

Game Biz

Tiago Tex Pine

@texpine

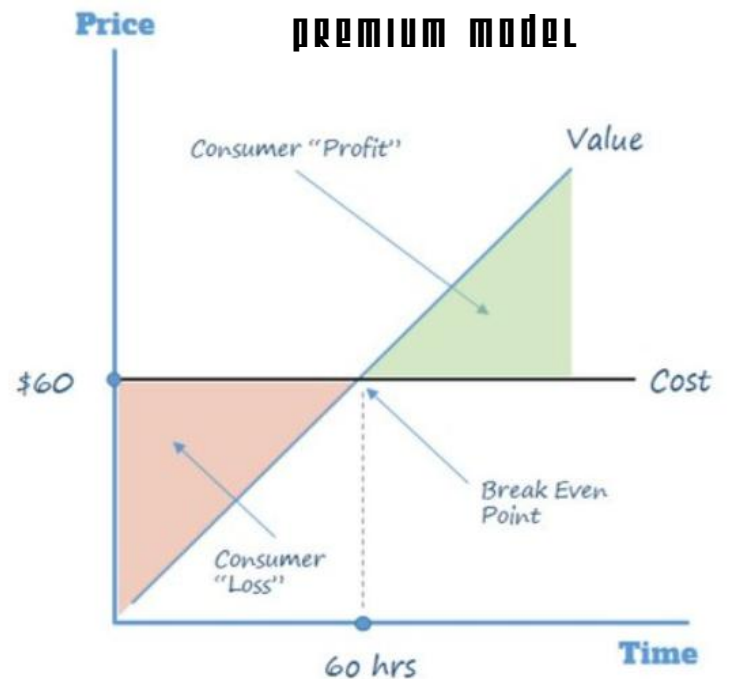


Design = Business Model

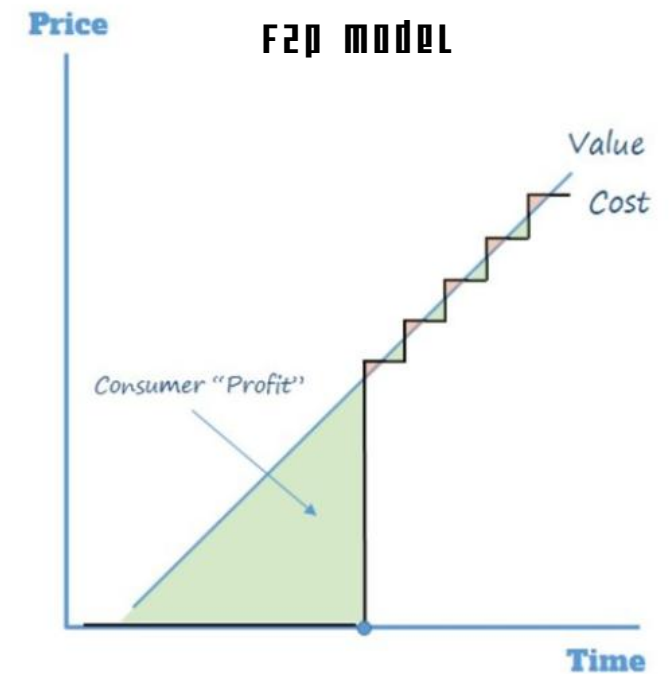
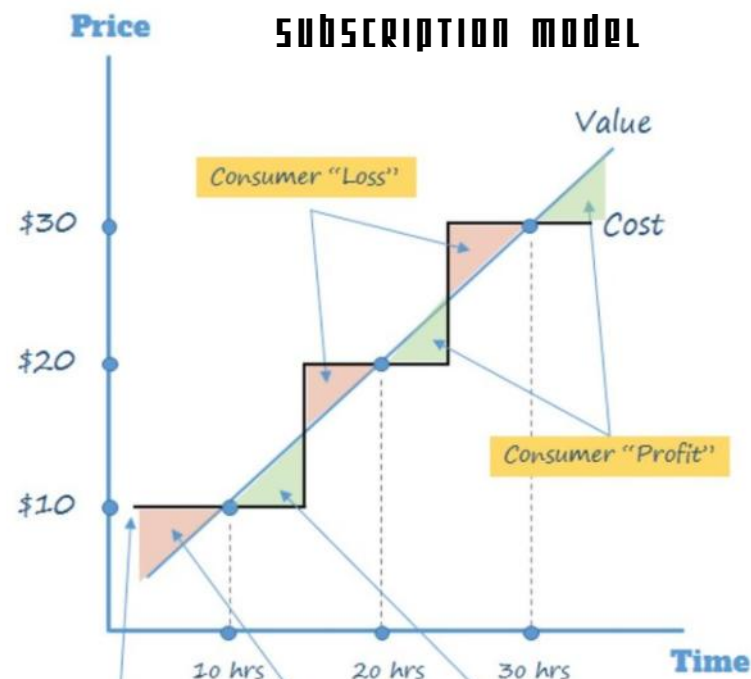


the Designer is the Business owner

Value timeline across business models

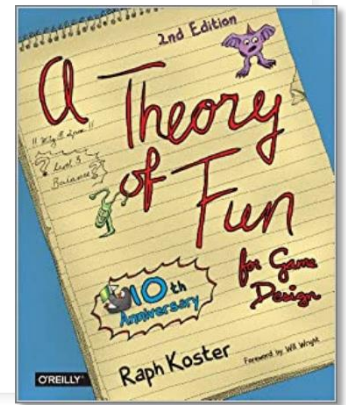


I pay a fixed cost of \$60 to buy a game, but the value that I get increases over time. The old school model of games.

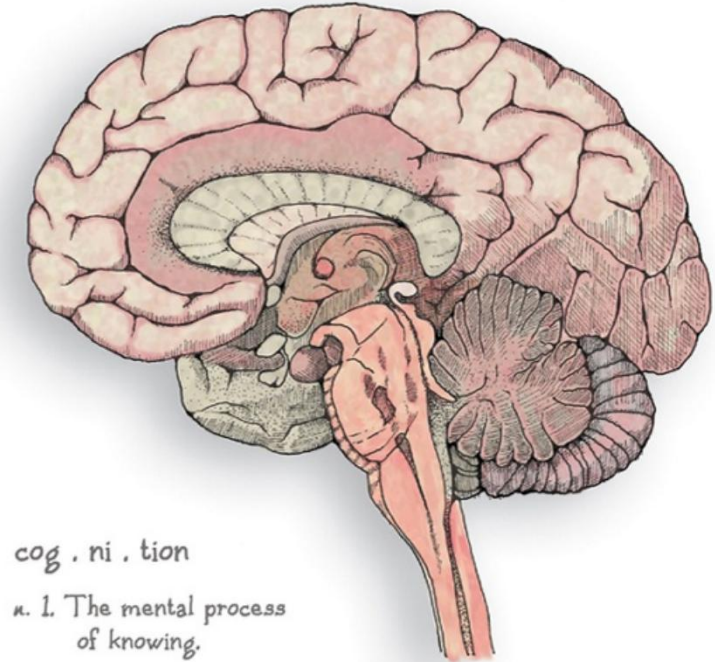


I get an abundance of consumer profit in the beginning with a free-to-play game because the cost to me is zero. However, as I progress I get pinched and I end up paying. At an aggregate level (i.e. across all players) users are paying for the product after trying it for free. Users get all consumer profit up front, but the profit is then split between consumer and producer later.

