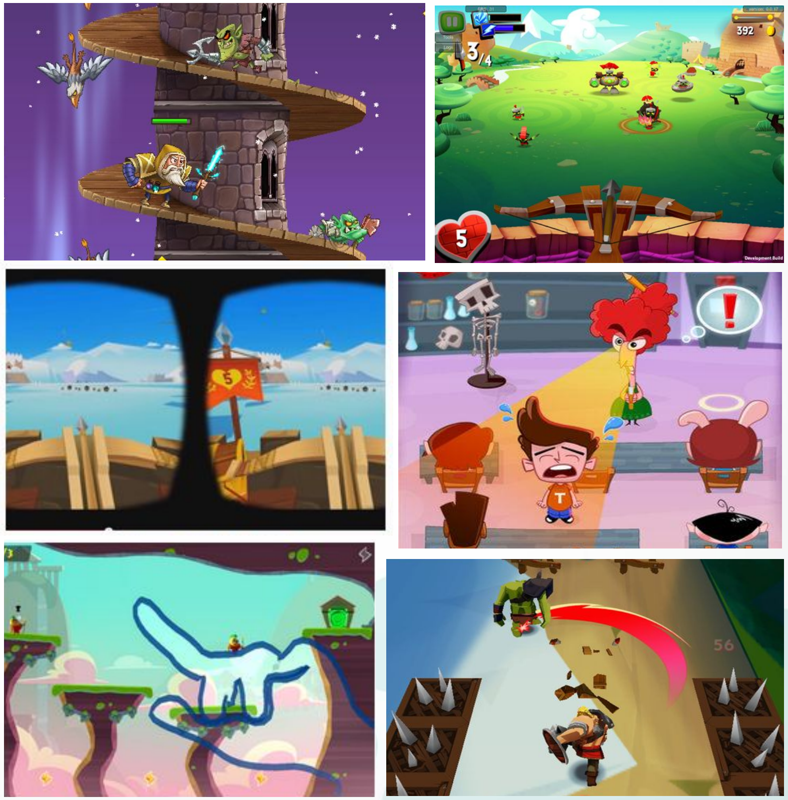
Practical Level Design

Elad Drory

Oh, hi

My name is Elad Drory. I'm a dude who makes games.

Dude



Games

What is this???   D:

Relax. Here's what you can expect:

- Level design - How to start
- Teaching and hooking
- Classic flow vs F2P flow
- A dash of metagame
- Applying this to endless games
- Designing tools and level editors

What kind of games are we talking?

Each game is a unique little snowflake, but we're going to focus on:

Level-based casual games

&

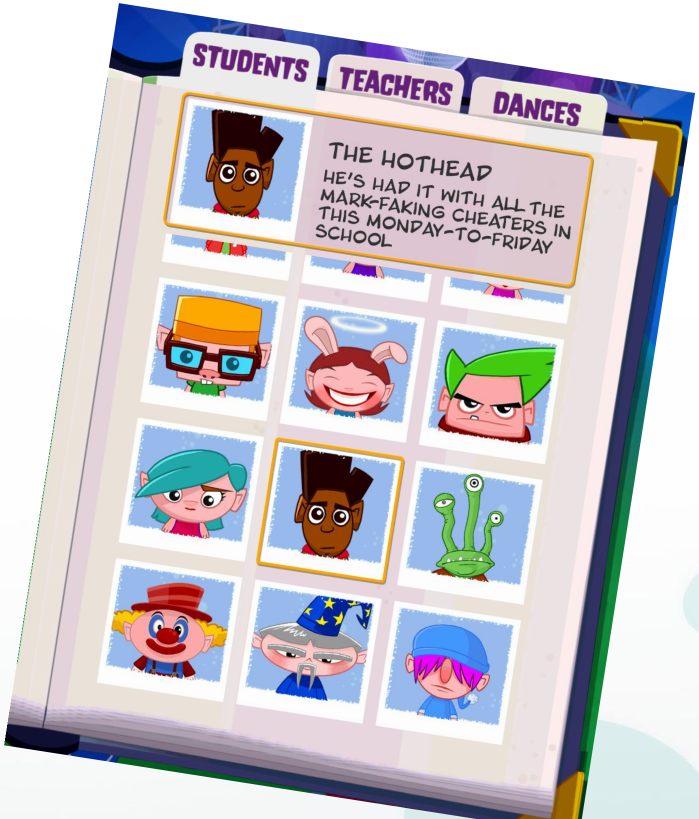
Endless casual games

Sketch out the plan

(I’m gonna assume you have a theme and core mechanic)

First: Brainstorm features

- Go wild. Good features:
  - Tie in with the theme and core mechanic
  - Add a random element
  - Make the player change their playstyle and break monotony
  - Can be combined together to create cool reactions and emergence

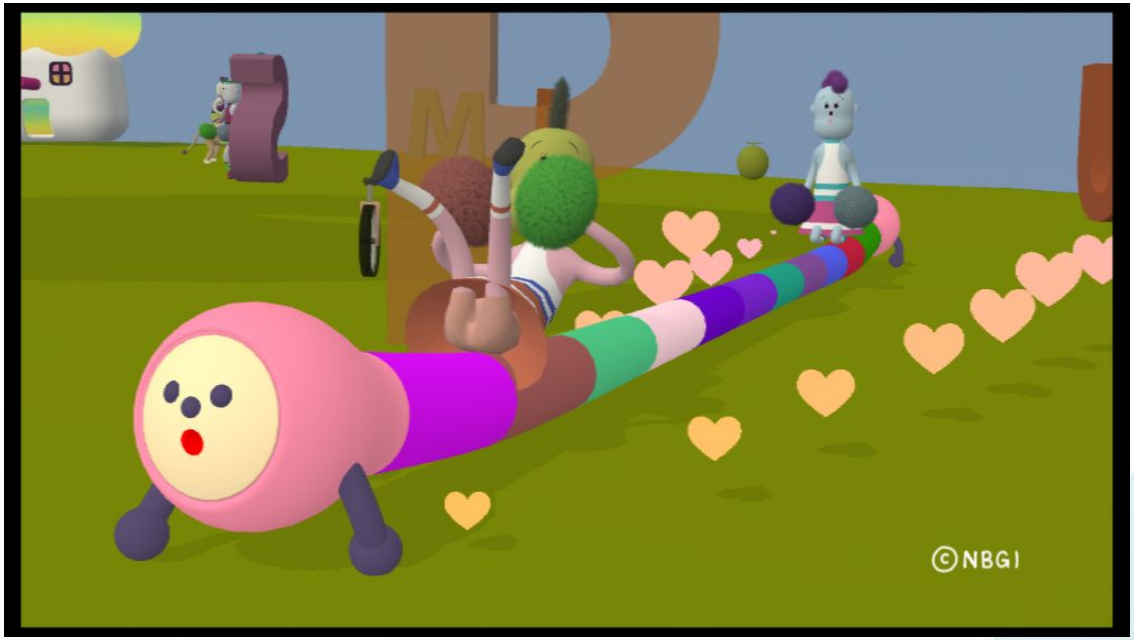


THE HOTHEAD

HE'S HAD IT WITH ALL THE MARK-FAKING CHEATERS IN THIS MONDAY-TO-FRIDAY SCHOOL

Stretch your features

- Production-wise, try to think of things that can:
  - Can be scaled/varied with parameters
  - Aren't gimmicky things that will get tired after 1 or 2 encounters



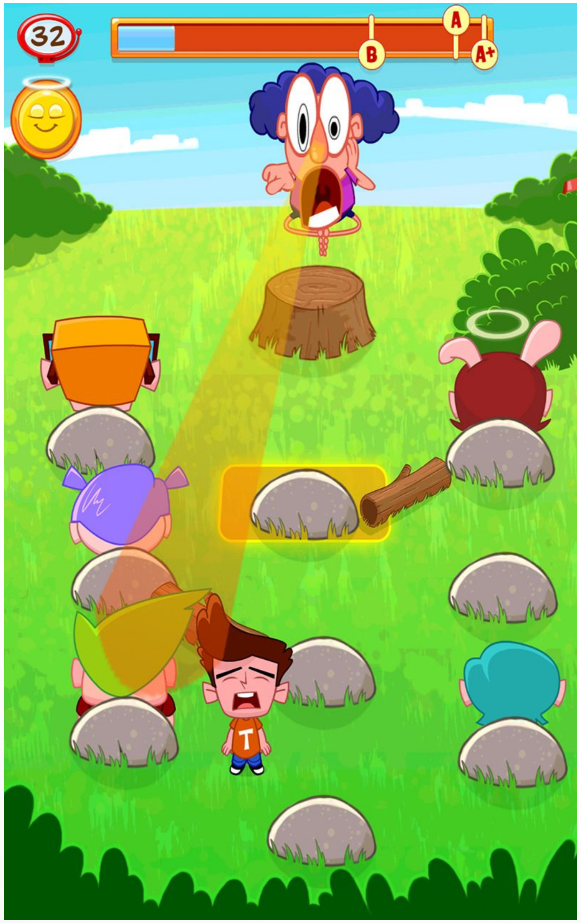
How many levels?

