

# GDC<sup>®</sup>

## Slime Rancher: A Preemptive Postmortem

Nick Popovich  
CEO, Monomi Park

# Slime Rancher: A Preemptive Postmortem

# Who I am

- ★ Co-founder, CEO Monomi Park
- ★ Game Designer of Slime Rancher, Spiral Knights
- ★ @nickpopovich



# What is Slime Rancher?



- ★ First-person farming/ranching slimes
- ★ Heavy physics simulation, chaotic
- ★ World emphasizes exploration and discovery, non-linear



# Development Data



- ★ 2 yrs dev time before release into Early Access Jan 2016
- ★ 2 person dev team, now 5!
- ★ Made in Unity, uses UFPS, SECTR
- ★ Bootstrapped
- ★ No crunch.



# Retail Rundown

- ★ PC, Mac, Linux, Xbox One
- ★ \$19.99 retail price
- ★ 96%+ aggregate review score
- ★ ~700k units sold so far
- ★ Top 100 Steam game of 2016



# Creating Slime Rancher

# Points to Cover



- ★ The original seed that grew into Slime Rancher
- ★ Our approach to development
- ★ Constraints



# The Birth of an Idea.

You will not stop us. The Order of the Black  
Worm will consume all in its path.







# Development Reality



- ★ Unknown company, 2 people
- ★ Could not build a better mousetrap
- ★ Bigger companies will always win



# Be Better Than Anyone

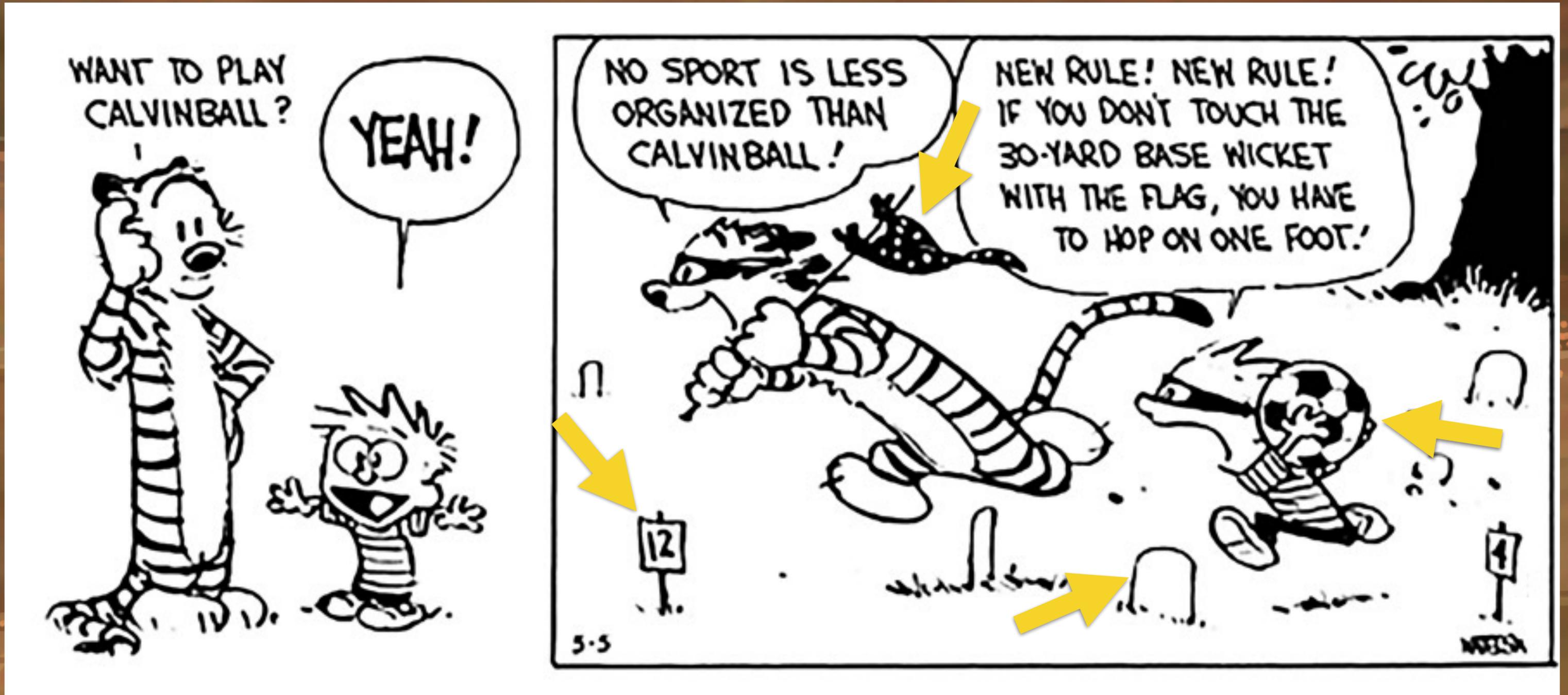


We decided to make a better SLIME RANCHING game than anyone ever. That was something that we COULD DO.