What is "fun"?



- Game design = creation of systems.
- The brain has fun when learning new systems.
 - Games are created to teach systems that modify the wiring in our brains
 - Art form that re-wires people's brains - "We actually get to engage in direct mind-control"
- Fun is very dependent on the neurotransmitter of reward: **dopamine**.
 - Dopamine enhances learning and memory. Teaching signal to the brain
 - Dopamine relates to predicting rewarding outcomes.
 - But also gets dumped when there are *unpredictable* situations as well, in order to encourage you to solve them



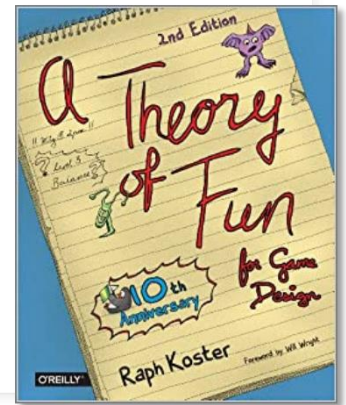
cog . ni . tion

n. 1. The mental process
of knowing.

[from Latin *cognitio*.]

-- they are about cognition,
and learning to analyze patterns.

What is "fun"?

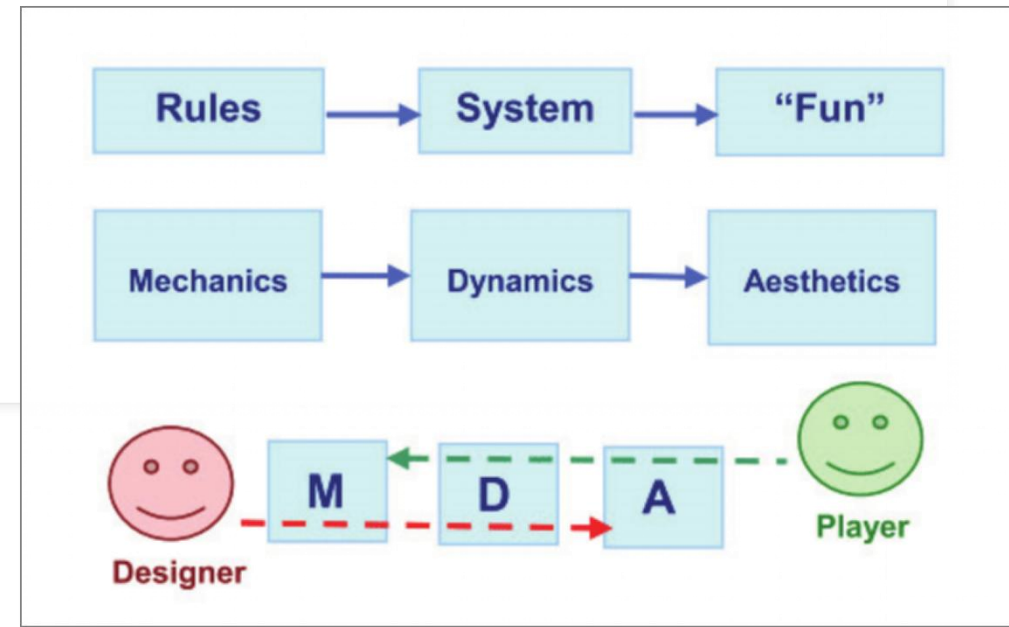


There are four core mechanics in games, Koster theorizes:

1. Solving problems *heuristically*:
 - Approximating solutions, using mental shortcuts, experimenting.
2. Understanding other people and social relationships.
3. Mastering physical space and physical relations.
4. Exploiting the *natural human difficulty in estimating probability*:
 - Not only gambling or loot boxes: card games, RPG loot, fantasy sports, war games, sims, etc.

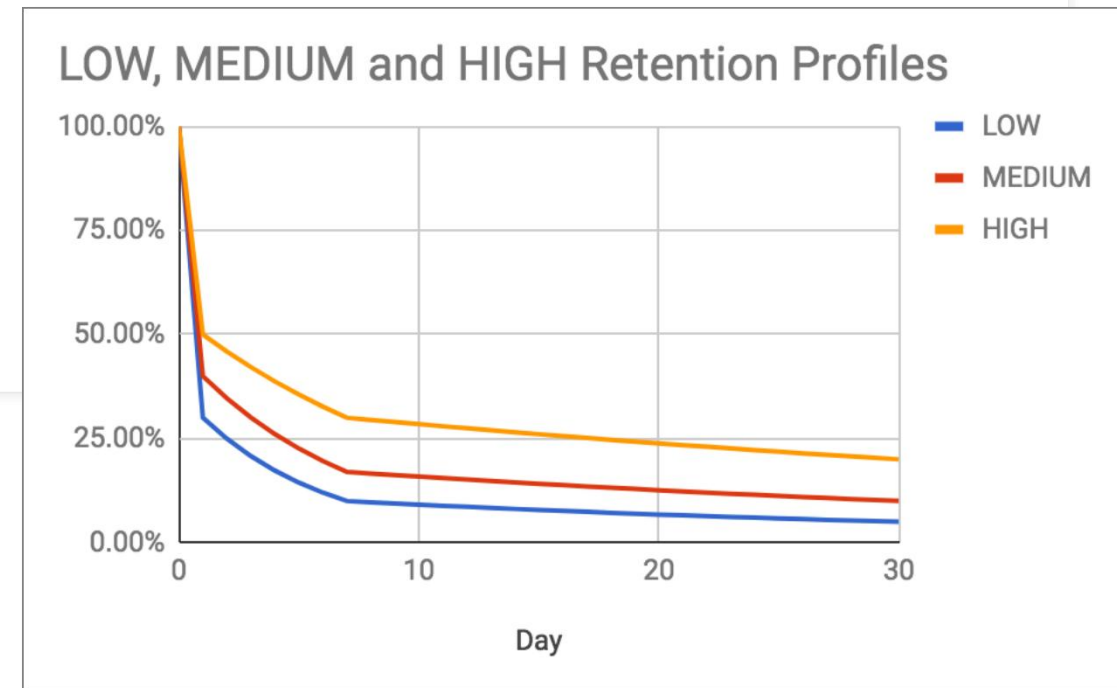
Layers of Design

- **MDA framework:** game design is split in layers between mechanics, dynamics and aesthetics.
- **Ludonarrative dissonance:** conflict between narrative told through aesthetics and narrative told through mechanics.
 - Gap between what the game *is* vs. what the game *surfaces*.
 - BioShock: while narrative promotes selflessness, mechanics promote self-interest
 - Uncharted: a likeable rogue we are supposed to like who kills hundreds of people
- Portal:
 1. Macro-level of escaping the AI and the lab.
 2. Smaller level of beating one stage of the game.
 3. Micro level to the subtleties of positioning the gun, dealing with physics, solving puzzles.
- Monetization is in the highest-level layers (premium/cosmetics) or the lowest level layers (resources/IAPs).