- Number of levels is partially a **business** and **production** decision, but you have to know in order to plan
- Make sure you leave some room to grow!

Planning progression

- Break down all **concepts** that player will encounter into a list
  - **A feature is not a concept!** Anything that's new to the players is.
  - Combining two concepts = new concept
  - Example: Static enemy, moving enemy, shooting enemy, moving shooting enemy

Example concepts: Cheating Tom 2

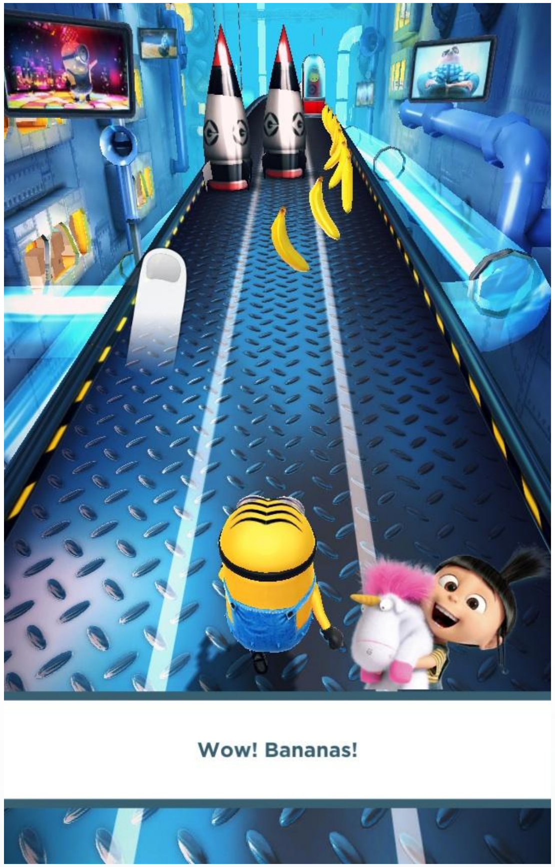


Basic controls	
Grade fillup	
Time limit	
Anger	
…	
Pickups:	Teleport
	Freeze time
	…
Kids:	Normal
	Bully
	Nerd
	…
Teachers:	RedHair
	PinkHair
	Hilik

The first levels

Teach 'em

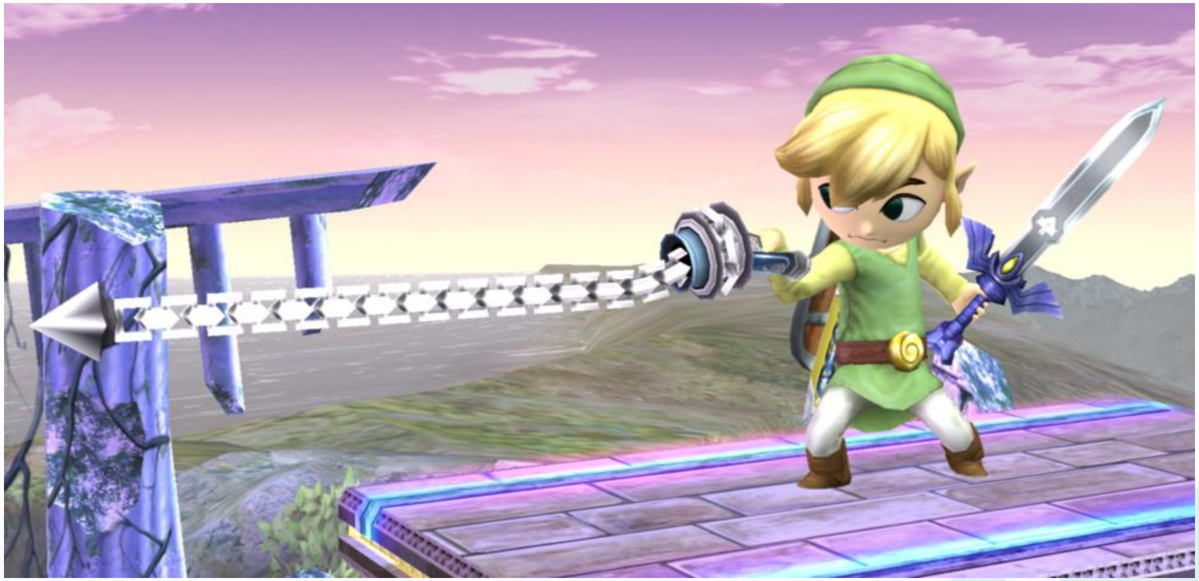
Teach the core mechanics during the first level
You have a small grace period where players know they're in a tutorial



Wow! Bananas!

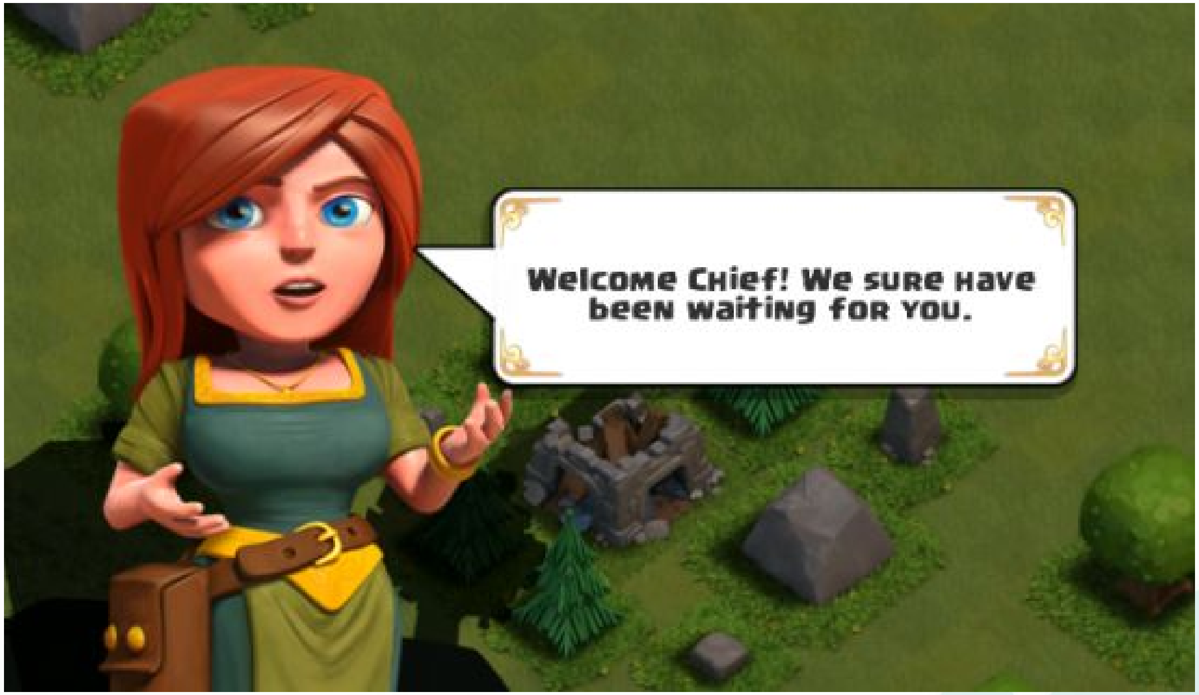
Hook ‘em

Hook players at **end of first level** or **during second level** with cool easy moment - a chain reaction, a big battle, etc.