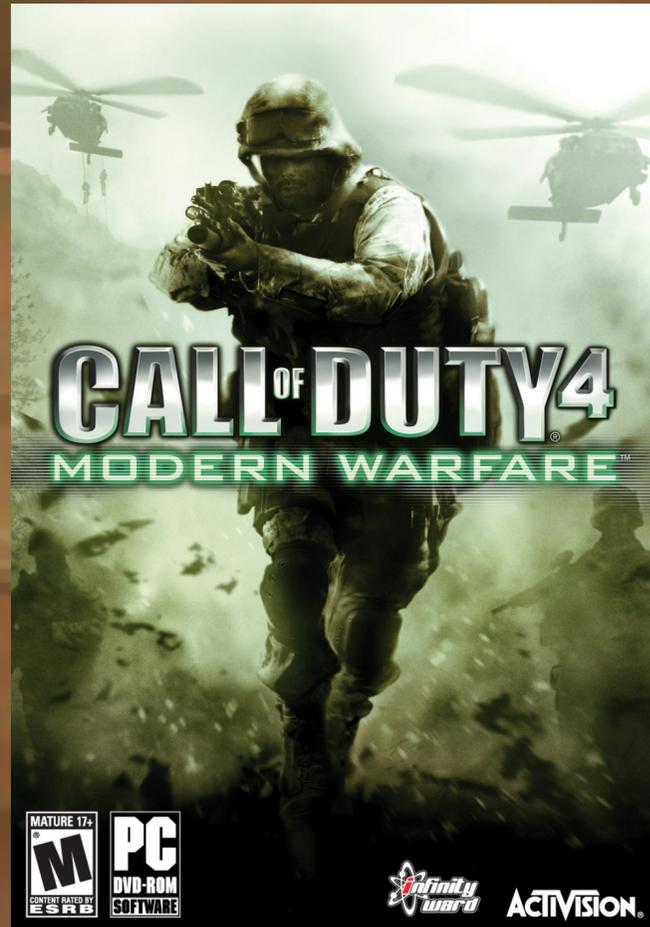
# Make Your Own Calvinball



# Make Your Own Calvinball



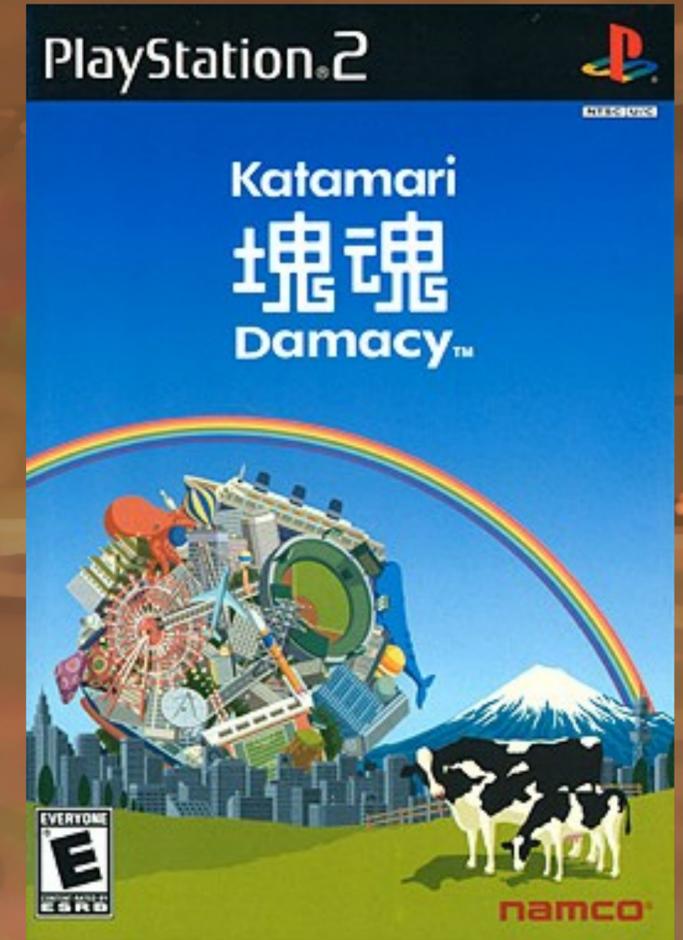
# The Weirdness Spectrum



Familiar



Somewhat familiar



WTF is this magic?

# Use Your Constraints



- ★ Can't animate, so limbless slimes
- ★ FPS = no character model
- ★ FPS = mainstream muscle memory
- ★ Open world because scripting is hard
- ★ Farming theme because it was relatable and the rest was weird as hell



# Takeaway



Slime Rancher is a weird concept played in a familiar way, wrapped in a familiar theme. Two of three battles are already won.

And the last battle is a game of Calvinball.