Retention

- Keep players coming back is the first tenet of monetization.
 - Premium games will benefit from word-of-mouth and healthy metagame.
 - Any game with IAPs will have opportunities to promote and sell new content.
- In order to keep players, they need to *keep exploring possibility spaces*.
 - Most games repeatedly throw evolving spaces at players so they explore the recurrence of symbols within them.
 - When you finish, the game will give you another space, and another, and another.
- It's tricky to improve Retention *after the game launches*. So it's important to understand what makes it work.



Retention Main Components

1. The Core Loop

- The basics: core gameplay, fun to play.
- Addictive and long-lasting.

2. Tutorial (On-Boarding)

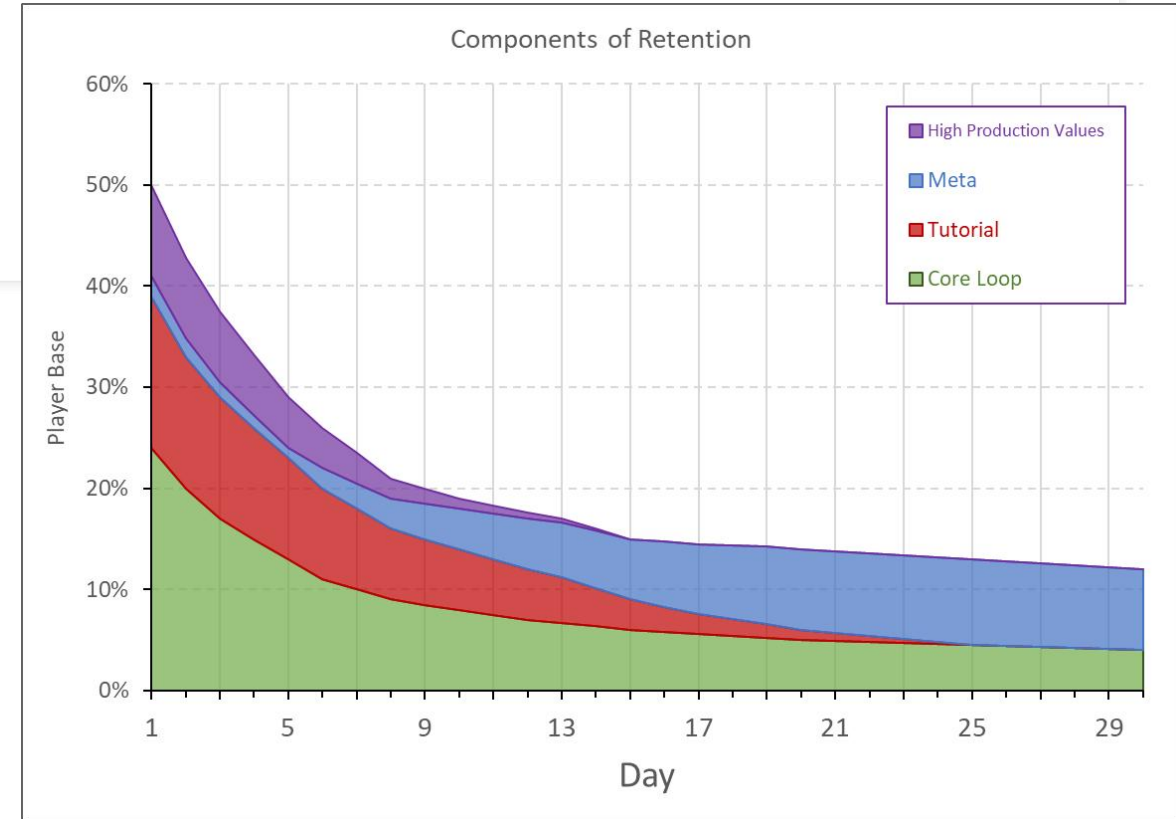
- Clear guidance improve early retention dramatically.

3. Metagame

- Systems that last, game that feels like it changes over time.
- Depth of interactions outside the core loop, players keep exploring new ways to play.

4. High Production Values

- Art, visuals, GFX works in the first few days. But may become a detriment on loading times later.



Retention - Other Basics

1. Game is not buggy and doesn't crash
2. UX is easy to understand and players know what they need to do
3. Notifications, timers or mechanics like daily rewards or quests pulling your players back
4. **Sessions aren't too short or too long** - you've built up a good heartbeat
 1. PC and consoles have much larger sessions, but even then it's healthy if players keep coming back for more sessions than having a few very long ones. The idea is, *the game enables short sessions* if players prefer.
5. Difficulty / pacing isn't too "spiky" - players aren't leaving due to poor balancing
6. **Each session the player feels like making progress** - returning always feels rewarding
7. Constantly updating the game to keep it fresh and give players more reasons to come back (and feeling like you care about the game)

Retention vs. Monetization

- It is a fallacy to divorce Monetization from Retention.
 - If Monetization improves and Retention holds steady, the game makes more money.
 - If Monetization stays the same but Retention goes up, the game makes more money.
 - While devs would ideally focus on both, in reality they can only focus on ***one at a time***.
 - The trick is to figure out at what point you can stop focusing on one or the other.
- And often, there is a trade-off:
 - Focusing on Monetization may add extra **friction and paywalls** good for revenue but bad for Retention.
 - Focusing on Retention may keep players for longer but **fail to solve the problem of declining revenues**, squandering opportunities for Monetization
- As the game ages over months and years, a declining user base will push devs towards Monetization features

What is a Monetization feature?

1. Content sold directly to the player
2. Content includes new ways to spend resources
 - It's a matter of *amount* asked.
 - The more gating the costs are, the more likely the feature was designed for Monetization.
 - Systems added to pace content or prevent players from running out are not for Monetization, like Energy systems.
3. Content forces players to engage in existing Monetization
 - Assume that a player won't spend unless they feel like they have to. A player must objectively feel pushed or compelled to spend in order for that content to affect monetization.
 - If a player does not feel objectively compelled or pushed to spend, then that implicitly means the content is not made to monetize.



Design Topics



Types of features that help monetize

Loading Times



- Very first front of Retention: how fast does the game *loads*?
 - Often *overlooked* by game developers, require serious engineering from the ground up.
 - It's also a design and art firection problem - super high-end art assets will never load fast.
- Different for every platform, as players consume content differently:
 - PC players are the most patient. Lots of other activities in the desktop, hard to measure.
 - Console players are patient. Loading times after **~20 seconds** starts affecting engagement metrics
 - New generation of SSD-based hardware will make console players *more impatient*.
 - In Mobile it's a huge problem. Anything above **8 seconds** lose players.
 - Users on Mobile don't want to wait not even 3 *seconds* in other types of apps and web
- But doesn't matter if free-to-play or premium.
- Sometimes engineering into loading times will make more for retention than fancy features - the key is to understand such engineering is a **game of diminishing returns**.

"Retention features"

- Daily Rewards

- Strict: lose one day, resets. These should be short, max 5 days.
- Lenient: by returning day, no matter how many days without activity. Can be longer, but not as good.
- Long schedules: don't do them. They punish players for *not* coming back.

