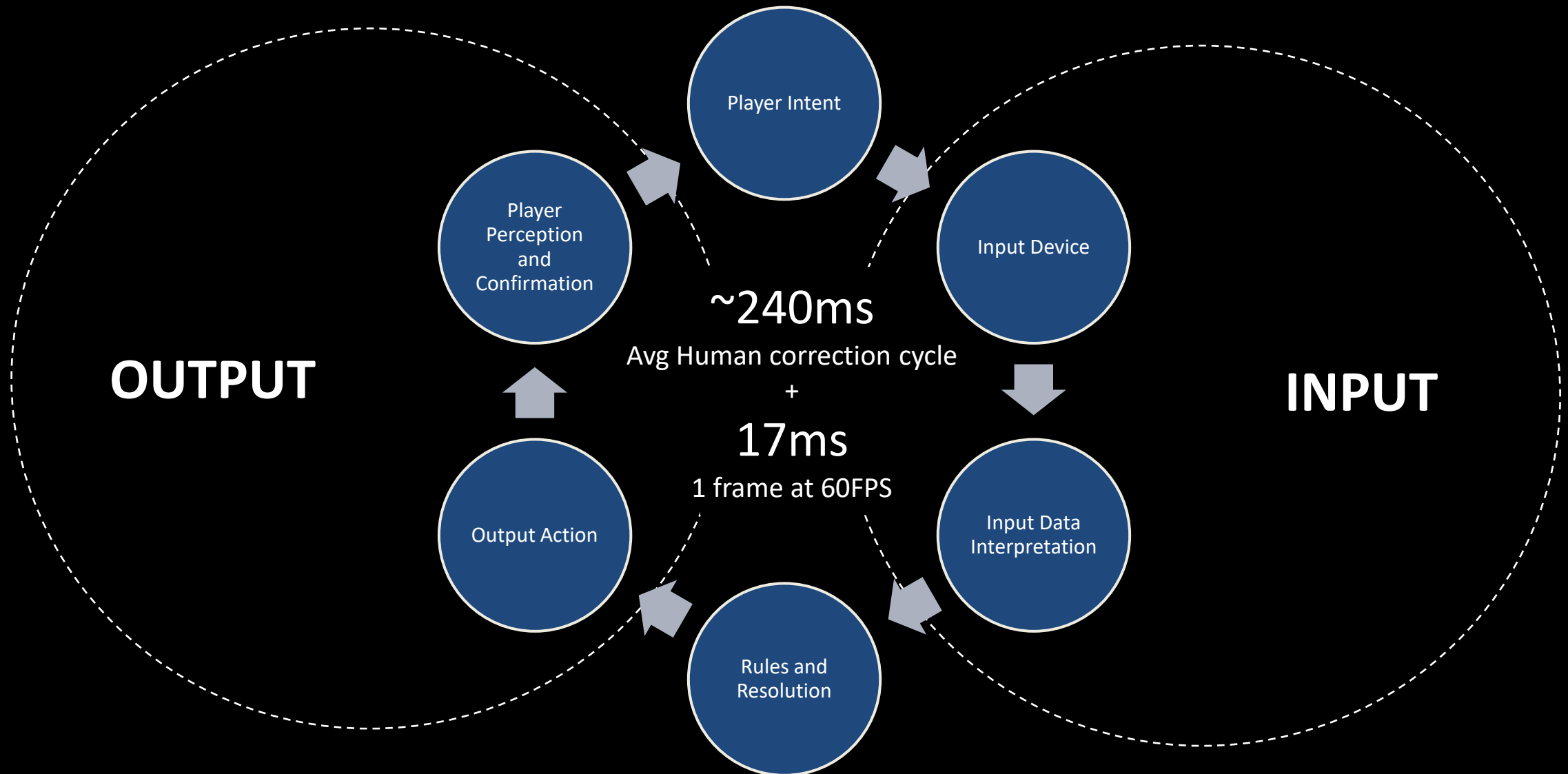
BUT FIRST
A quick primer on HCl



Human Processor Model

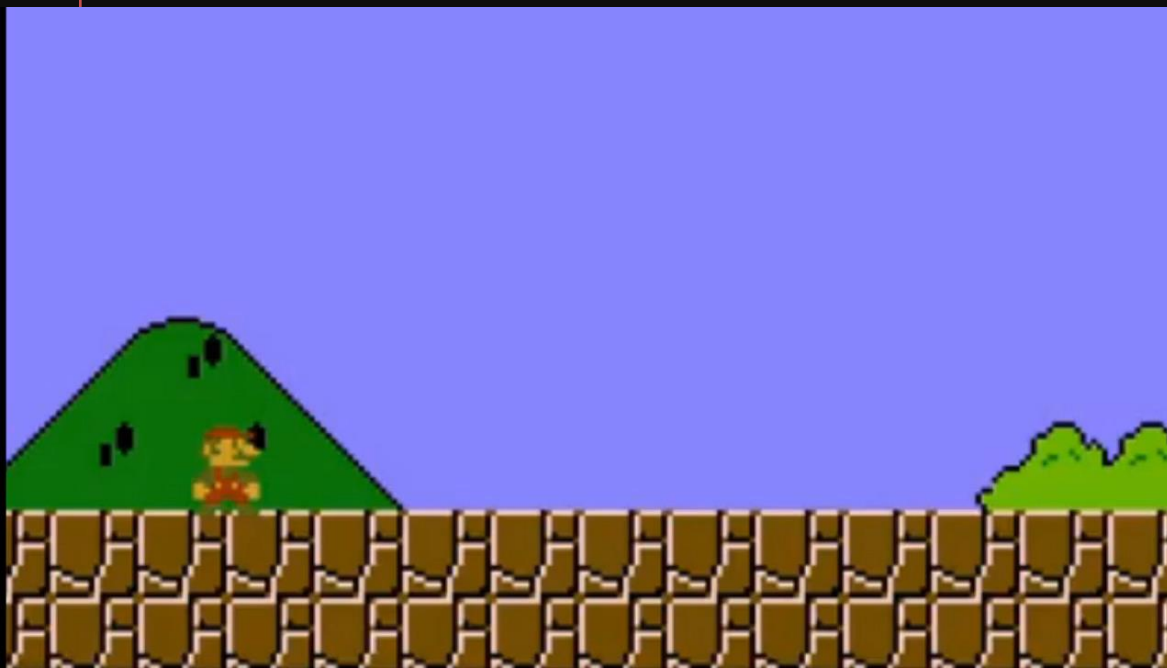
Parameter	Mean	Range
Eye movement time	230 ms	70–700 ms
Decay half-life of visual image storage	200 ms	90–1000 ms
Visual Capacity	17 letters	7–17 letters
Decay half-life of auditory storage	1500 ms	90–3500 ms
Auditory Capacity	5 letters	4.4–6.2 letters
Perceptual processor cycle time	100 ms	50–200 ms
Cognitive processor cycle time	70 ms	25–170 ms
Motor processor cycle time	70 ms	30–100 ms
Effective working memory capacity	7 chunks	5–9 chunks
Pure working memory capacity	3 chunks	2.5–4.2 chunks
Decay half-life of working memory	7 sec	5–226 sec
Decay half-life of 1 chunk working memory	73 sec	73–226 sec
Decay half-life of 3 chunks working memory	7 sec	5–34 sec