# Building Guarantees into Slime Rancher's Design

# Points to Cover



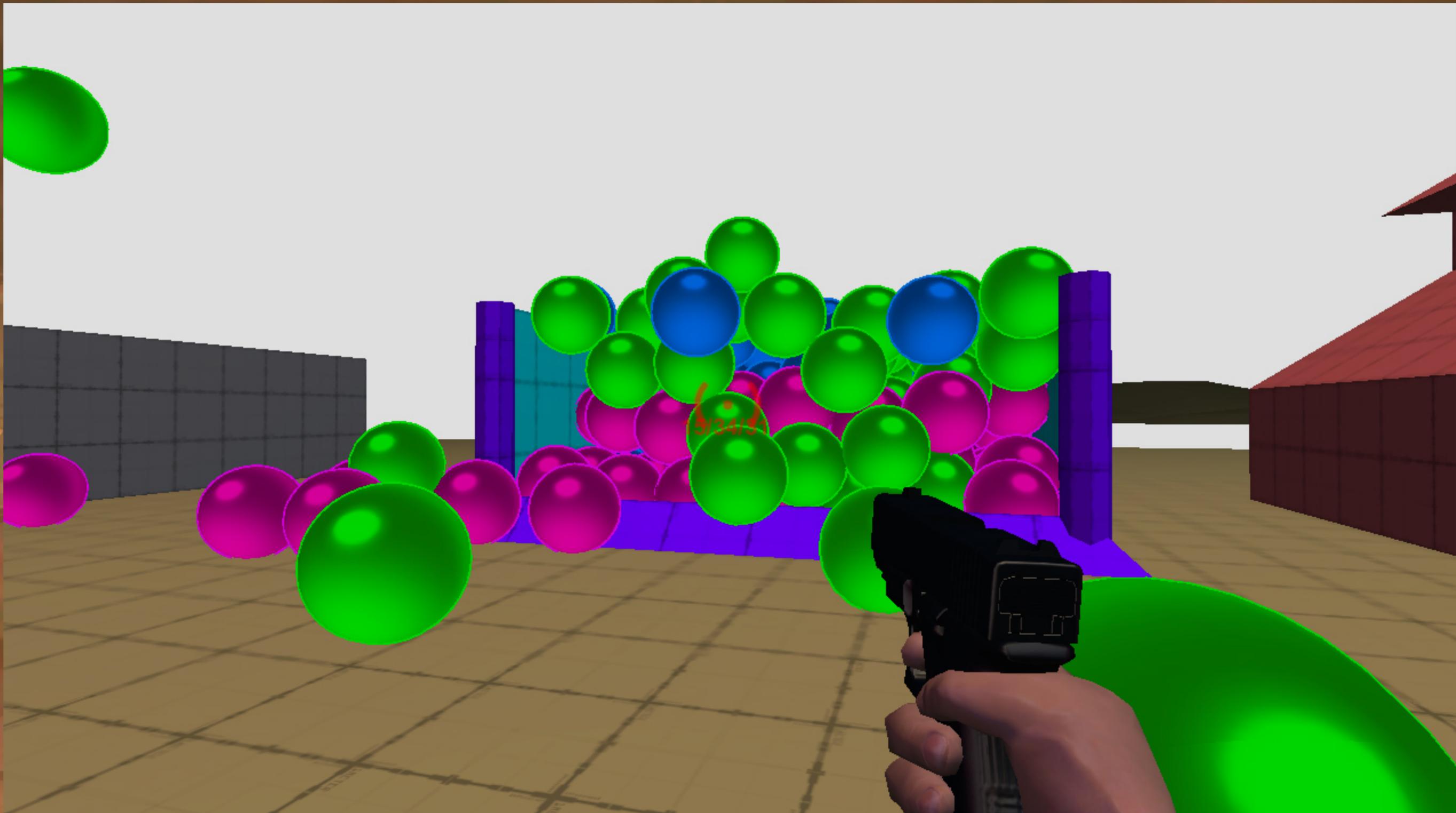
- ★ Our approach to prototyping a game and how we evaluate it
- ★ Building fun loops, the funnest of all known loops
- ★ How this all guarantees a strong foundation for a the final product



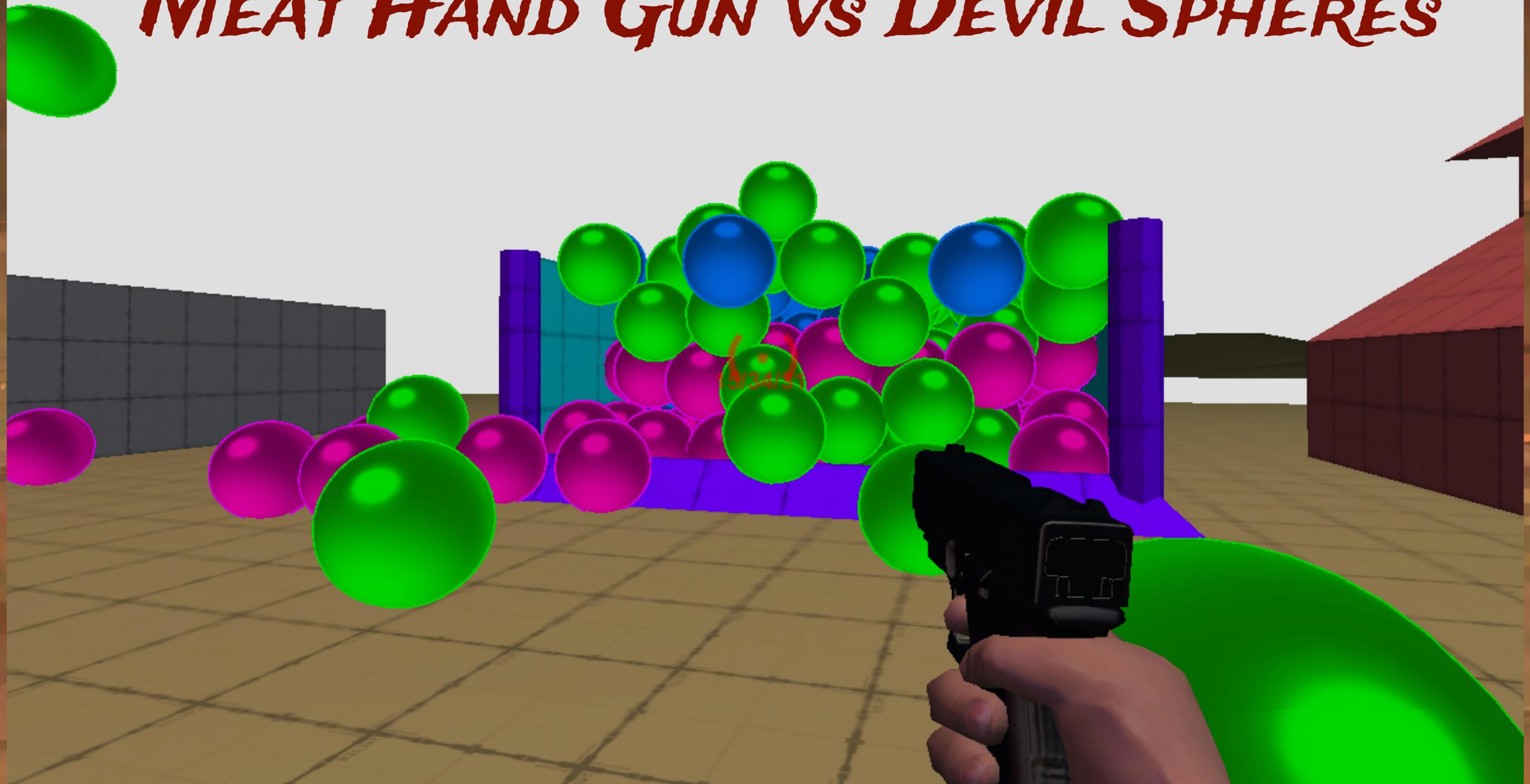


Health:    
Green Arrow:  