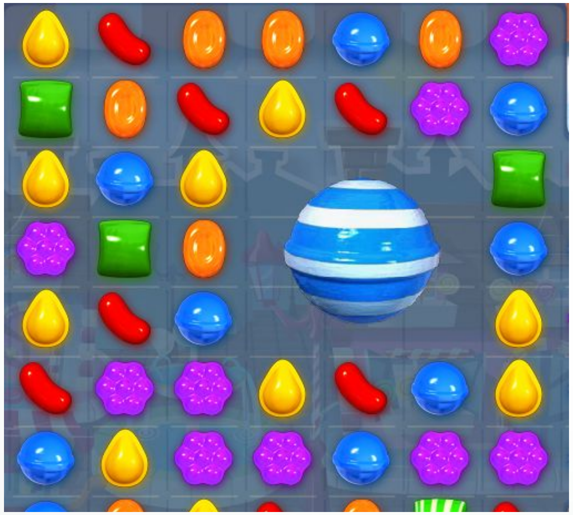
A bit of story helps too

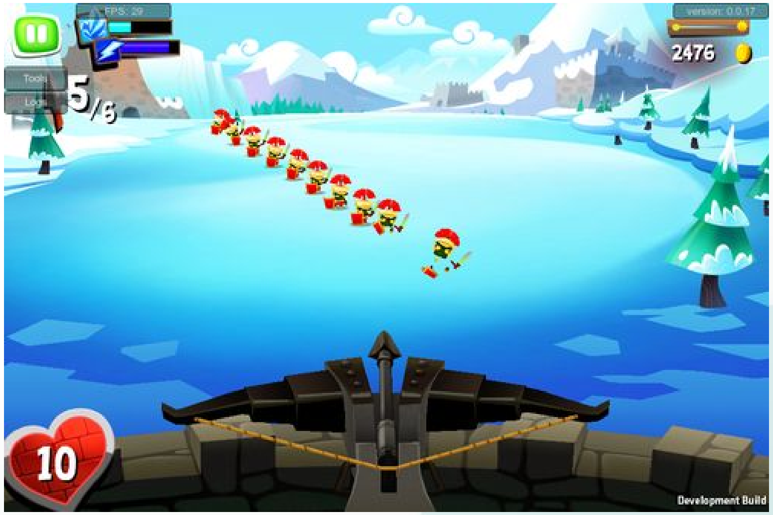
Make the player feel like they're a part of the story



Don’t hold back

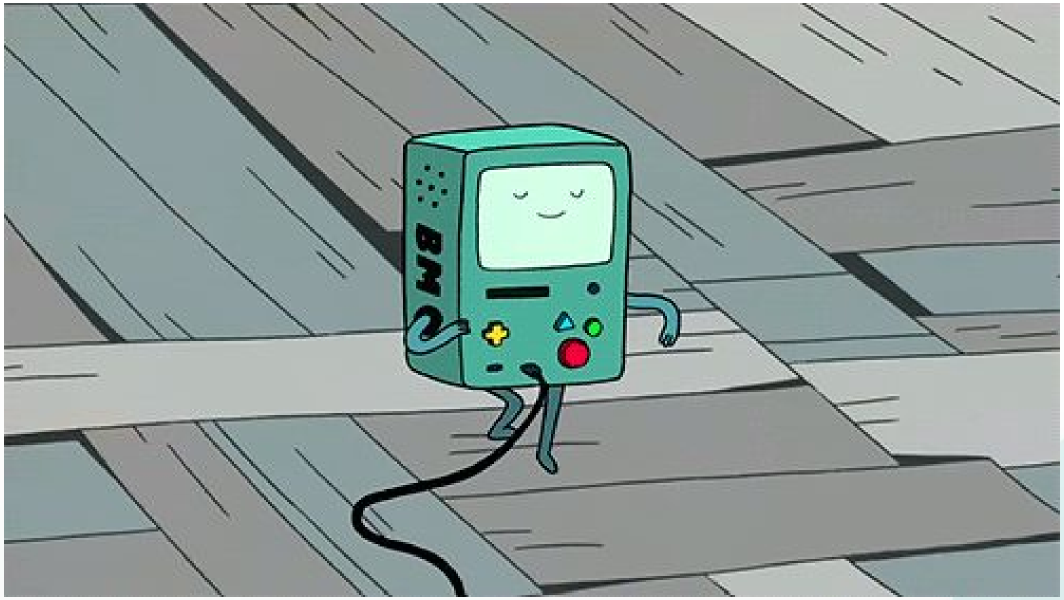
- Make sure to **show off** cool features and make **big visual changes** early, show players there’s more to expect
- If your game has some visual progression, consider teasing them with view of leveled up player or character





Get it in player’s hands

- It’s not a game until someone is playing it - **playtest ASAP**
- In soft launch, the first thing you’ll be doing is A/B testing the first levels



Initial drop: Not just level design

Every game will experience a drop in users during first session, these are players deciding whether the game is for them.

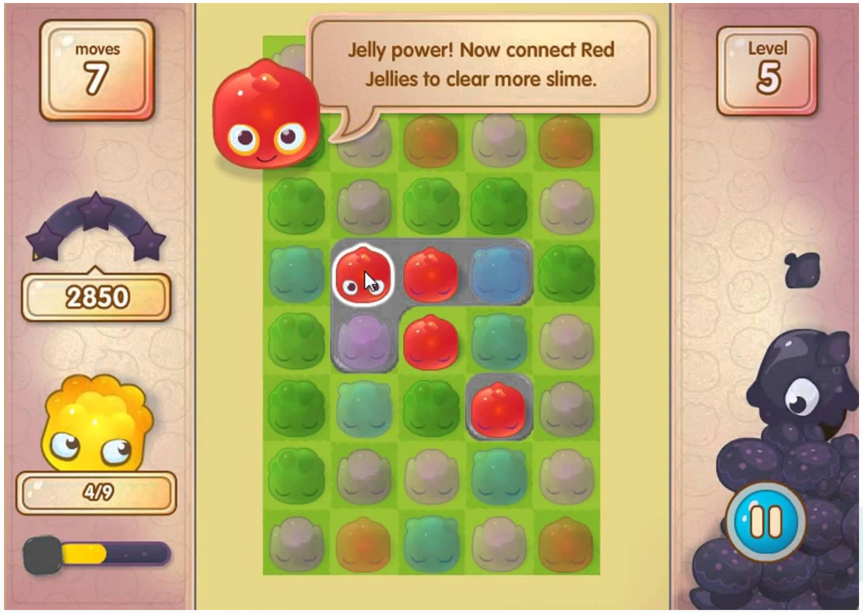
The main things determining the initial drop are:

- The theme and core mechanic
- Marketing - was the game communicated properly to players?