Low Level Cognitive Loop for Action Games



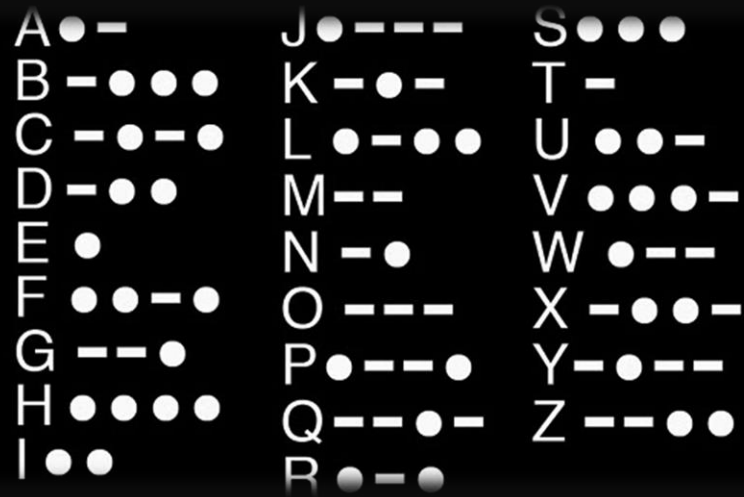
With this model in
mind, let's evaluate a
simple motion model
on a 100% digital
controller





Mario Movement

- Continuous input to drive movement
 - Polling input every 17ms
 - (but on average we only correct every 240ms)
 - Jump Height determined by how long you hold down A button to up to apex
 - Momentum accumulating on left/right on d-pad, different ground vs air
 - Speed modifier on B button
- As you press buttons over time, all of these things stack and combine in very interesting ways



DAC: Input Layering

- Everything is binary (on/off), but the combination of various binary states over time can result in a very analog feeling output
- Binary representations of multiple states, like Binary Numbers or Morse Code – are sequential
- Is the very nature of continuous digital input a sort of DAC?
- Button timing is an inherently analog axis that is binding everything together!

Binary to Decimal

This binary Number... **1 1 1 1 1 1 1 1** Equals this Decimal number

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
-------	-------	-------	-------	-------	-------	-------	-------

128 + 64 + 32 + 16 + + + 2 + 1 = 255

This binary Number... **1 0 0 1 0 1 0 1** Equals this decimal number

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
-------	-------	-------	-------	-------	-------	-------	-------

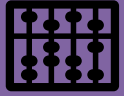
128 + 0 + 0 + 16 + + + 0 + 1 = 149



Well sort of! It depends on what we're converting to.

If the entirety of morse code only mapped to the letter A, that's kind of a DAD.

The expressive output complexity matters!



How do you evaluate what you've created?

Let's build a quick heuristic model to compare and quantify input complexity and expressiveness!



$A > D$

- Think of what is affecting an action as a function of simultaneous input
- Don't need to measure each loop of motor → perception → cognition in milliseconds in a human processor model
 - Maybe more useful to think about it more abstractly and in relative terms
 - Analog input (A) generally takes more load than a simple digital (D) action

3D

1A

Input Evaluation



Horizontal Movement (D)

- Left/Right to add velocity in that direction



Jump (D)

- Press to initiate jump, hold to continue jump



Sprint (D)

- Modifier on top



Timing (A)

- To bind it all together off/on states



1D

2A

Output Expressiveness



Horizontal Velocity (A)