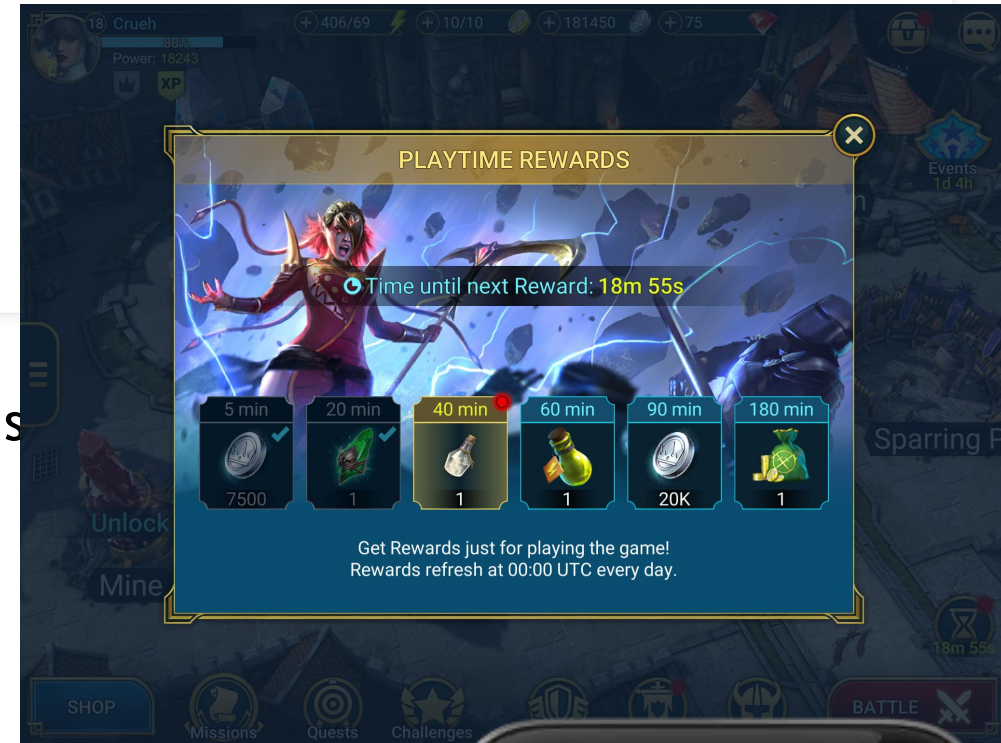
- Daily Quests

- Because it requires constant action, works to keep *players already committed*. Keep them engaged for longer.
- Don't work with casual players - unless the action required is extremely simple.



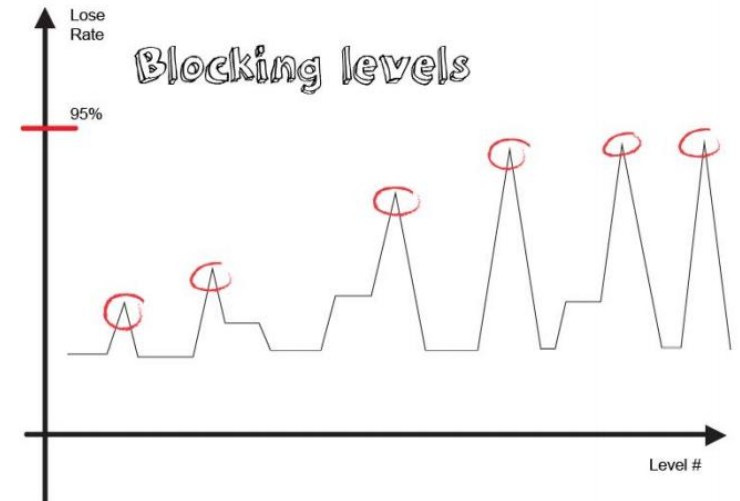
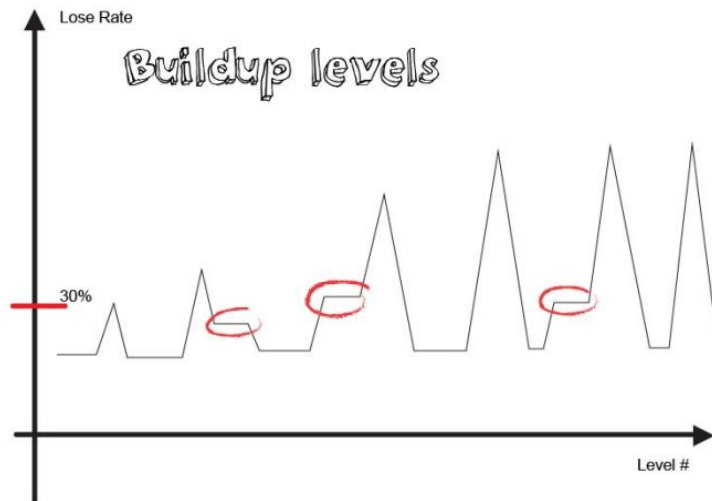
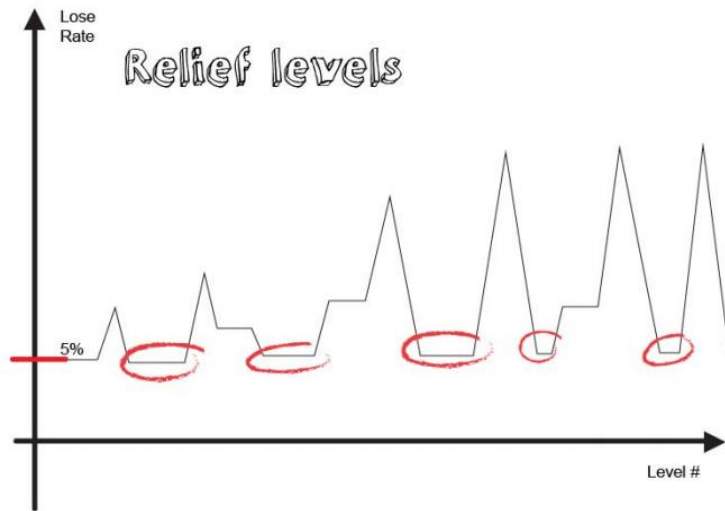
"Retention features"

- **Appointment mechanics:** timer-based content players seek to collect.
 - Every few hours to 72 hours, depending on design intentions of amount of sessions per day.
 - Most games can find a context that works within its fantasy: harvesting, nurturing, free drops, free boxes.
- **Achievements:** always good for completionists (~39% of your audience)
- **Notifications:** mobile-only, around 3% of users receive them - but the 3% most valuable!
- **Social media sharing:** waste of time, only works for influencers.



Difficulty Balancing

- You don't want progression to grow linearly. Instead, it should be valleys and peaks alternating.