The long haul

Baby steps

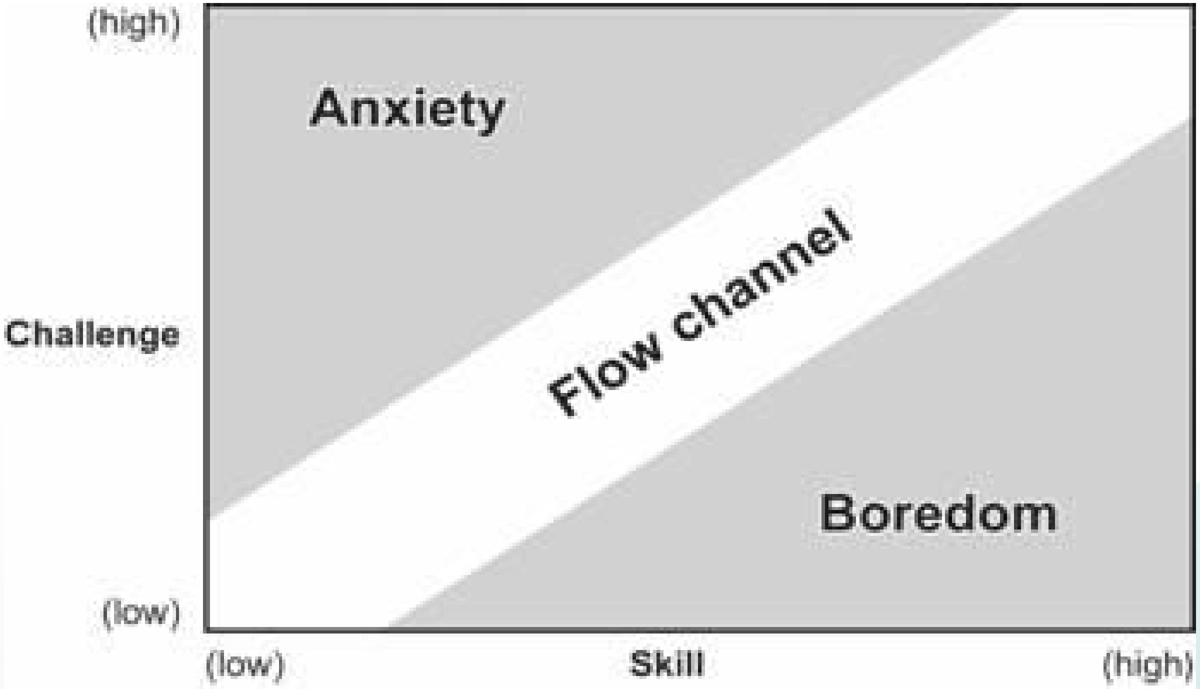
- Introduce every concept in a controlled environment with low risk to player
- Then combine it with everything else learned so far and ramp up difficulty a bit
- Have lots of concepts at start, slowly introduce less as you progress



Jelly power! Now connect Red Jellies to clear more slime.

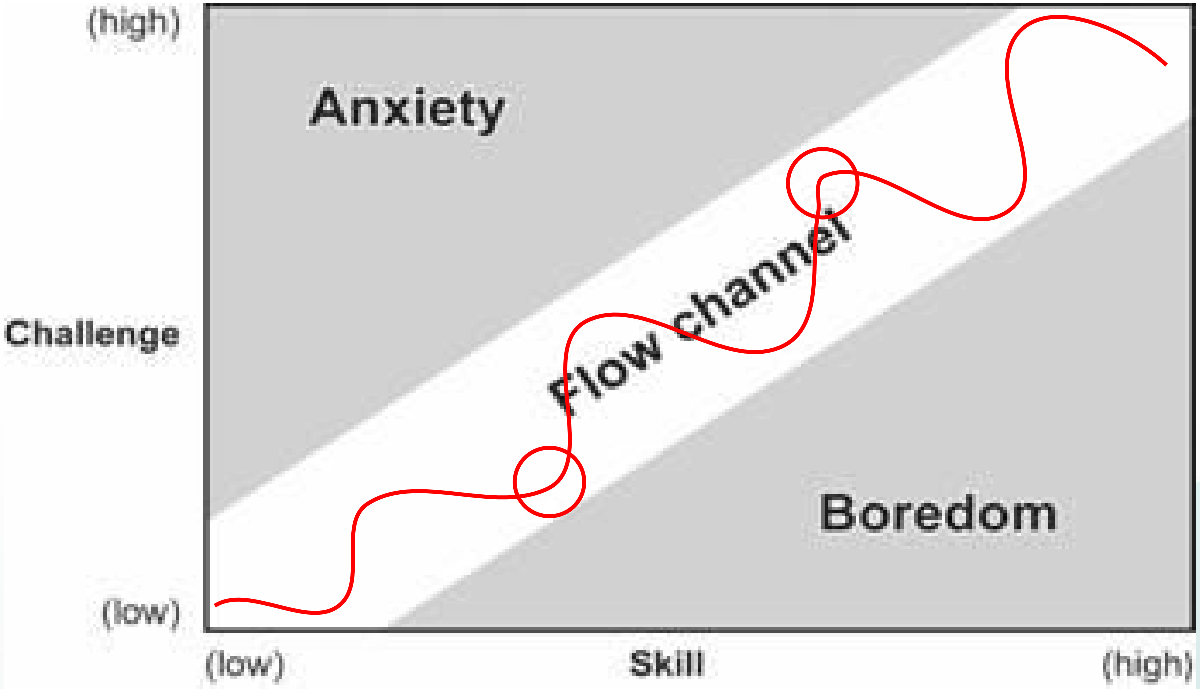
Our Old Friend Flow

Recognize this?



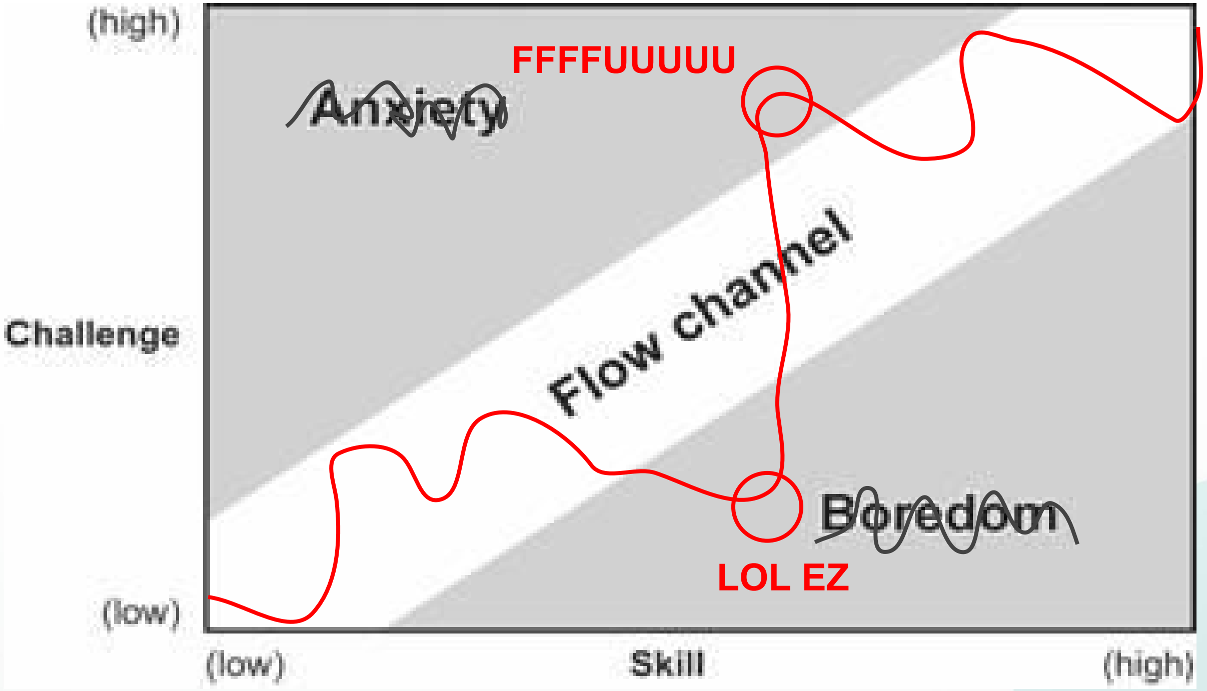
Classic progression

Staying in the channel leads to joy and fulfillment :)



Of course! But maybe...