- **Left/Right** to add velocity in that direction



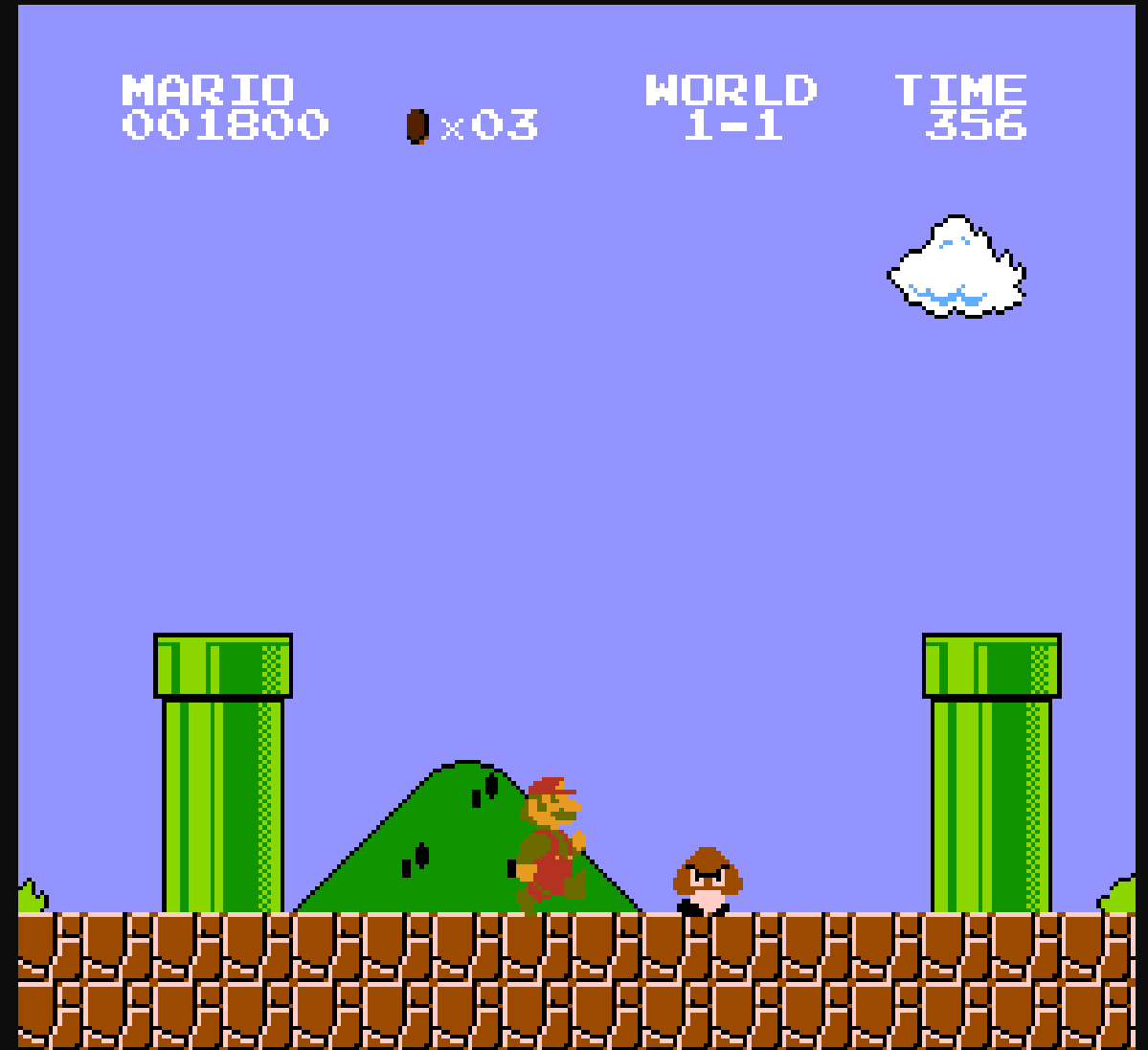
Vertical Velocity (A)

- Higher based on amount of time **Jump** held down



Velocity Magnitude Shift (D)

- Holding/releasing **Sprint** modulates total output



2D
1A

Input: Donkey Kong



Horizontal Movement (D)

- Change direction

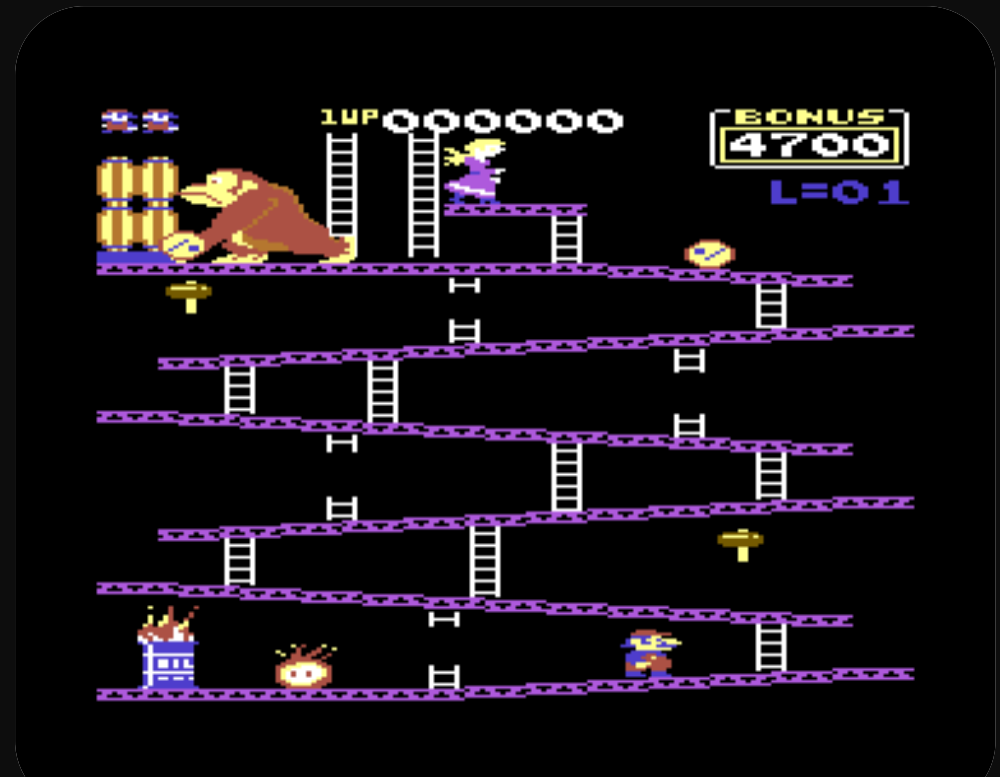


Jump (D)

- Press to initiate jump



Timing (A)