1	2	3	4
			
23	86	0	50



# MEAT HAND GUN VS DEVIL SPHERES



Score: 0  
12:51



Heart icon: 100  
Arrow icon: 100

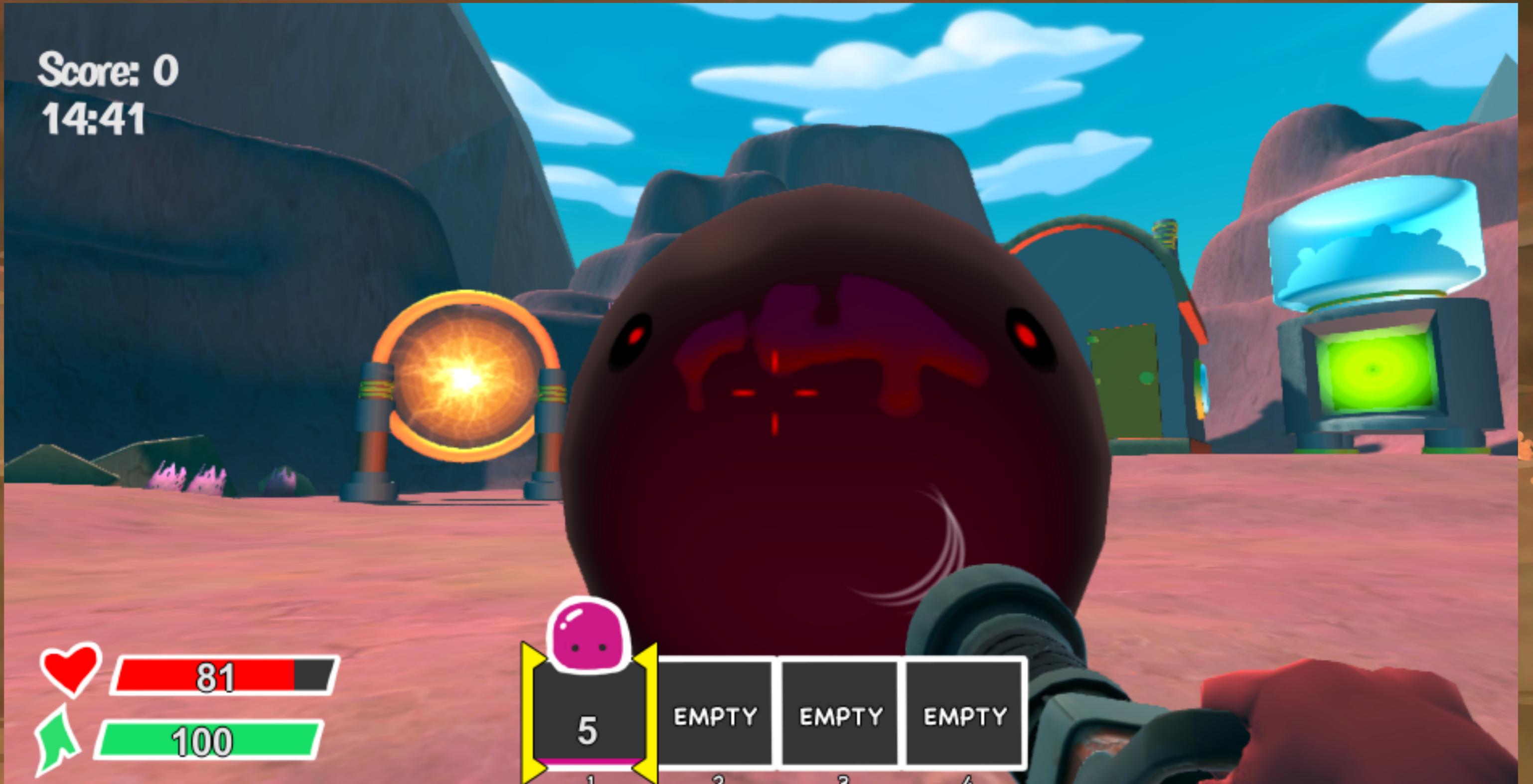
			
5	1	EMPTY	EMPTY
1	2	3	4

# Slime Rancher

Score: 0  
14:49



Score: 0  
14:41



# Prototyping Approach

- ★ Our prototype focused entirely on the core game loop, not a 'vertical slice'
- ★ Prototype loop and final game's loop are almost exactly the same



# Why This Approach?

- ★ Almost every game has a 90% loop: the thing you do 90% of the time
- ★ This loop is the machine that makes players want to PLAY and REPLAY your game
- ★ It is not perceived as the experience as a whole, it is what drives the experience