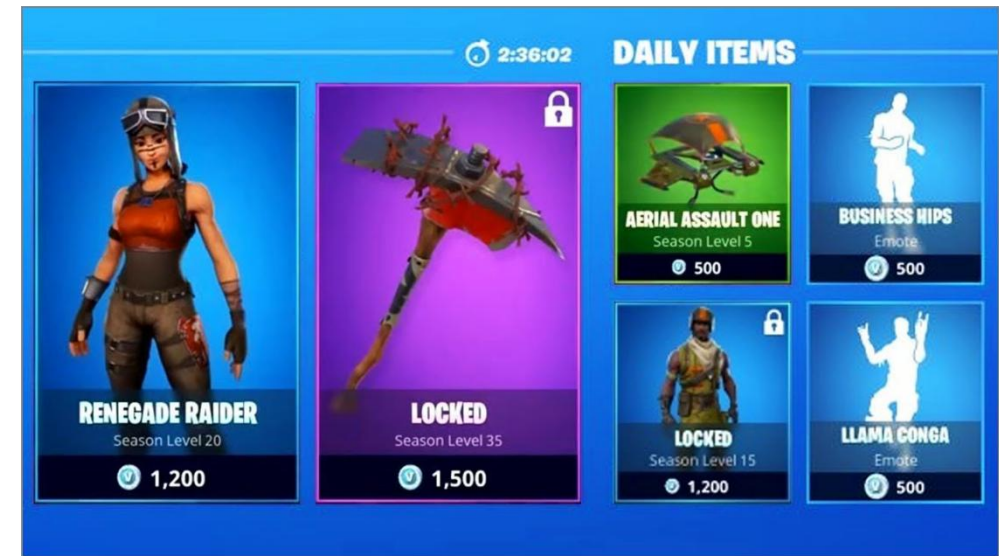
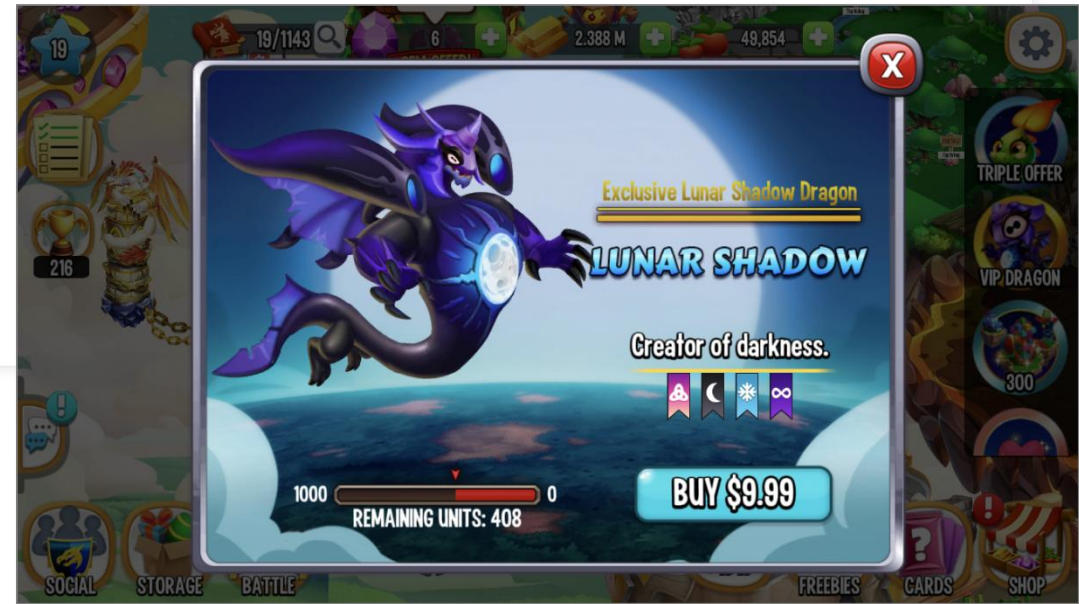
In-game Store

- **Bundle Packs:**
 - Lots of items at once, easy to value and to *anchor*.
 - Seasonal bundle packs - Thanksgiving, Xmas, national dates.
 - Game code and UI to support dynamic creations of Live Ops.
- **Starter Packs:** below-\$10 purchase for first conversion.
 - Always the hardest conversion.
 - Easier to re-convert the same user later.
- **Claim free items in the shop**
 - Claiming could be done in any UI of the game.
 - But shop is chosen in order to bring players into it, show what they could buy.



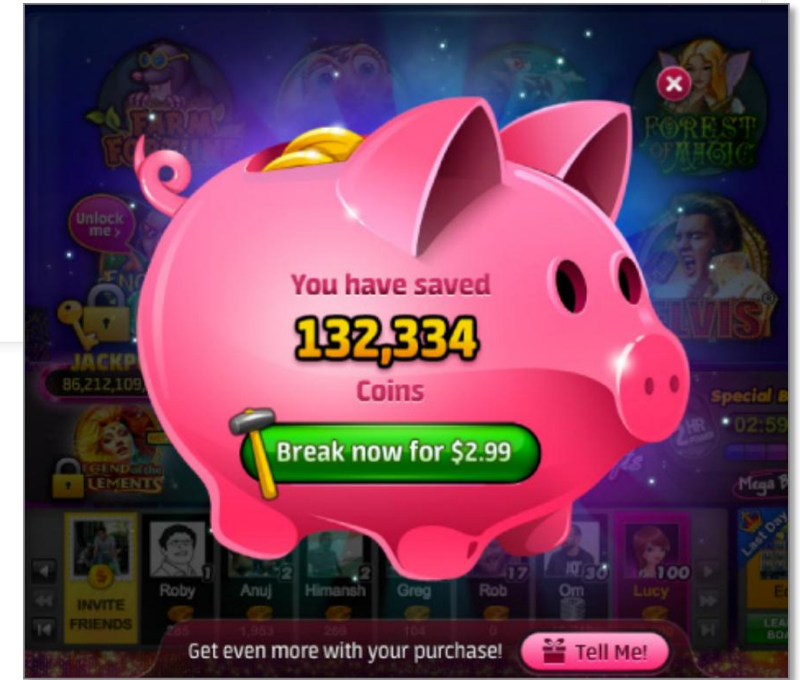
In-game Store

- Artificial scarcity:
 - Timers
 - Max amount of purchases per player
 - Max amount for all players - hard to dev but creates auction-style pressure.
- Anchoring tags:
 - Crossed prices
 - "X% OFF" badges
- "Available soon" or level-locked:
 - Foreshadows items for players, "just keep playing to get there"
 - Players can also strategize and save currency.



Piggy Banks

- Piggy Bank is a deposit feature where players accumulate currency as they play.
 - Currency is stored into a deposit - a piggy bank - which players can claim for a fixed price.
 - The offer grows more appealing the longer the player waits and the more she plays.
- A lot more value for money than normal IAPs
 - Kind of rewarding for coming back.
 - Appeals to the *Endowment Effect* ("it's yours, you saved")
- Popular in Casino and casual games, but even World of Tanks implemented a version of it *that is not purchased*, but unlocked via Battle Pass.



Cosmetics

- Popular on hardcore games as a means to monetize without interfering in the game balance.
- However, revenue *per user is low*.
 - Expect much less than \$1 as an average per active user
 - Conversions will only be higher than 1% if the game has a social context where cosmetics can be "showed off".
- Therefore, games that rely on cosmetic monetization **need**:
 - A large user base, and the means to keep it large.
 - A large art pipeline to pump out content with *cheap marginal costs*. (Like a factory.)
 - Always something new in the shop, to re-convert loyal paying users as much as possible.