2D

Output: Donkey Kong



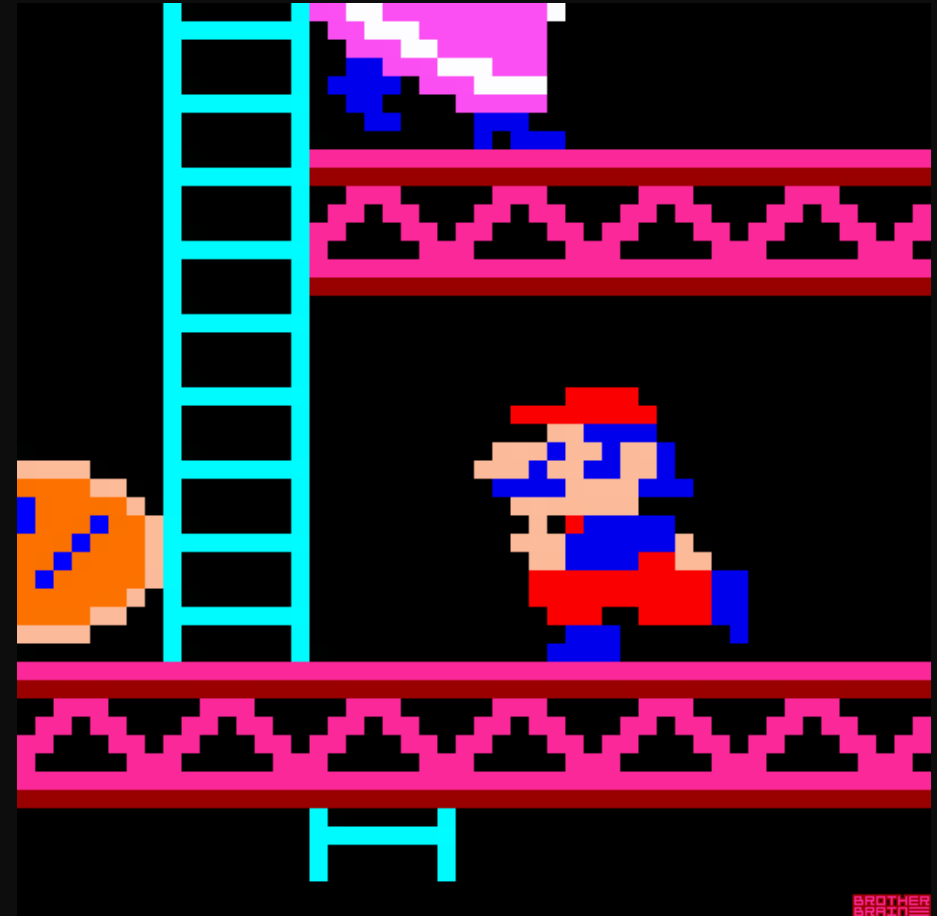
Fixed Horizontal Velocity (D)

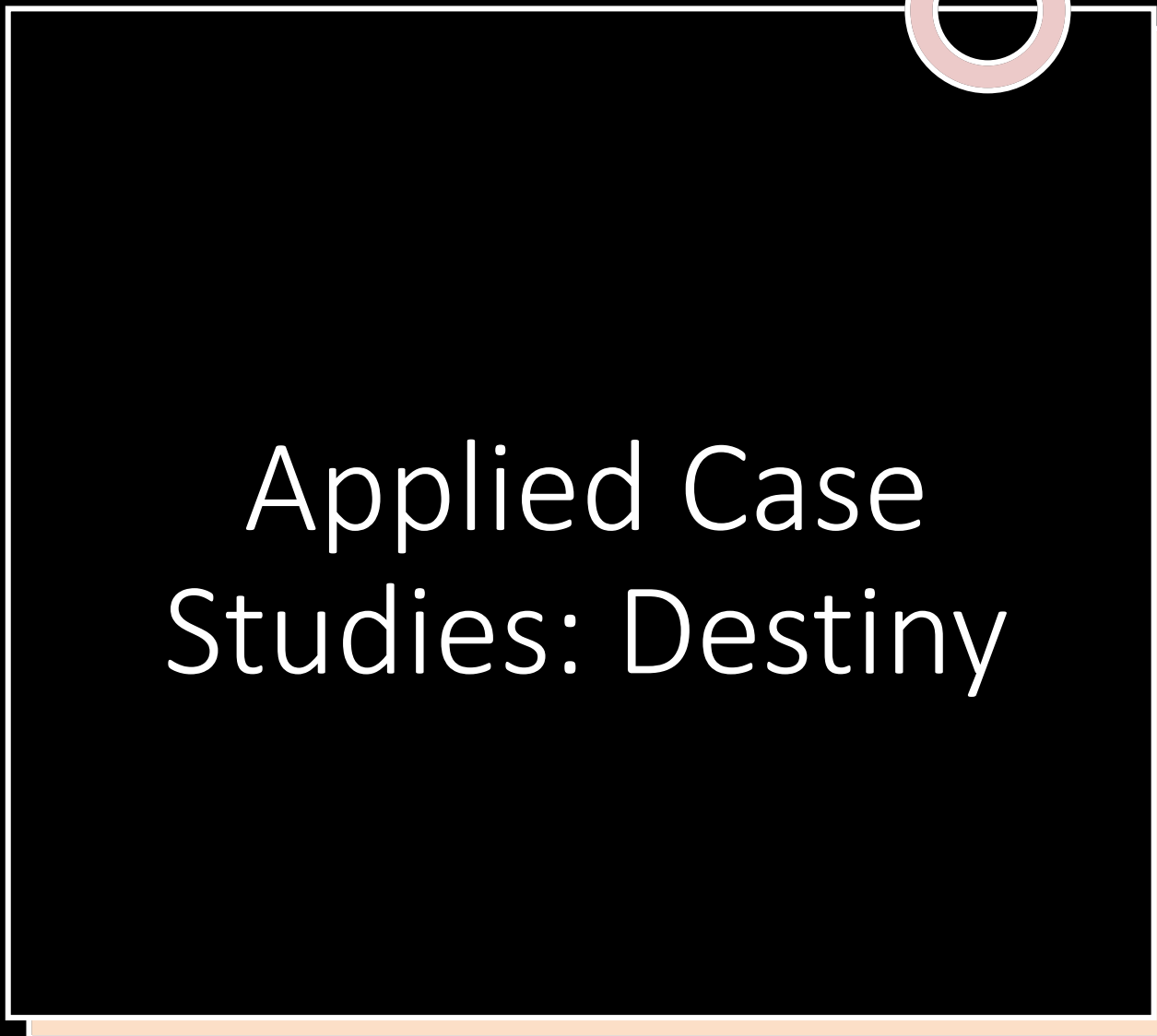
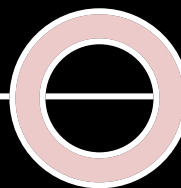
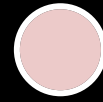
- Pressing **Left/Right** to turn on a dime



