

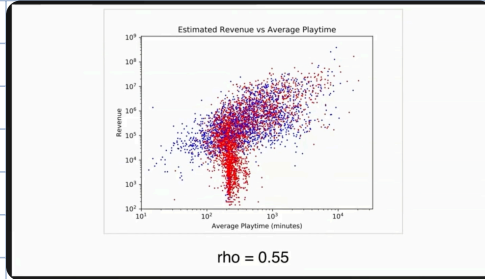
Self Learning Note

Game Industry

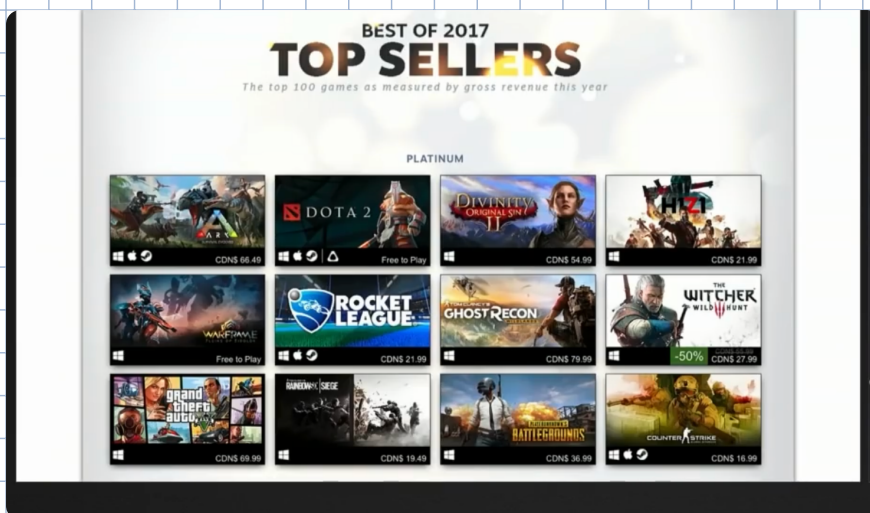
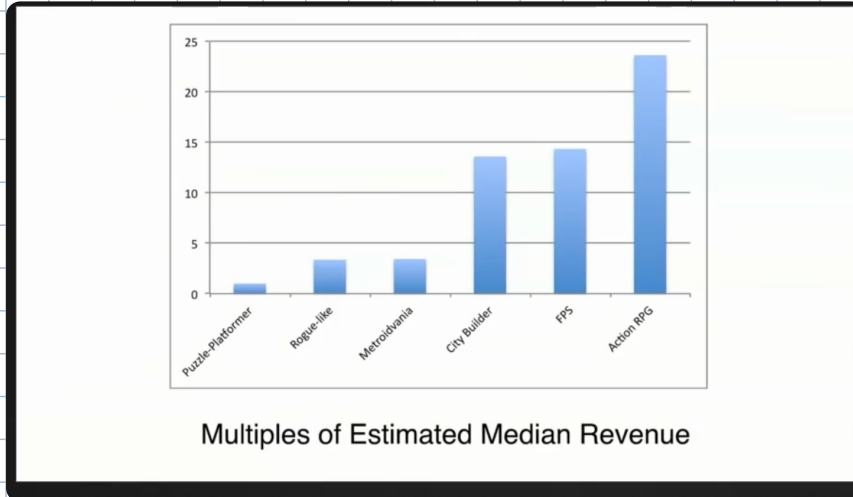
• Revenue

— Element that impact revenue

- Played Time



◦ Game Genre



◦ Streaming & Player Response

Streaming – Player Response

- YouTube and Twitch are huge
- But value isn't universal

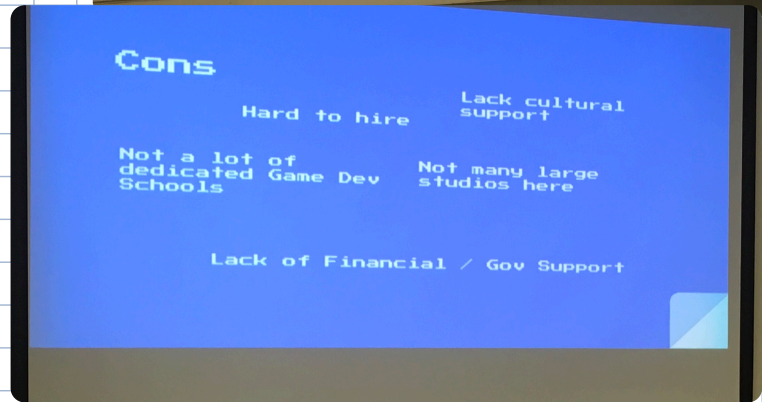
◦ Player Community



◦ Target Player

What About Targeting a Niche?