Time-Limited Events



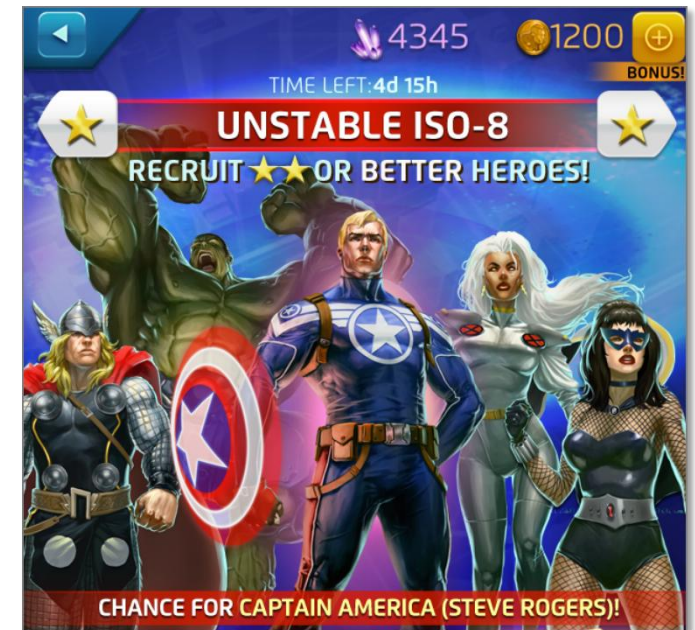
- Temporary mechanics or progression systems.
 - Resource systems must be *designed to accommodate rewards*
 - Time-limited resources / Specialized currencies
 - Using the same resources as the core loop would result in inflation.
 - Extra progression systems
 - Marketing windows for new content.
 - The same event can *rotate* after a few weeks / months.
- **Seasons** are the latest incarnation of this idea.
 - Longer intervals of time, months.
 - Enable developers to focus on bigger new mechanics.
 - Sometimes content sticks around in later Seasons.



Time-Limited Events

Other forms:

- Launch new content in windows of time.
- Resources that can only be earned in specific intervals of time.
- Materials that can only be obtained in a specific day of the week or the month.
- Can be tied in any existing Season/Ladder reset. Examples:
 - The monthly free rune unsocketing day in Summoners War.
 - Weekly Fusion Boosters in Dungeon Hunter 5.
- Have more control over a part of the player's Progression and the Metagame.



15-Minutes Break

Depth of Systems

- Depth around your main / most expensive assets will help on Live Ops sustainability and ROI.
- Horizontal systems, where assets have unique effects, require more frequent updates and expensive asset-creation.
- Vertical systems, where assets are stats-based, allow to extend the useful life of assets with high production cost attached (art, tech, design, balancing, QA)



Depth of Systems

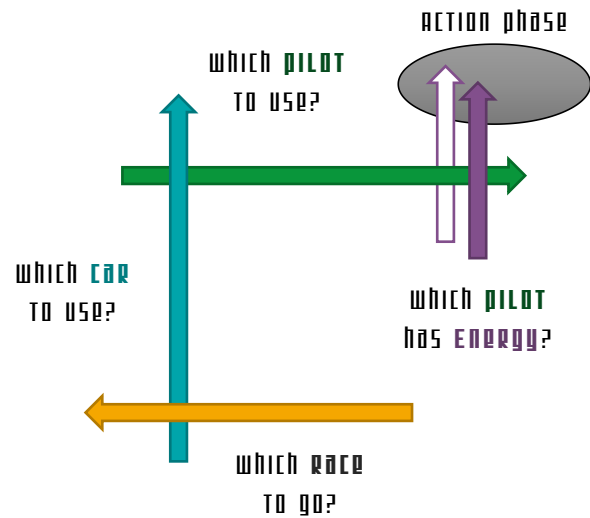
- Specialization systems opens up for more emergent meta game.
- Also the extra resources enable more varied rewards for different Events
- Reduces the risk of power creep or nerfs in updates.
- More orthogonal systems give designers more tools to solve imbalances in the gameplay



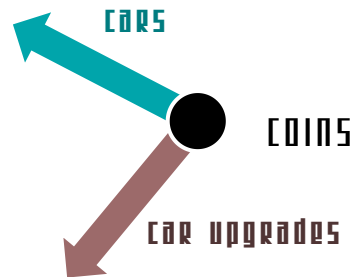
Depth in Decisions vs. Depth in Economy

angry birds go!

decisions in the core loop

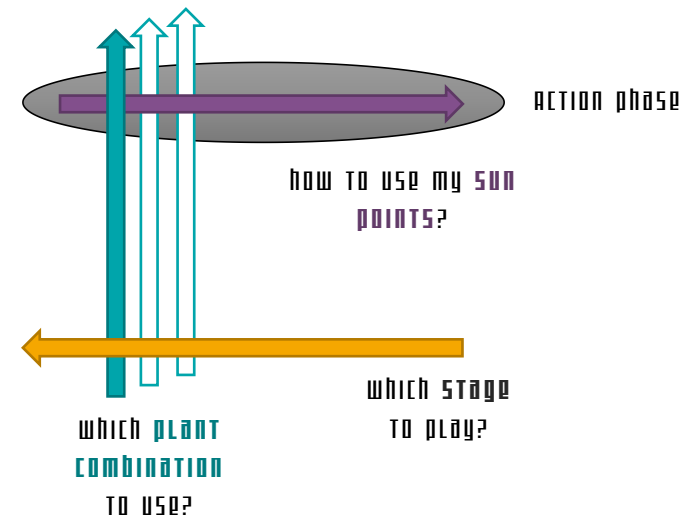


economic decisions

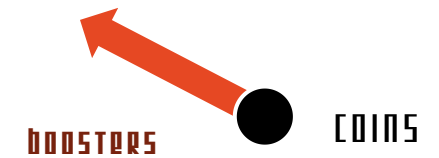


plants vs. zombies 2

decisions in the core loop

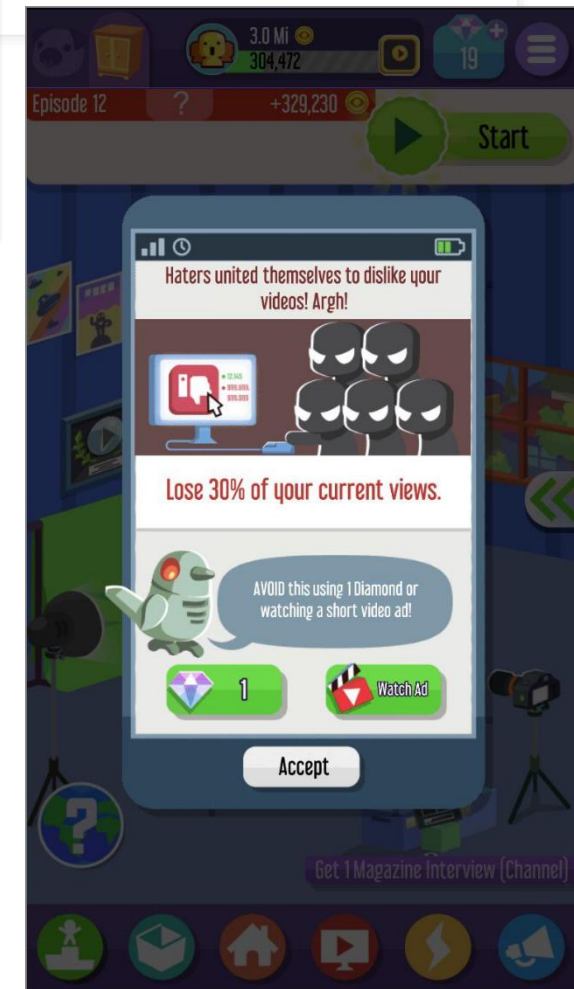


economic decisions



Ads

- Most effective forms:
 - Rewarded Videos
 - Playable Ads
- The Ad needs to be **non-intrusive** in the UI.
 - A UX feature enable the player to voluntarily click to see it.
 - Interstitials and banners are bad for game flow and few revenue
- To be effective, **Ads need to reward users**.
 - Give currency, "revive", time-limited bonus.
 - Anything that would charge a small amount of hard currency.
- Providers: Unity Ads, AdMob, Facebook, IronSource and others.