Fixed Vertical Velocity (D)

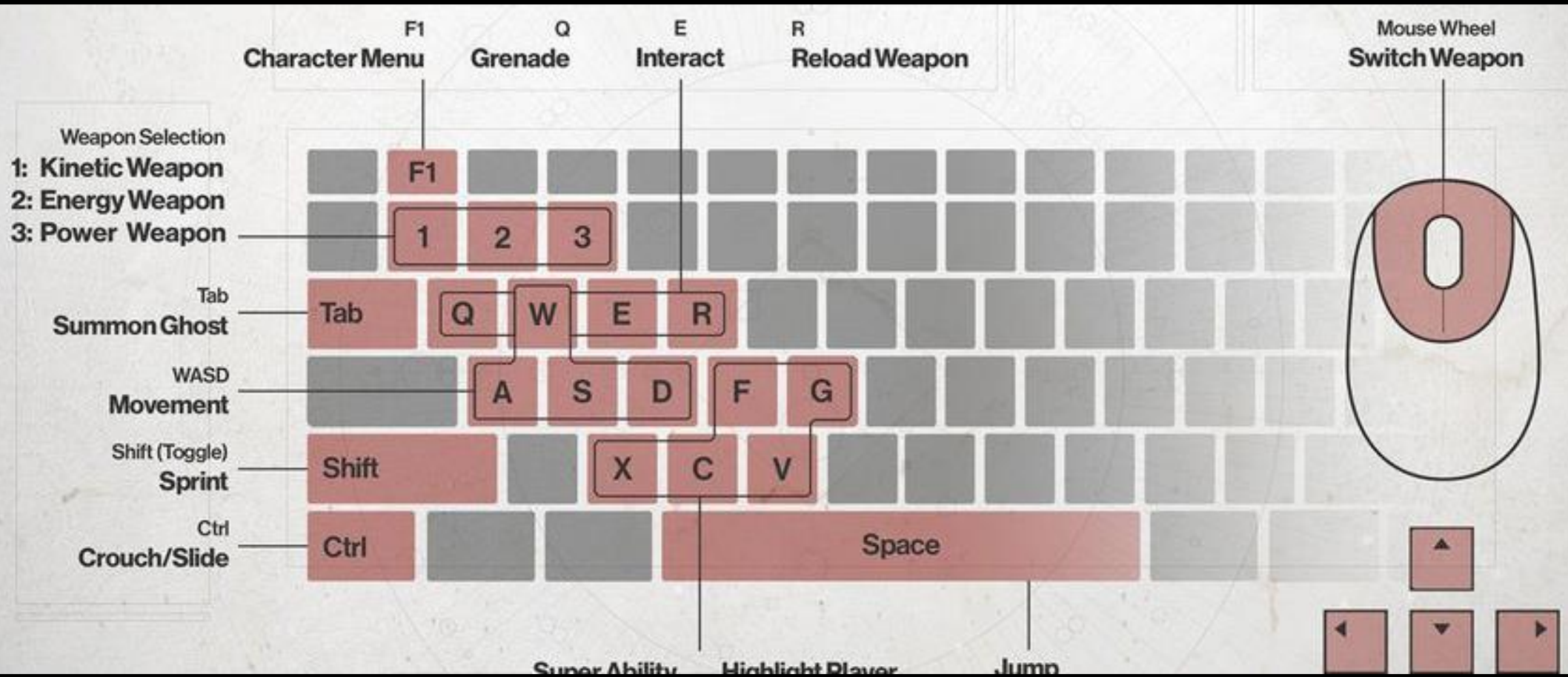
- Pressing **Jump** will do it's thing





Applied Case Studies: Destiny





Analog to Digital to Analog: Destiny PC Controls

- The first versions of keyboard/mouse had instant velocity switching on WASD (the general convention for a lot of PC games)
- But Sandbox wanted movement to still have some analog feel, retain sense of momentum with increased but not instant directional acceleration

4D
1A

Keyboard Input Model



Directional Movement (D)

- Left/Right



Directional Movement (D)

- Forward/Back



Jump (D)

- Higher based on amount of time held down



Sprint (D)

- Modifier on top



Timing (A)

- To bind it all together off/on states



2D
1A

Keyboard V1 Output Expressiveness



Directional Velocity (D)

- Up/Down/Left/Right to set velocity in that direction



Vertical Velocity (A)

- Higher based on amount of time **Jump** is held down



Velocity Magnitude Shift (D)

- Holding/releasing **Sprint** modulates total output



1D

2A

Keyboard V2 Output Expressiveness



Directional Velocity (A)

- Up/Down/Left/Right to accelerate/accumulate velocity in that direction



Vertical Velocity (A)

- Jump height based on amount of time held down



Velocity Magnitude Shift (D)

- Holding/releasing **Sprint** modulates total output





Keyboard



Controller



One double edged sword of velocity accumulation, is that when left uncapped you can kill yourself in a very long corridor.

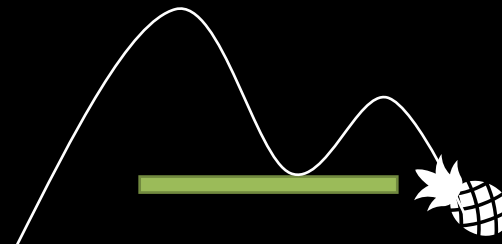
This is a technique our playerbase has named Titan Skating. We have since capped it.



Advanced Case Studies

Grenades





1D
3A

Input



Grenade (D)

- Press to throw grenade



Movement Stick (A)

- Projectile inherits player motion at time of release



Aiming (A)

- Where you aim changes the arc



Timing (A)

- Must predict based on detonation time cued on bounce

4A

Output



Initial Trajectory (A)

- Aim



Bounce (A)

- Higher based on amount angle of aim



Initial Velocity (A)

- Inherited from Movement



Damage (A)

- Based on Distance to Target and final location

4A

Won't that be really hard to process?