- Feeling OP is fun!
- Bringing players to brink of quitting encourages them to convert and builds love/hate relationship



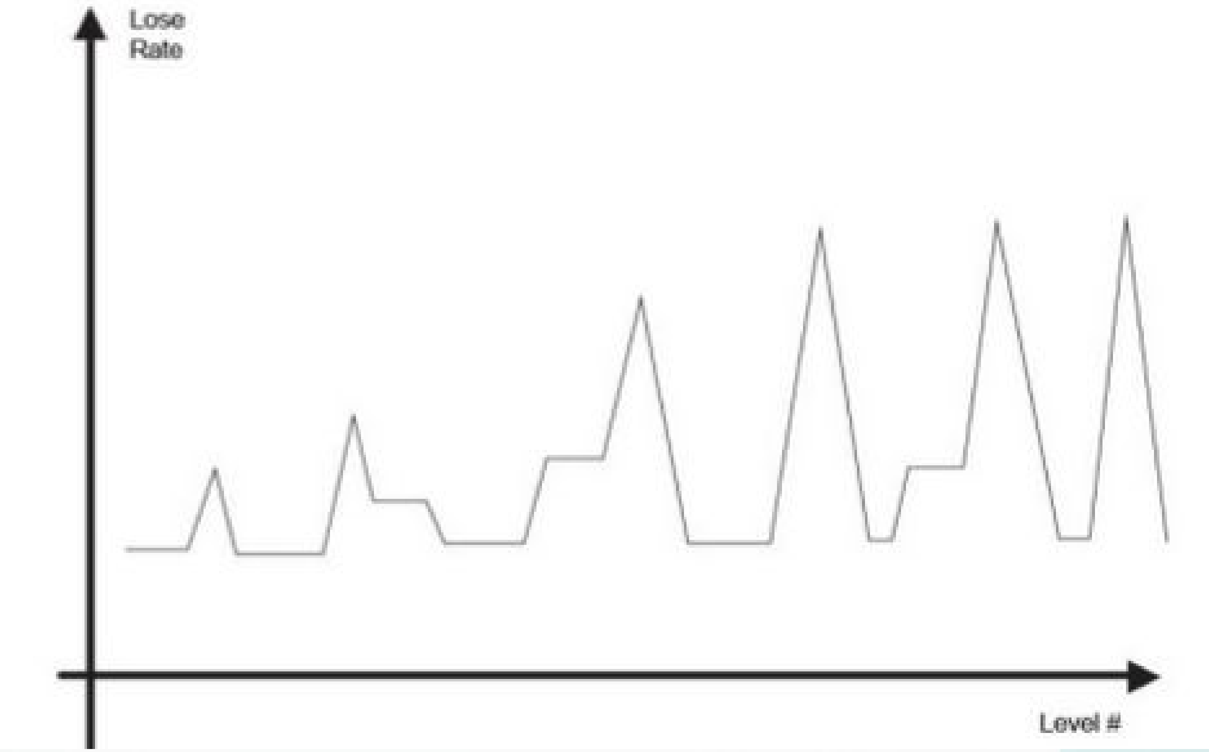
F2P Level Flow

The following couple of slides are graciously borrowed from Florian Steinhoff's [excellent presentation on Jelly Splash level design](). You should go look at that.

The gist of it: Players need challenge to stay interested. Build a love/hate relationship with the game