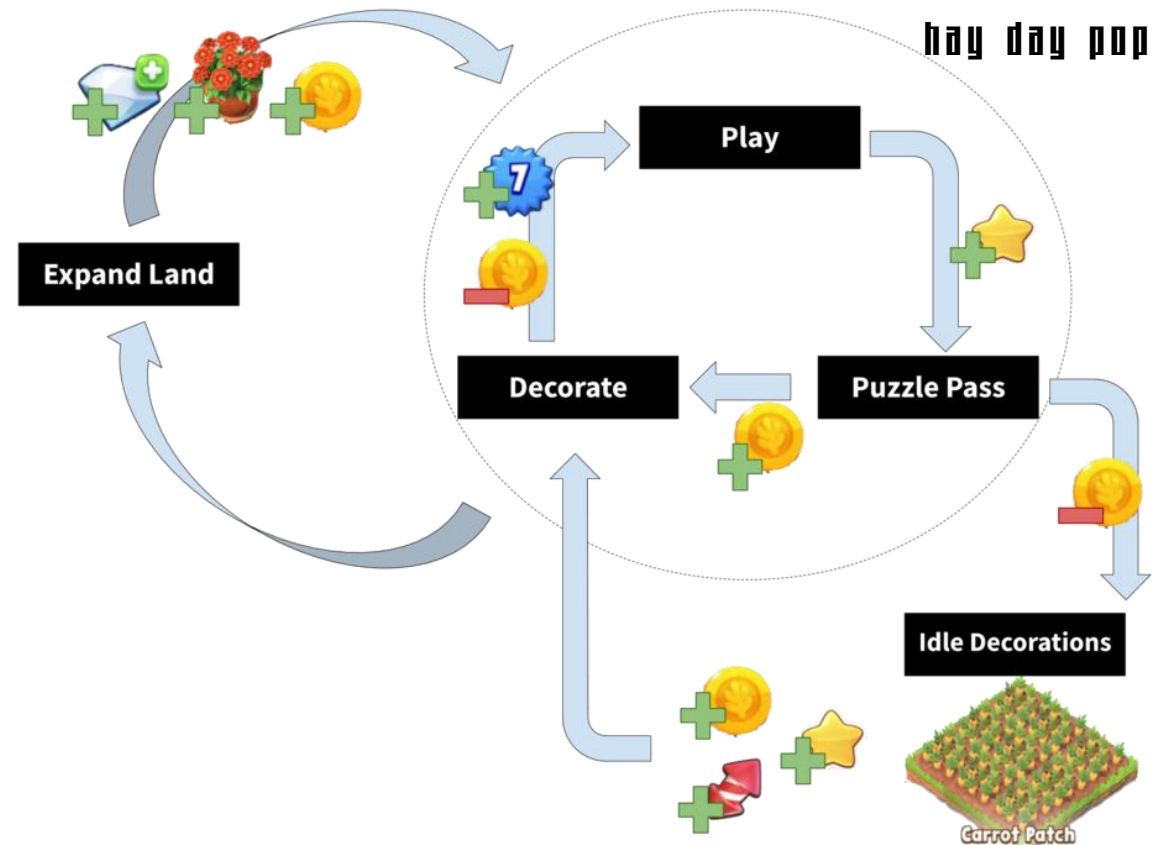
Designing Loops



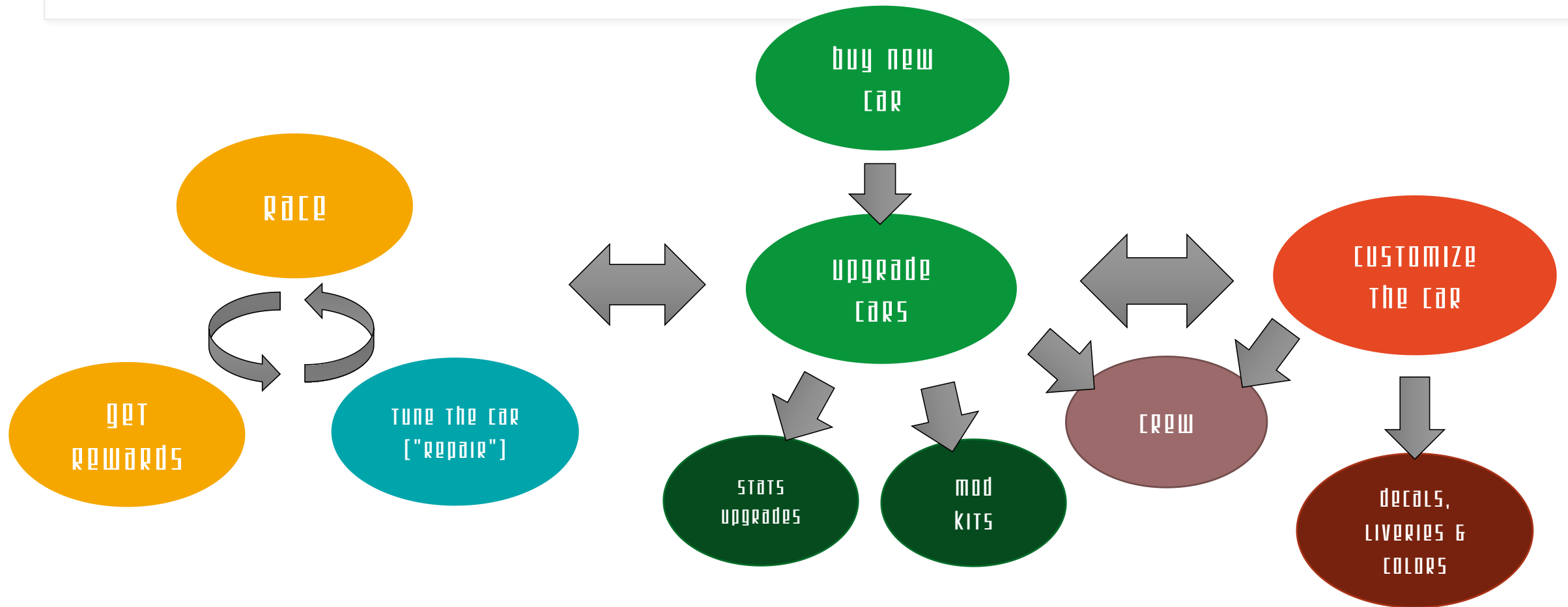
Connecting all different pieces

Game + Resource Loops