? Impulse Properties

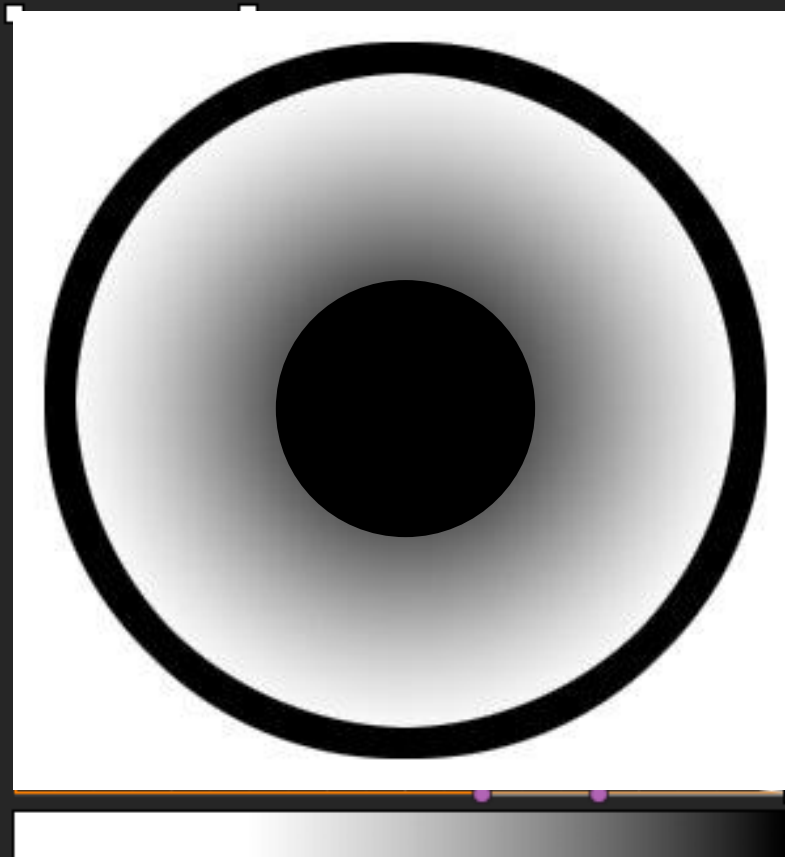
? Impulse speed limit  meters / second

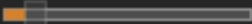
? Maximum high priority object hit...

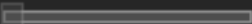
? Maximum low priority object hits...

 Mouse Position -- X: Y:

? Impulse intensity falloff function



Max Splines per Spline2D: 

ADC

The Destiny Bow Archetype



1D
3A

Input



Bow Draw and Release (D)

- Analog input mapped to digital output, we throw out data
- Drives analog system - animation draw time & release



Aiming (A)

- Guaranteed hitscan at full charge



Movement on top to assist fine reticle positioning (A)



Timing Release (A)

- Before we force you to release for overdraw



1D
2A

Output



Damage (D)

- Based on charge time stratified into 3 tiers



Initial Trajectory (A)

- Angle of aim

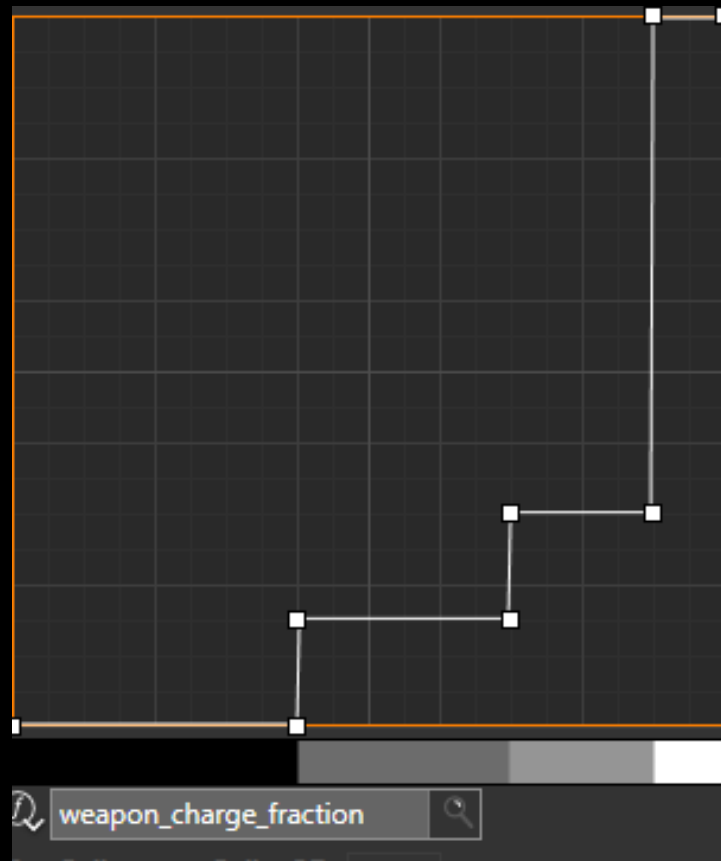
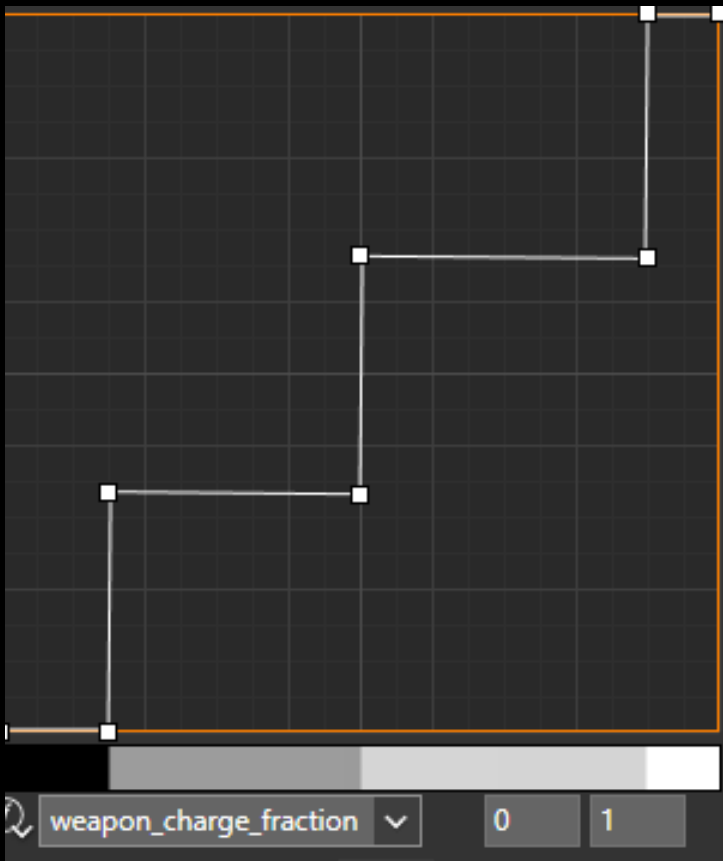