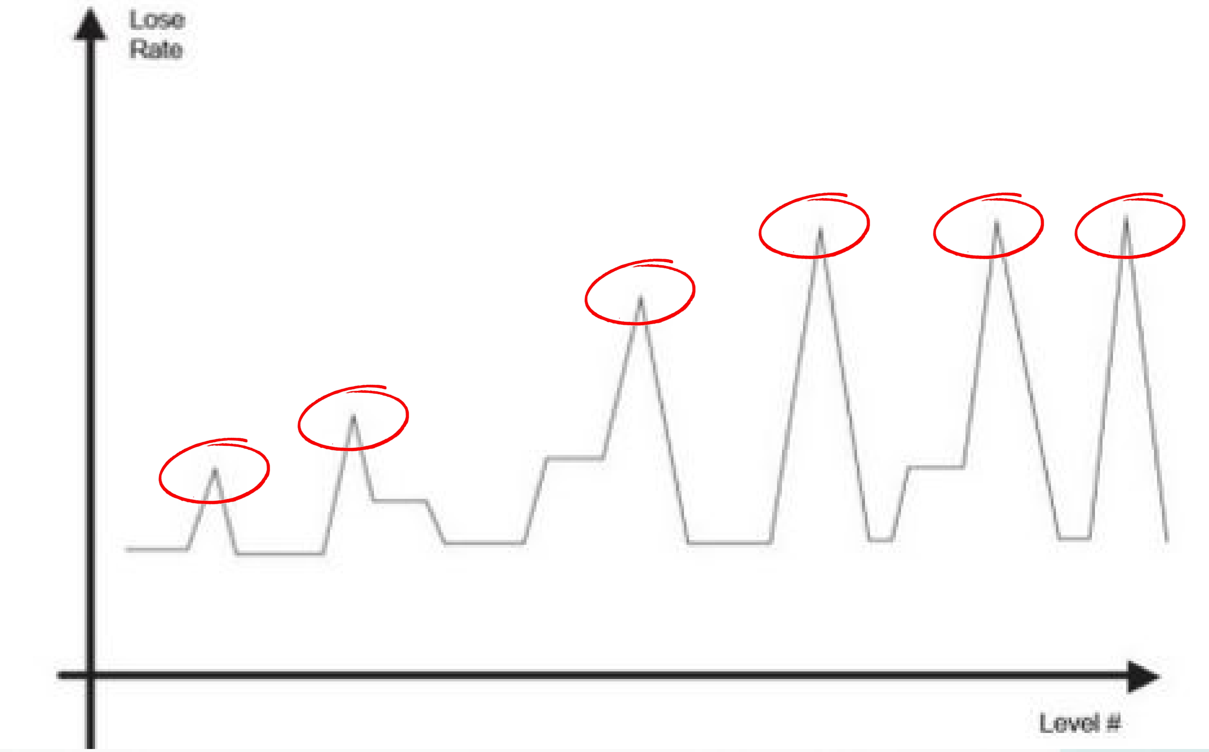
F2P Level Flow

The ideal progression graph



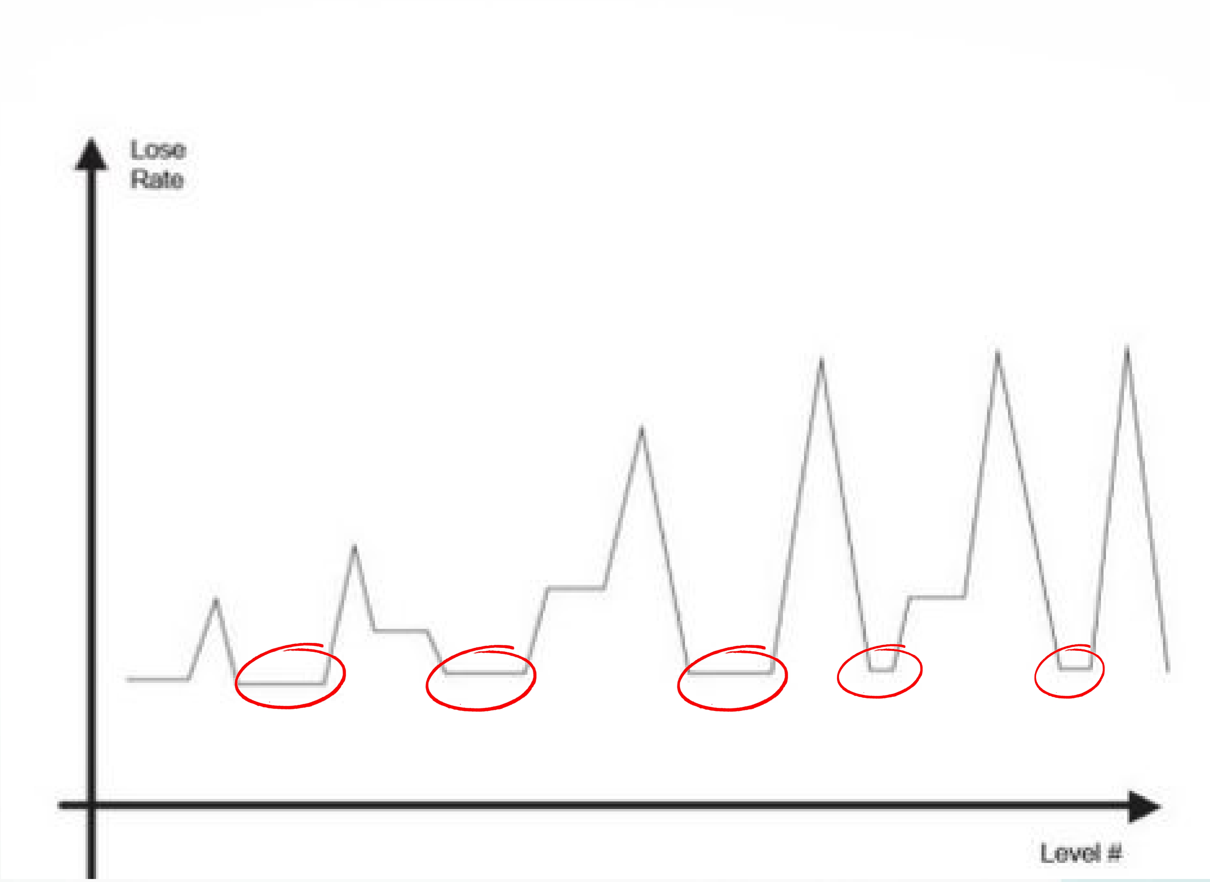
F2P Level Flow

Blocking levels



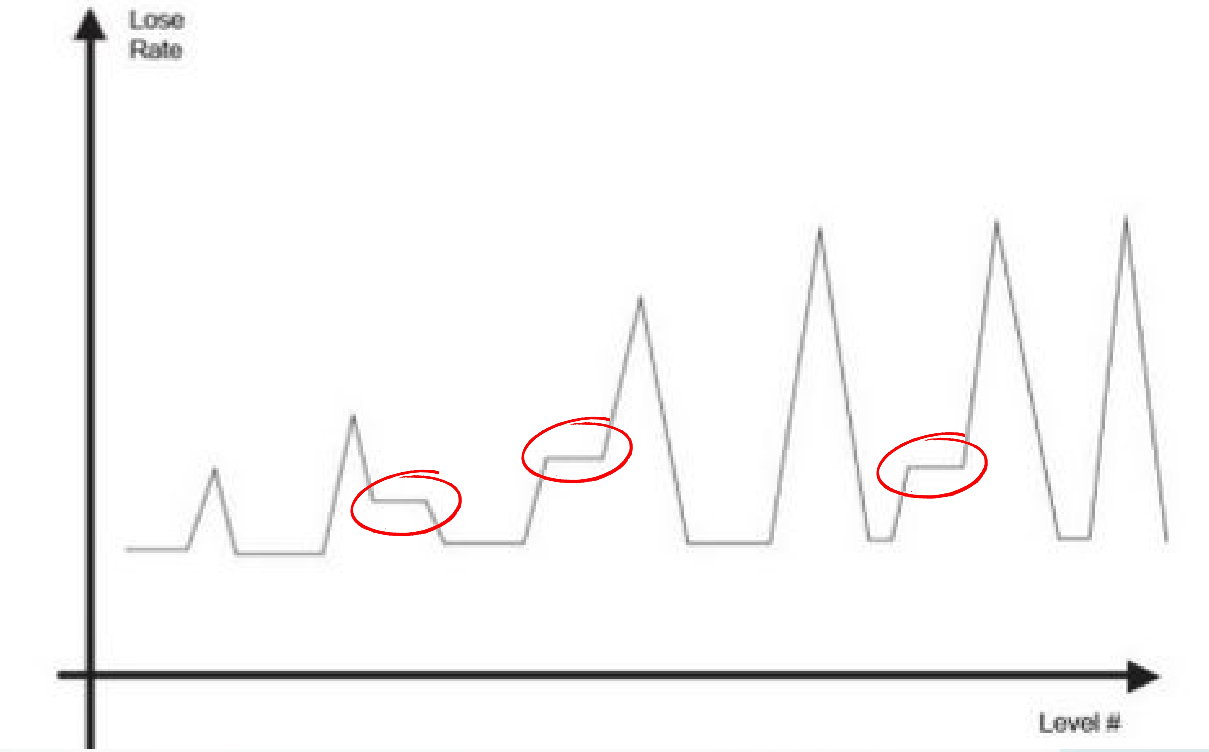
F2P Level Flow

Relief levels



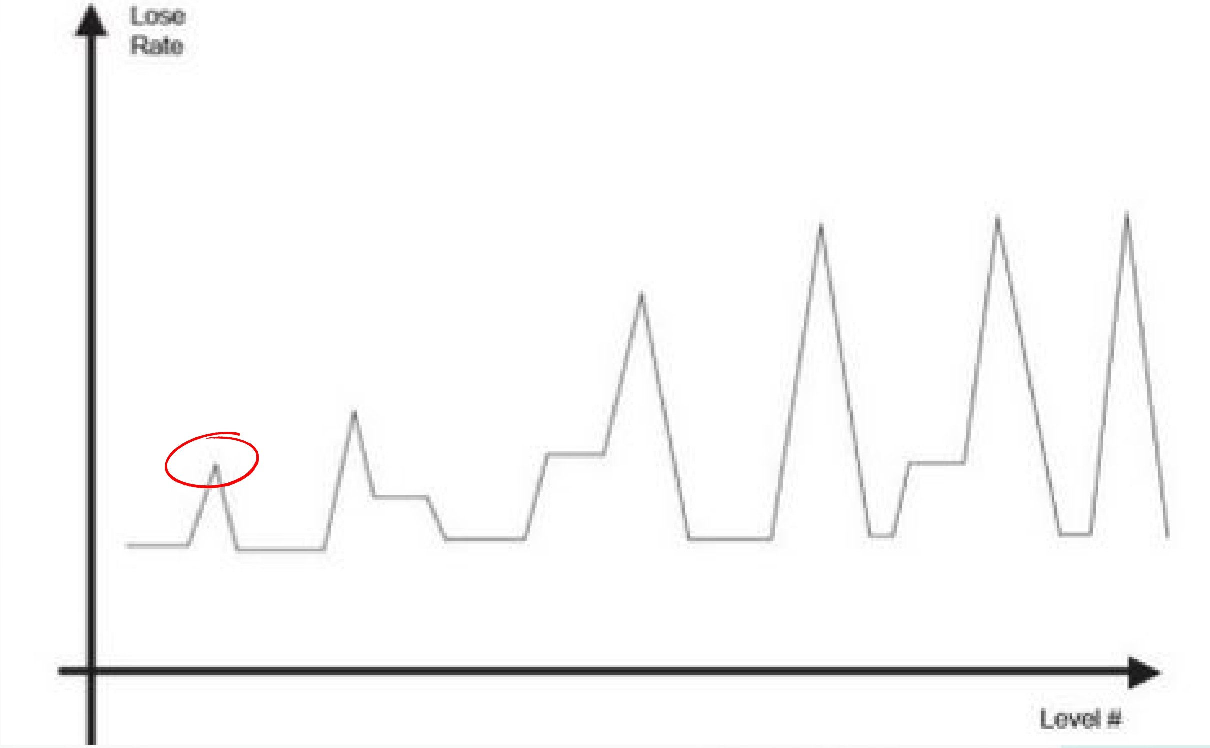
F2P Level Flow

Buildup levels



F2P Level Flow

Force first session loss



The FUUU Factor

FUUU factor =

$$\frac{\textbf{\# of tries until won}}{\textbf{\# of tries nearly won}}$$

Example plan: Cheating Tom 2

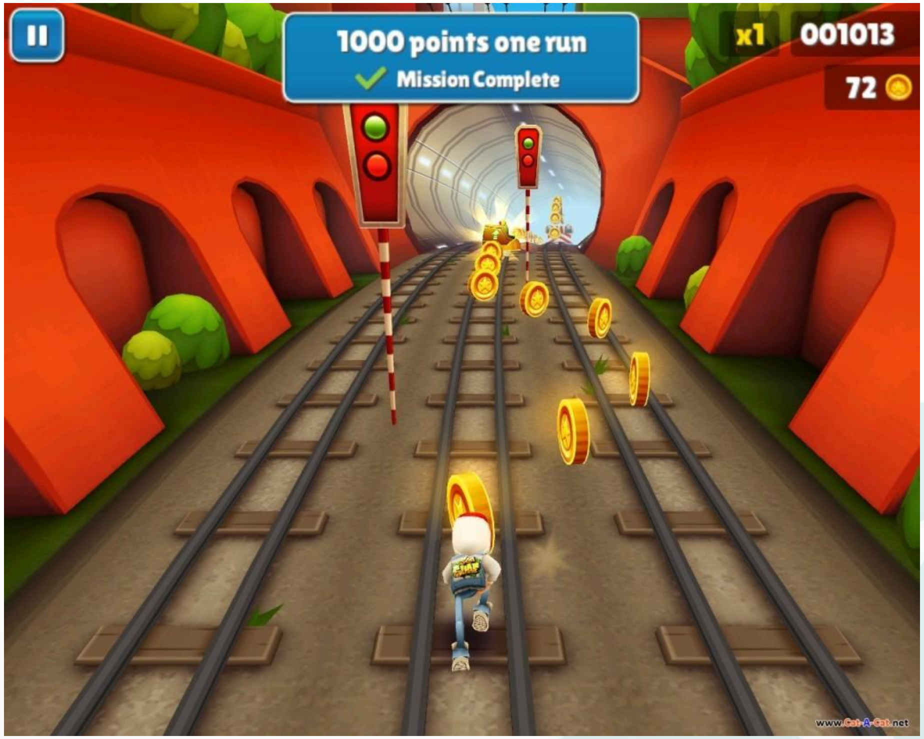
Lv	New concepts	New students	New Teacher	Mode	Diffi-culty
1	Basic controls, grade fill-up	Blonde girl	RedHair	Normal	1
2	Teacher looking, blocking students	New girl		Normal	1
3	Timed test, angry students	Nerd	PinkHair	Normal	2
4	Yearbook (pre level)			Normal	2
5	Mode - Anger MGMT			Anger MGMT	2
6	Teacher distract			Anger MGMT	2
7		Hothead		Anger MGMT	3
8	Ghost powerup			Normal	2
9	Upgrades (before level)	Bully		Normal	2
10	Bully + anger level			Anger MGMT	5

Test your plan in the real world

- A/B test several variations in **big chunks** of levels - don’t pick and choose small changes