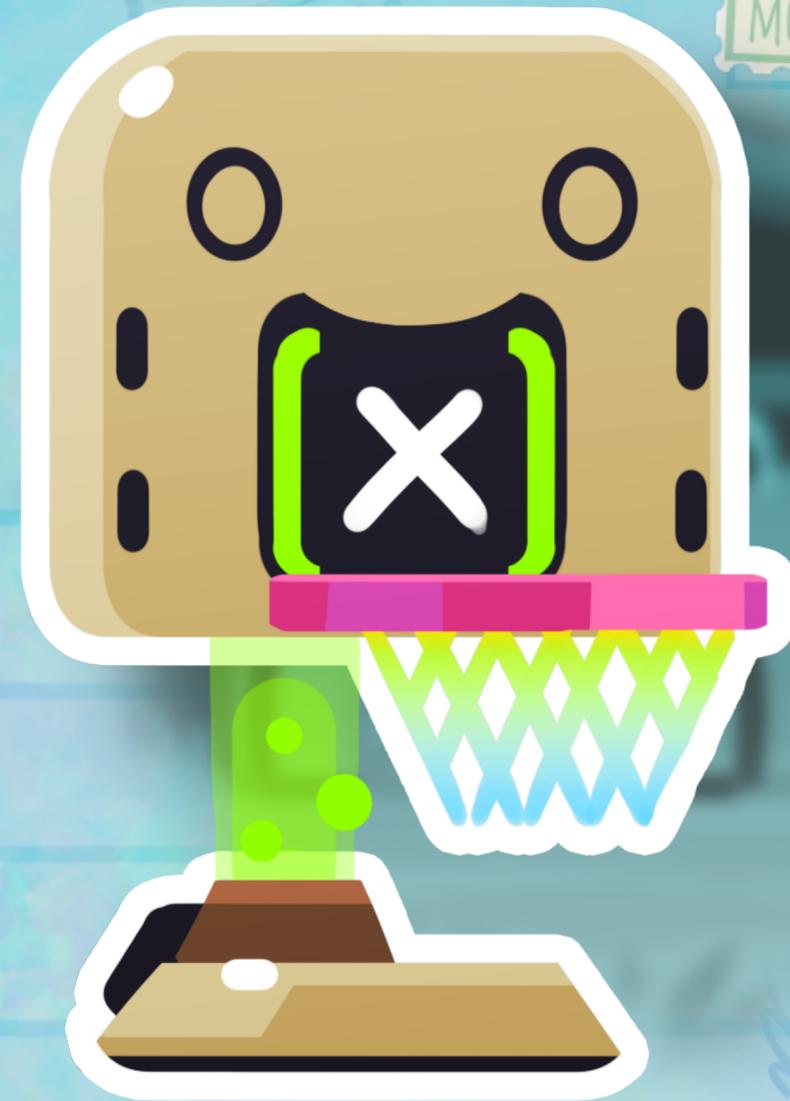




# It's in All Forms of Play



- ★ While choosing teams in basketball you bounce the ball
- ★ Between turns in a board game you can't wait to roll the die
- ★ All of these things are the parts of the game that FEEL GOOD



# Testing Approach

- ★ Presented the prototype to testers as a mostly finished game (it wasn't)
- ★ Asked to evaluate only what they see
- ★ 'Would you buy it?'



# Why This Approach?



- ★ You can't hide behind cool features that aren't yet implemented
- ★ You can't trust your friend's feedback (especially devs)
- ★ If a success, more cool features just makes it better because the foundation is solid



# Takeaway



Build a prototype that proves whatever the player is doing 90% of the time is fun.

Prove this by testing it for what it is, not for what you hope it will become.

The fun loop should be fun on its own.

# And Now A Quick Biology Lesson

(I promise this is relevant to Slime Rancher's development.)



This is a great white shark.

# Cool Shark Facts:

- Rows of razor-sharp teeth, rad!
- Can detect 1 drop of blood in 1 million drops of water, whoa!
- Few evolutionary changes in like almost forever
- Even their dorsal fins are scary.





Summary: Every part of the shark's design makes it better at being a shark.



This is a pug.

# Stupid Pug Facts:

- Has trouble breathing, rad!
- Can't give birth without human intervention, cool!
- Eyes can literally pop out of head. Srsly.
- Known to wear dumb costumes even in the 16th century.





Summary: Loaded with features that hinder its ability to be a successful dog.