Velocity (A)

- Charge amount & constant gravitational acceleration

Wait let's simplify again!

ADC (charge time -> damage)

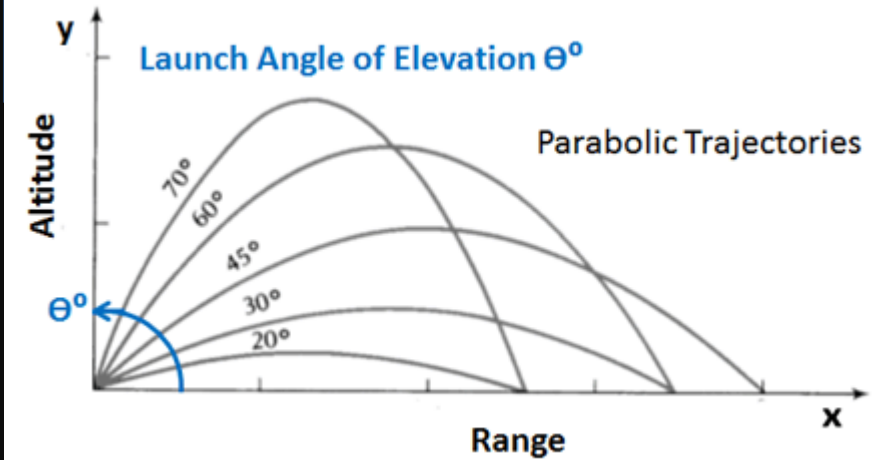


- Charge builds up in an analog way, but is digital in terms of damage & feedback
 - Weak Shot - Special animation & SFX
 - Average Shot
 - Perfect Shot - Bonus VFX/SFX
- Projectile arc (speed vs gravity) is purely analog
 - For accuracy of physical simulation
- In game usage – most people just use the max charge state & perfect shot and ignore the rest

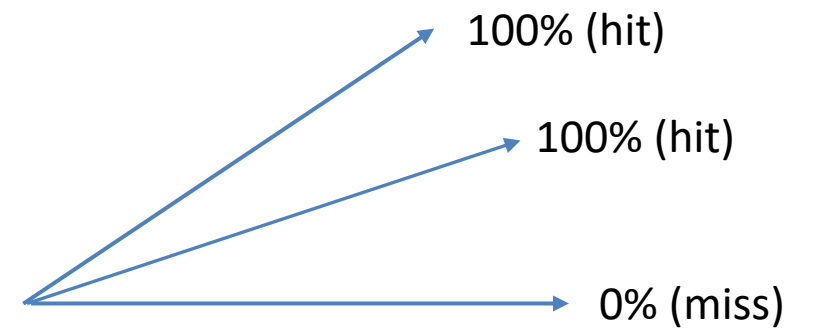
Opt-in Expressiveness



EXPERT CASE

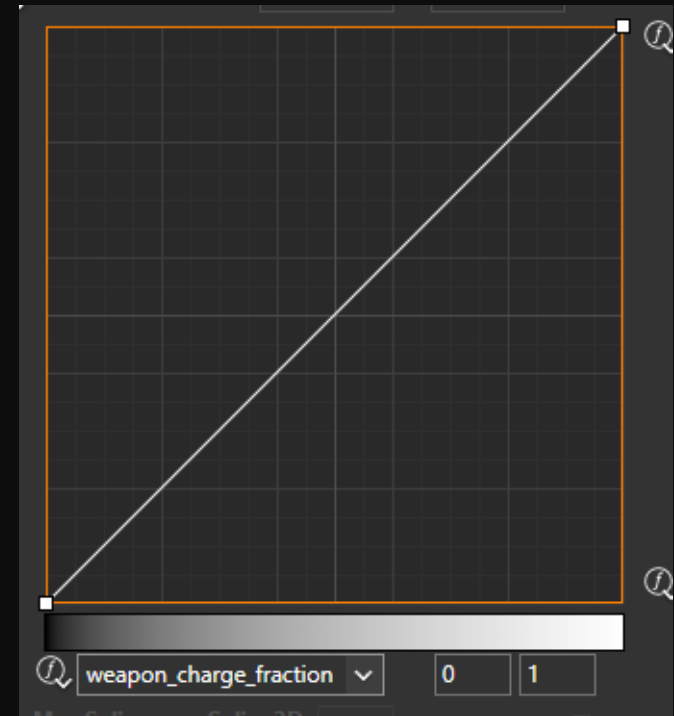


AVERAGE CASE



What if bow damage was more analog?