- Start with your first 10 levels, then move on to the next 10, etc
- Be brave, make extreme changes

What if my game is endless?

- Flow is still king!
- Difficulty builds up
- Spikes of difficulty every X seconds
- Relief after every spike

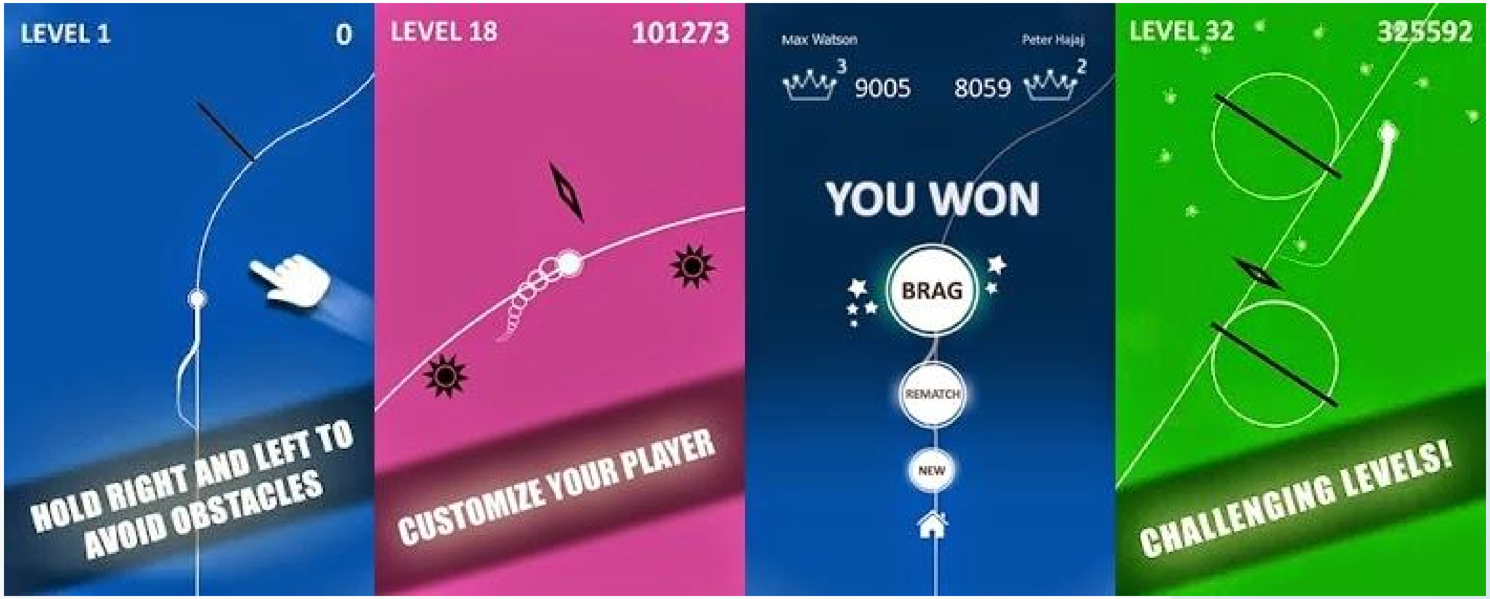


Start harder!

- If you start too boring, advanced players will get tired of replays
- Difficult starts can motivate new players
- Make **restarting quick** and easy!

Progression in endless games

- Add optional challenges during difficulty buildup
- Mission systems can replace cross-level progression by making players change playstyle and revisit concepts
- In Linebound, we used an XP method that proved itself very well



You have a plan.
Get to work!

Rule #1: PLAY PLAY PLAY

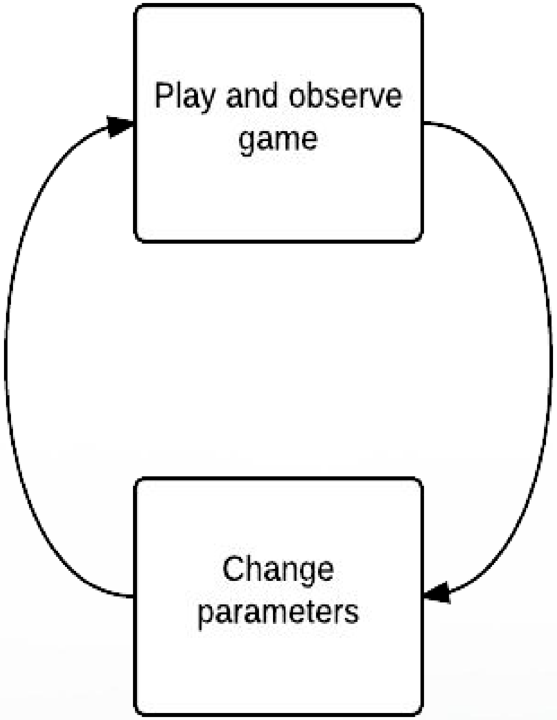
- Most of your time should be spent playing your game
- If your game has random elements, tough break.
  Play each level 10 times to avoid wrong conclusions

Designing the right tools

Your workflow requires you to constantly make changes and replay.
You're a designer, so design the right tools for yourself!