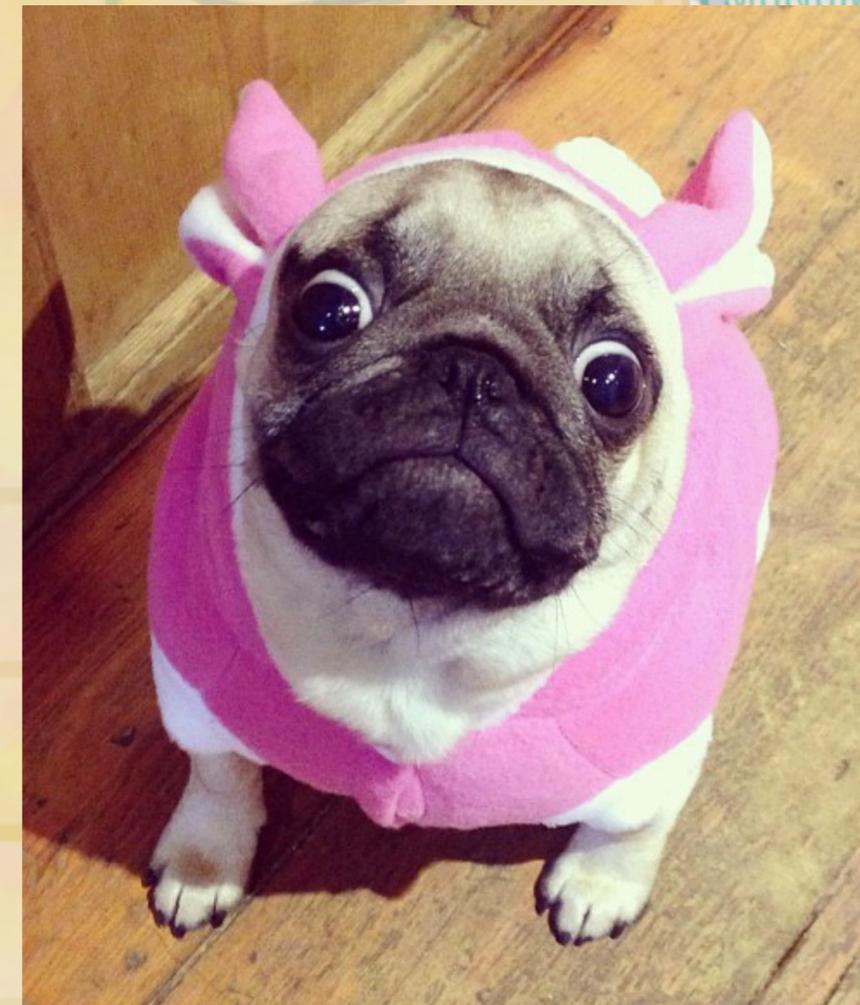
# Lean Game Design

- ★ Every aspect of SR's original design informed the core gameplay
- ★ It created a very focused, confident game right off the bat
- ★ Plus, it is way cheaper to develop



# Pug Features We Ditched

- ★ Randomly generated world
- ★ Procedurally generated vegetation
- ★ Very open world, vehicle required
- ★ Dozens of unique tarr (enemies)
- ★ Even more enemies because why not



# Takeaway



Make your game a shark: every feature informs the core experience and makes it a killing machine.

Cut your crafting systems, multiplayer, or whatever is nice-to-have until your game matures.



# The Release and Our First Year

# Points to Cover



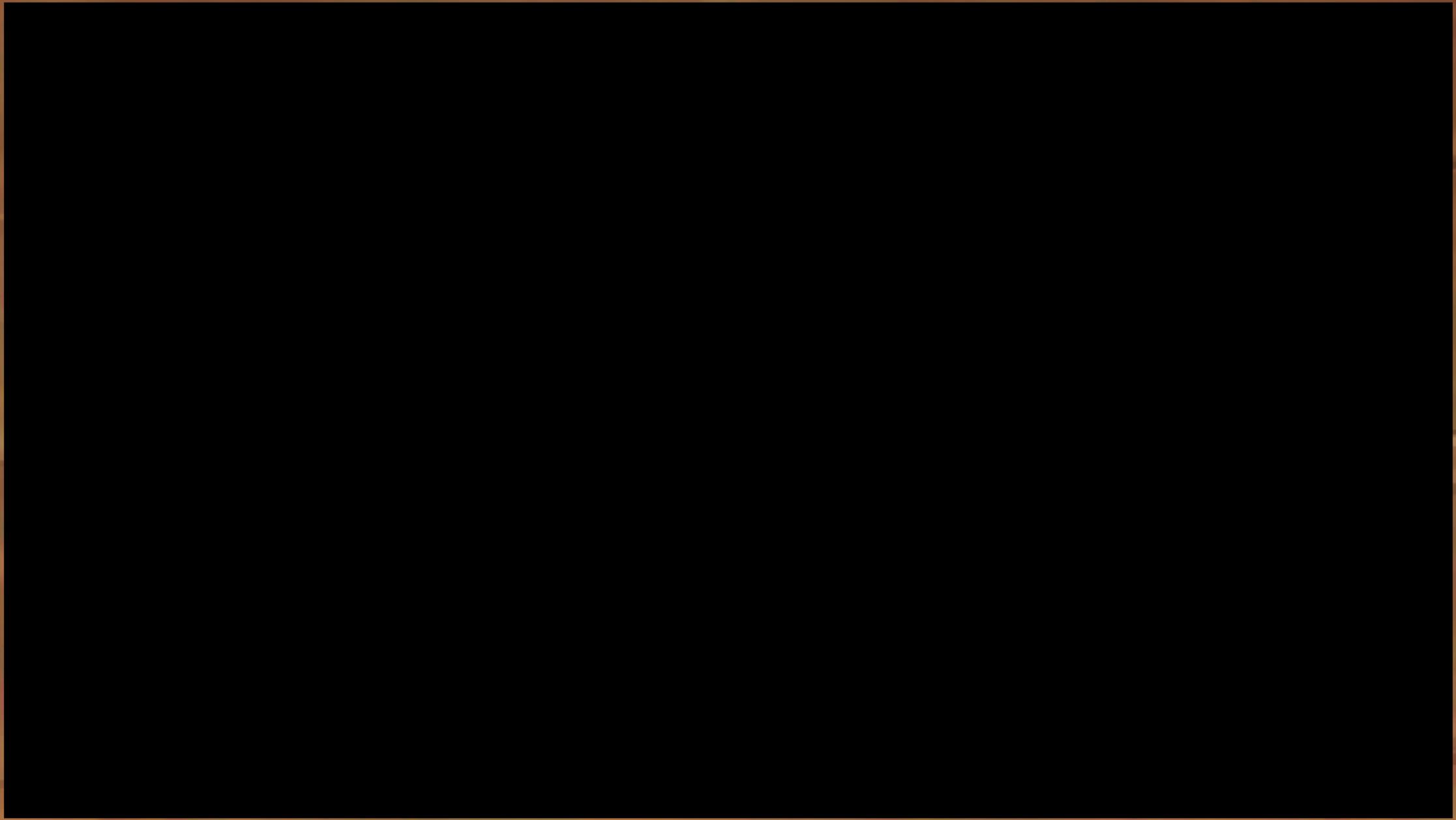
- ★ Pre-release strategy
- ★ Early Access plans and how they got thrown out
- ★ How Early Access actually works