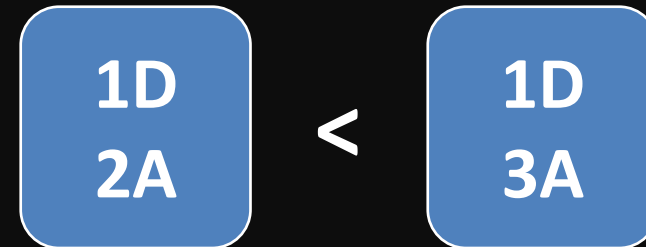
- Would there be more reason to use the in between states?
 - More amorphous success state, more to learn and master
 - In theory interesting -could I visually gauge how much health my opponent has and shortcut TTK slightly?
 - Unlikely unless I am a savant
 - Also harder to balance average case of efficacy, especially in PvP
- Make the decision as a designer
 - Which skills I want the player to master
 - Success criteria for those skills



Damage = Time Drawn (ms)	Shots to Kill Full Health Guardian @ 185 HP	TTK
100	2	2s
93	2	1.84s
62	3	1.86s

What About: Player Acceptance

- This would also steepen the learning curve for a weapon that is already a bit more complex than the everyday weapon
- Base shoot to fire weapon input (auto rifle)



EFFORT vs EFFICACY



- How much does this fight against a player's natural tendency for optimizing for efficiency?
 - Players won't pay 150% effort for 115% benefit
 - Sometimes not even a 150% effort for a 150% benefit
- **How powerful does this weapon need to be to compensate for that?**



Why does
expressiveness
matter?

Sandbox Values

**Sandbox at Bungie
strives to make the
best action experience
in videogames.**

**Mechanical execution
of power fantasy.**

**The controller falls
away.**

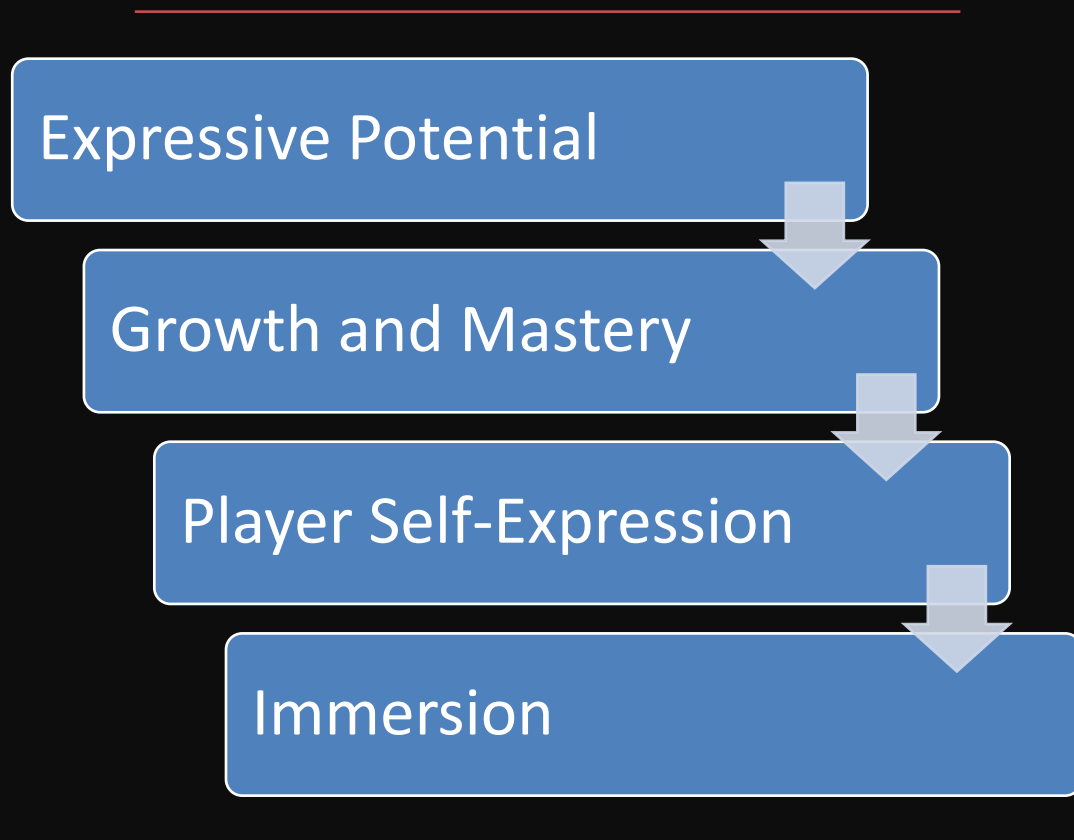
**Content is
communication.**

**Mastery and self-
expression.**

**Universal consistency
in a shared world.**

Believable fantasies.

**Ingredients are
stronger than recipes.**



- There are things to learn and master in this world. Repetition and practice result in improvements to execution.
- Immersive gameplay with tight feedback that prioritizes player intent leads to growth and mastery.
- Growth and mastery lead to self-expression. The game gives all players the same tools, and players find their own style.
- When I can express myself through gameplay I feel like I am a living part of the imagined world.

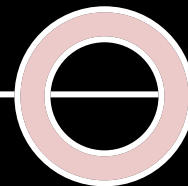
Expressiveness Fosters Innovation



=



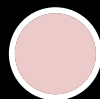
Same energy lol



Thanks for listening!

Go forth and make more
expressive games!

Also please join my team!
Looking for: Art Director, Gameplay
Engineer, User Researcher, and more!



@gregcpeng



BUNGIE®

IS HIRING



GDCEVENT@BUNGIE.COM

#WEAREBUNGIE



SCAN TO
LEARN MORE

Citations & Thanks

- Claude Jerome @Claude_Jerome
 - Presenter, Advanced Game Feel, GDOC
 - https://www.youtube.com/watch?v=EABa8X3IEPM&feature=emb_logo
- Steve Swink
 - Author, Game Feel: A Guide to Virtual Sensation
 - https://www.amazon.com/Game-Feel-Designers-Sensation-Kaufmann/dp/0123743281/ref=sr_1_2?dchild=1&keywords=game+feel&qid=1609877134&sr=8-2
- Grant Mackay @GrantGMackay
 - Feature lead on on Destiny PC SKU
- Jon Weisnewski @sharkeatsman
 - Original wording on Sandbox Pillars
- Human Processor Model
 - https://en.wikipedia.org/wiki/Human_processor_model
- Games Referenced in order of appearance
 - Devil May Cry 4
 - Super Mario 64
 - Super Mario Bros
 - Donkey Kong
 - Quake 3 Arena
 - Super Mario World