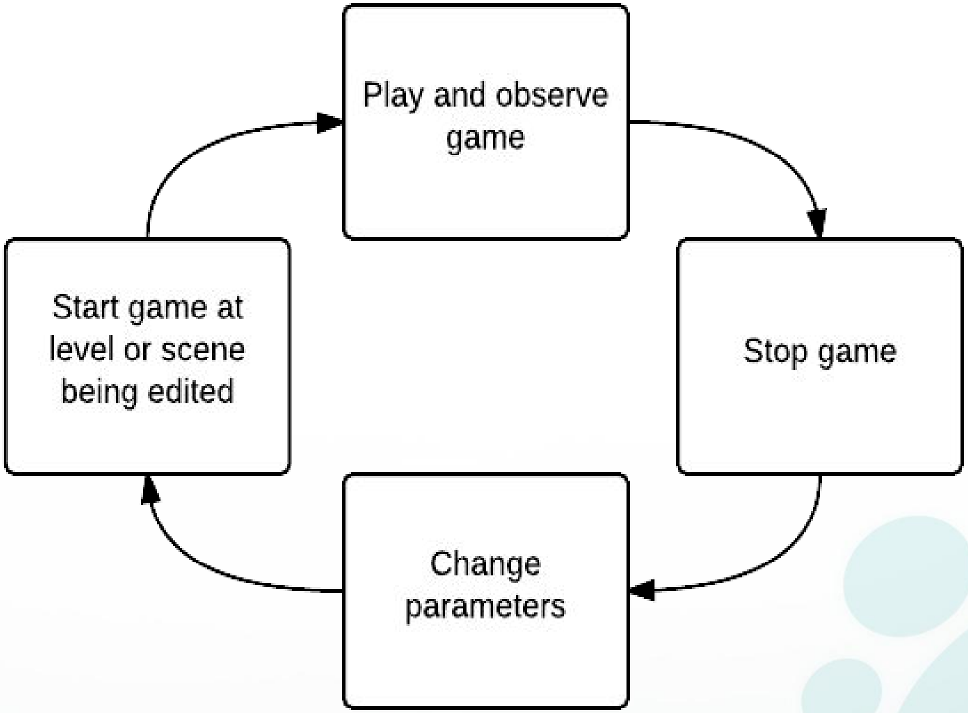
The re-iteration loop

Real time editing



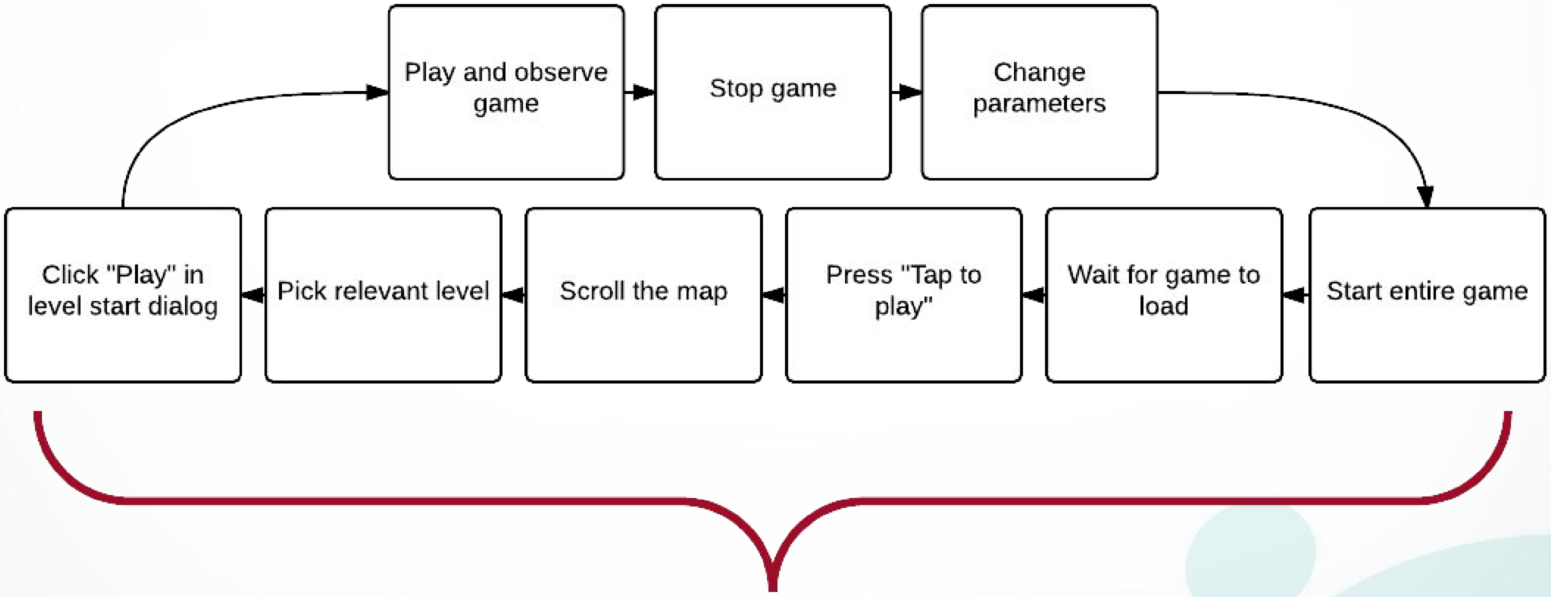
The re-iteration loop

Restart relevant section between edits



The re-iteration loop

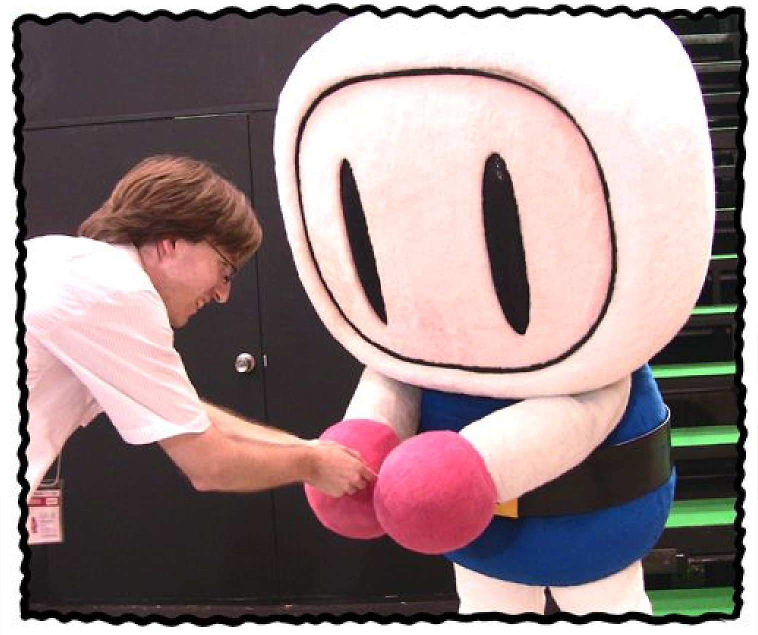
Restart whole game between edits



Wasted time and energy

Listen and compromise

- Listen to the technical guys, they may have better ideas on how to build your level editor
- Compromise! Your workflow is important, but it's ultimately a trade-off between dev time and your time
- No matter how good the editor is, be prepared for lots of hard meticulous work



Over exposure

- Beware exposing all parameters in editor, or “overriding” on game elements
    - **For the sake of the player**, the game must have logic. If they see the same element, they don’t expect its behavior to be different
    - **For the sake of your workflow**, it’s easier to create a variant element in a different editor, and just choose that in level editor.
For example: “orc-tutorial” and “orc-normal”

More tool tips

- Anything chosen from a finite list should be in a dropdown menu or typed with auto-complete
- Parameters related to visuals or positions in the world should be edited visually, preferably with drag and drop, scaling, etc
- Don’t clutter your editor with too many options. Current thing being worked on should fit in single screen
- Separate things into several bite-sized editors

That's it! Go be awesome

- Plan your progression with flow in mind
- Hook players with an awesome start
- Endless games also need spike and relief
- Shorten the iteration loop when working
- Play the crap out of your games

Thank you!

Feel free to send me questions!
www.drory.co
eladd@tabtale.com