# Pre-Release

- ★ Released v0.2.0 to friends, family, and some strangers (knew it would leak)
- ★ Small file size allowed people to share via email (shareware style)
- ★ Suddenly: Youtube
- ★ Moved right to Greenlight to utilize the momentum, used a YT video





# Early Access Plans

- ★ Outlined release plans: small updates, and big ones worth coming back to play
- ★ Expected players would come and go from SR: told them to 'walk away'
- ★ Small team, updates would take time
- ★ All of this hinged on players understanding development: **HUGE MISTAKE**



# Early Access Reality



- ★ Players complained if updates were too infrequent, or too small
- ★ Youtubers needed a clear signal
- ★ Realized that we needed a new strategy
- ★ Turned to our MMO dev experience



# New Approach

- ★ Treated Early Access dev like running a released, live game
- ★ Big updates became like content updates/dlc, had big effect on sales, Youtube
- ★ This was all mostly a change in messaging, which is a lot of work
- ★ Changed 'waiting' into 'hype'



# Provocative Takeaway!



Early Access doesn't work.

(at least, not as intended.)

# EA: What You Might Expect

- ★ Players join devs on a magical journey in game development!
- ★ There are ups and downs, but that's ok, because it's a game **IN DEVELOPMENT!**
- ★ Devs can change their minds about stuff during development because it's normal!



# EA: The Cold Reality



- ★ Players expect regular content updates, yesterday
- ★ Early Access means player wishlist features might still happen
- ★ 'I understand it's Early Access but...'
- ★ Players dislike changes, however small





Pink



Temp Puddle



New Puddle

# Takeaway



Releasing into Early Access is releasing your game, period.

An Early Access game should be treated like a live game with content updates that drive excitement. This might mean changing dev style.

Excitement = more Youtube videos, more sales



# Quick Review Roundup

# Positive Reviews

- ★ Slime animations and personality
- ★ Exploration and discovery-based learning, no hand-holding
- ★ It's happy. Most games aren't happy.
- ★ 'Chill' or 'zen' game



# Negative Reviews

- ★ \$20 price tag, 'should be \$15'
- ★ Lack of content (avg play time is about 12 hrs)
- ★ Can't run on their system
- ★ No multiplayer



# One More Thing on EA Reviews



If you made a good game, most of your negative reviews and criticisms will be a result of your being in Early Access.

Those reviews will likely not go away after leaving EA.

# Youtube! What? How?



# Points to Cover



- ★ Slime Rancher's Youtube success
- ★ Why Slime Rancher works on Youtube
- ★ What else works on Youtube



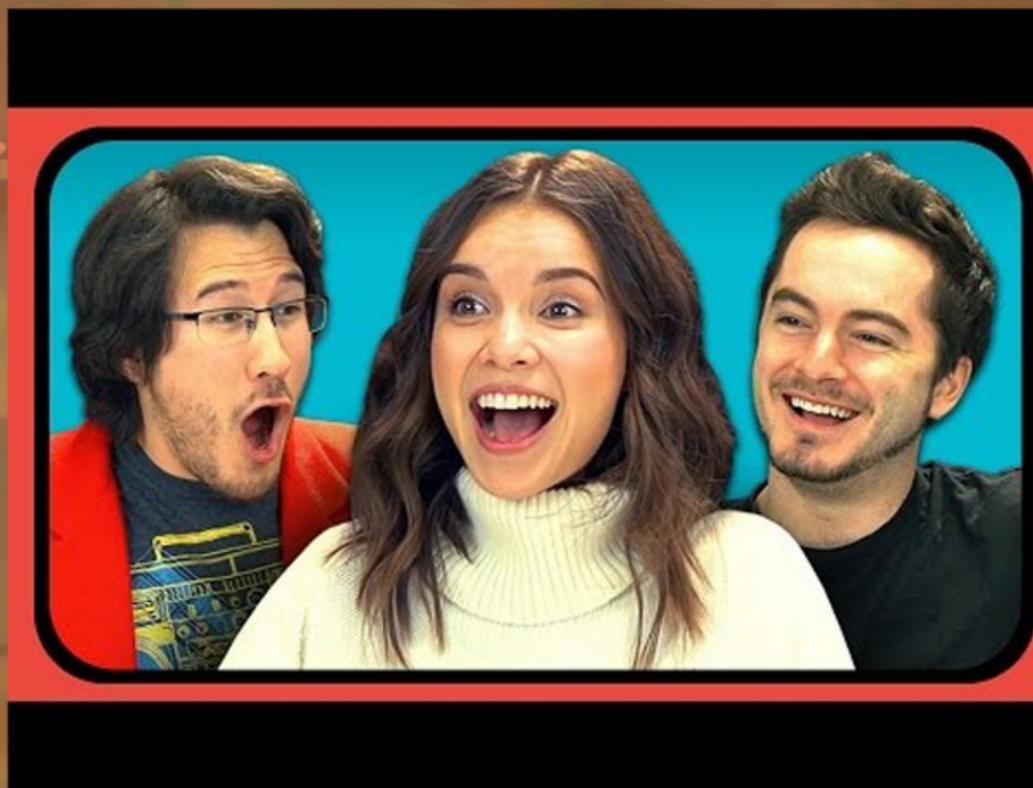
YouTube

twitch

+



= Free Marketing



# Slime Rancher on Youtube



- ★ Hit on YT: over 1mil views per week
- ★ YT is our marketing: no shows, no PR
- ★ We don't buy YT video coverage



# Why SR Works on Youtube



- ★ Charisma: slimes have personalities, fun to watch
- ★ Action happens toward the camera
- ★ Discovery-based learning: audience observations feel rewarding
- ★ Lots of unpredictable events keeps it fresh
- ★ Customization features make videos unique



# What Else Works on Youtube

- ★ Games that allow you to start 'projects'
- ★ Games that leave the audience with a 'question' to answer: survival, exploring the unknown, etc
- ★ Anything that prompts interaction w/audience
- ★ Games that leave something for the audience to try on their own



# Additional Youtube Thoughts



- ★ I would not consider developing a game that's not YT-friendly. It's that important!
- ★ Make it easy for people to request keys: keymailer.co, promotor app, distribute ( )
- ★ Games move up the YT food chain: start small
- ★ Videos get about 10% views to subs in 3 days



# Takeaway



Youtube and streaming is the best way to market your game.