- Yes, find an audience that wants what you are making!
- Be sure it is big enough to meet your commercial goals
- Is your niche saturated?



✳ Notes from Tencents

立口座BB

1. 市場重方向

- 科技發展
- 人的變化
- 人的習慣

2. 目標群骨豐

- 規模
- 市場 (消費力)

3. 與開發的橋梁

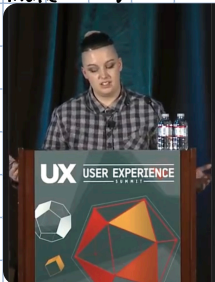
- 反饋給開發團隊

Game Design

• User Experience

- Design Mindset

- Care about player experience
- Value player
- Make Player feel good



Conclusion

- Our player's brain in a mindset of play is more likely to overlook little lies to make themselves feel very good and positive things
- We want to ensure they know that while game designers craft the experience for them, we do so because we care about them more than anything else in our jobs.
- The very essence of our job is to bring things to players they didn't even know they wanted

...that's why we all have probably fallen in love with games in the first place.



Player Experience of Need Satisfaction (PENS)

- This could be a whole 3hr lecture on its own
 - Suggested Reading
 - Essential understanding of player motivations
- Primarily affects the **competency** element
- Allows players to feel **mastery** over the game systems
 - Skill
 - Experience (not XP!)
- A powerful **intrinsic** motivator
 - Intrinsic vs Extrinsic
- The concept of "optimal challenge" is similar to "Flow"

But Why ?

- The Golden Rule
 - Games are products designed for a specific audience
- Were trying to deliver an **experience**
 - Players won't get the full experience if they drop off
- The goal is ***not*** to punish the player
 - 80% of people don't complete games
 - Steam Achievement Research your favourite games
- Spending money on content is a waste if players never see it
 - Might as well not exist

How Do Developers Increase Difficulty ?

- Define the **axis** against which the players skill is challenged
 - Will need to get pretty low level
 - Think about the second to second mechanics and inputs
- **Example: Racing games**
 - Ultimately handily tricky to navigate tracks at speed
 - Variables
 - Speed (fast is bad - downhill before a sharp turn)
 - Friction (rain reduces grip)
 - Corner Severity (many sharp corners back to back)
 - Track Familiarity (mirror the track, dynamic hazards)
 - Blockers (other cars prevent ideal racing line)

How Do Developers Increase Difficulty ?

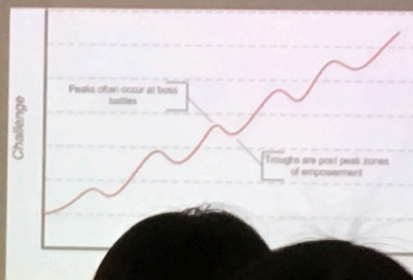
- **Escalate existing challenge**
 - Enemy Health
 - Enemy Damage
 - Enemy Speed
- **Introduce new challenges**
 - New Enemies
 - New Enemy Behaviour
- **Disrupt player**
 - Remove functionality from tool box

Mitigating Difficulty

- There are a number of ways:
 - Player Skill
 - Familiarity of game systems
 - Hero progression (hp+)
 - Better loot / gear
 - New skills and abilities
- Be careful to not to make the player too powerful!
 - Fun at first but gets boring

Difficulty Curves

- A useful way of seeing the game difficulty progression at a glance
- Challenge vs Time
- Different curves for different game types



Example - Legend of Zelda: Breath of the Wild

- Start with Simple Red Bokoblin
 - Learn how to beat
 - Familiarise with the enemy
- Introduce Blue Bokoblin
 - Same Core behaviour (speed, attack patterns, attack range, rate of fire)
 - Increase the health - tougher!
 - Change the weapon - does more damage!
 - Can kick away bombs - removes a player attack
- Introduce Lizalfos
 - Whole new attack pattern!
 - So much faster!
 - And he can jump!



- Juice

- Definition
 - Polish the game
- Elements
 - VFX
 - SFX
 - BGM

- Animation
 - Self Animation
 - React Animation
- Characterize Game Objects
- Screen Effect
 - Camera Effect
 - Fog
 - Color
 - Reverse
 - Shake
 - Blur
 - Etc....