Being Youtube-friendly will give your game a much higher chance of success.



# In Summary...

# Be Better Than Anyone

- ★ Don't be limited by your constraints, embrace them
- ★ Don't compete with established games/genres, but use their muscle memory
- ★ Invent your own Calvinball



# Prototyped for Success



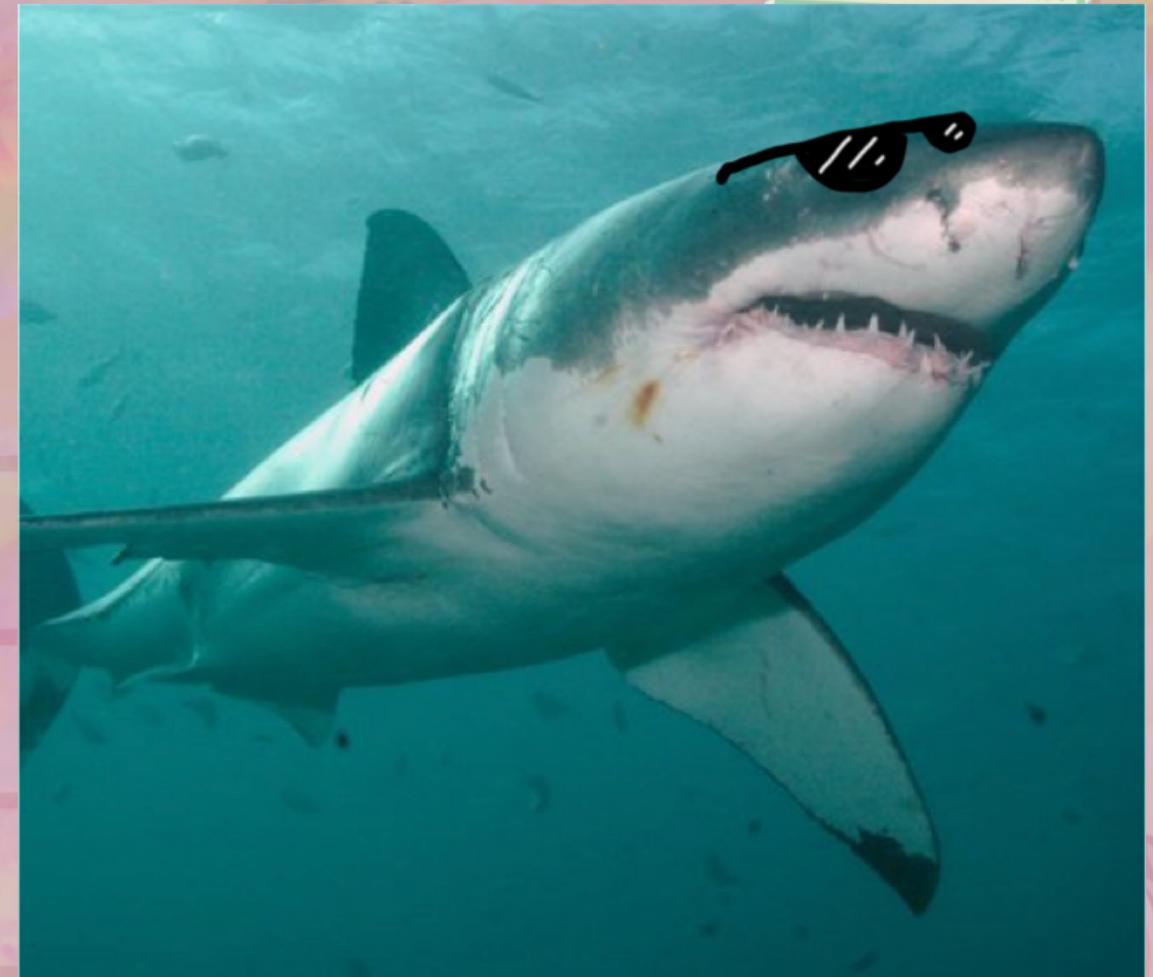
- ★ Focused entirely on the 90% loop
- ★ Tested SR as a 'complete game'
- ★ Allowed us to be very confident in our design early on



# Make Your Game a Shark



- ★ Shark features inform the core design
- ★ Pug features aren't nice-to-have but not essential and bloat your game
- ★ Shark design is more focused and cheaper to develop!



# Early Access = Live Game

- ★ Players expect your game to update regularly and not change too much
- ★ Build updates around exciting stuff that generates hype and Youtube videos
- ★ This may require restructuring how you would otherwise develop your game



# Youtube and Twitch

- ★ There are qualities and game designs that are Youtube-friendly
- ★ Audiences want to feel engaged, not just a spectator
- ★ Make it easy to request keys, but just give them away to start



# Thank You!

(now time for some Q & A)



[www.monomipark.com](http://www.monomipark.com)

@monomipark

@nickpopovich