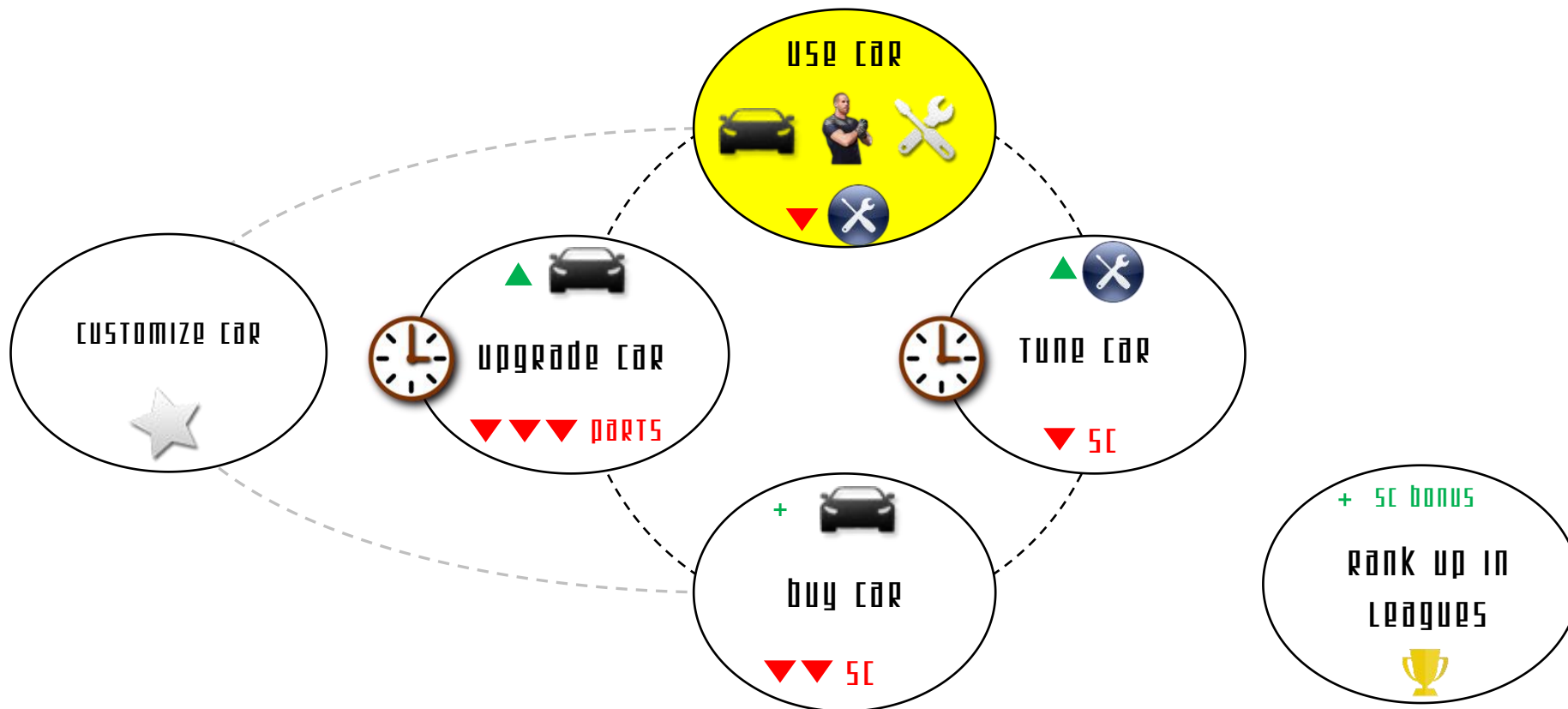
Township



From Scratch: A Racing Game



Inner Loop



DEFICITS:

▼▼▼ SC

▼▼▼ PARTS

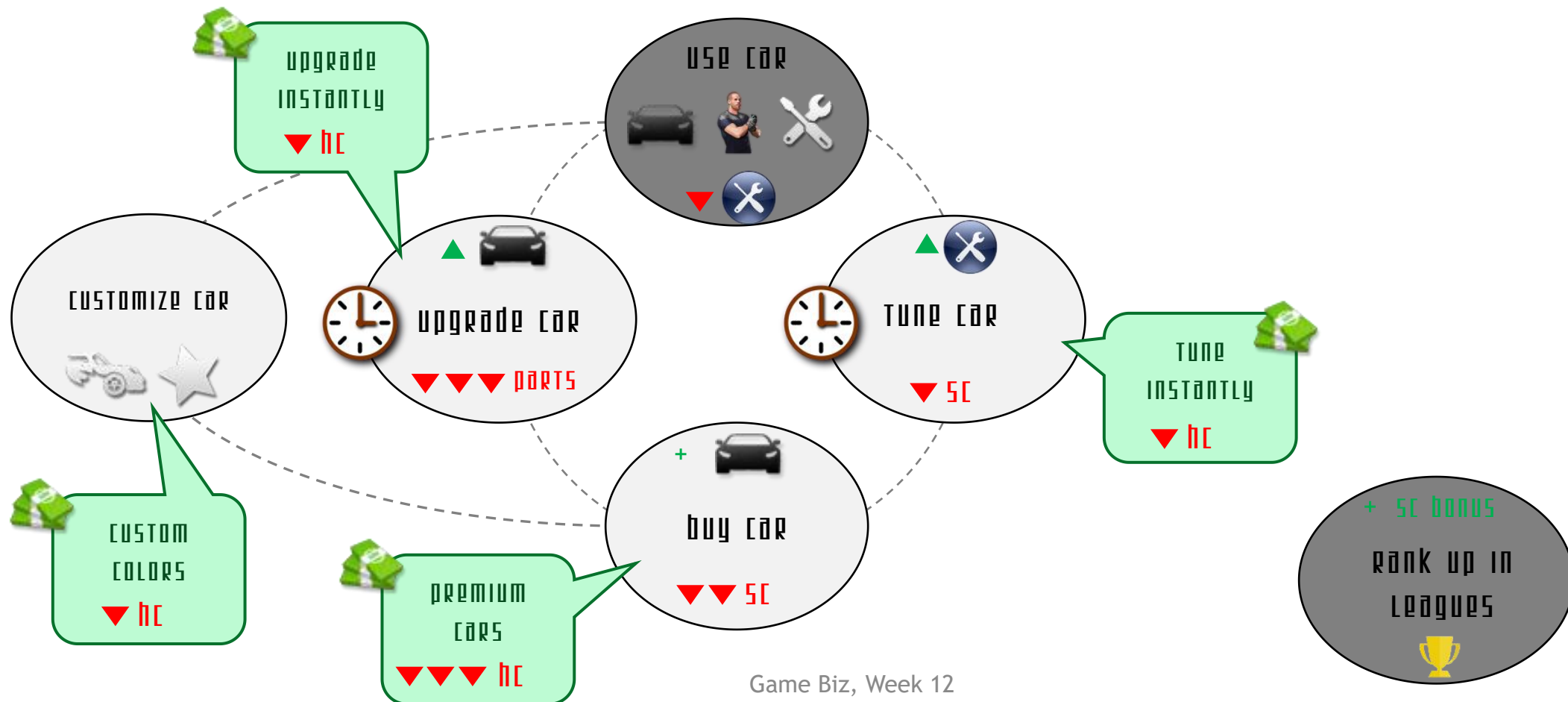
★ DECALS

👤 CREWMEN

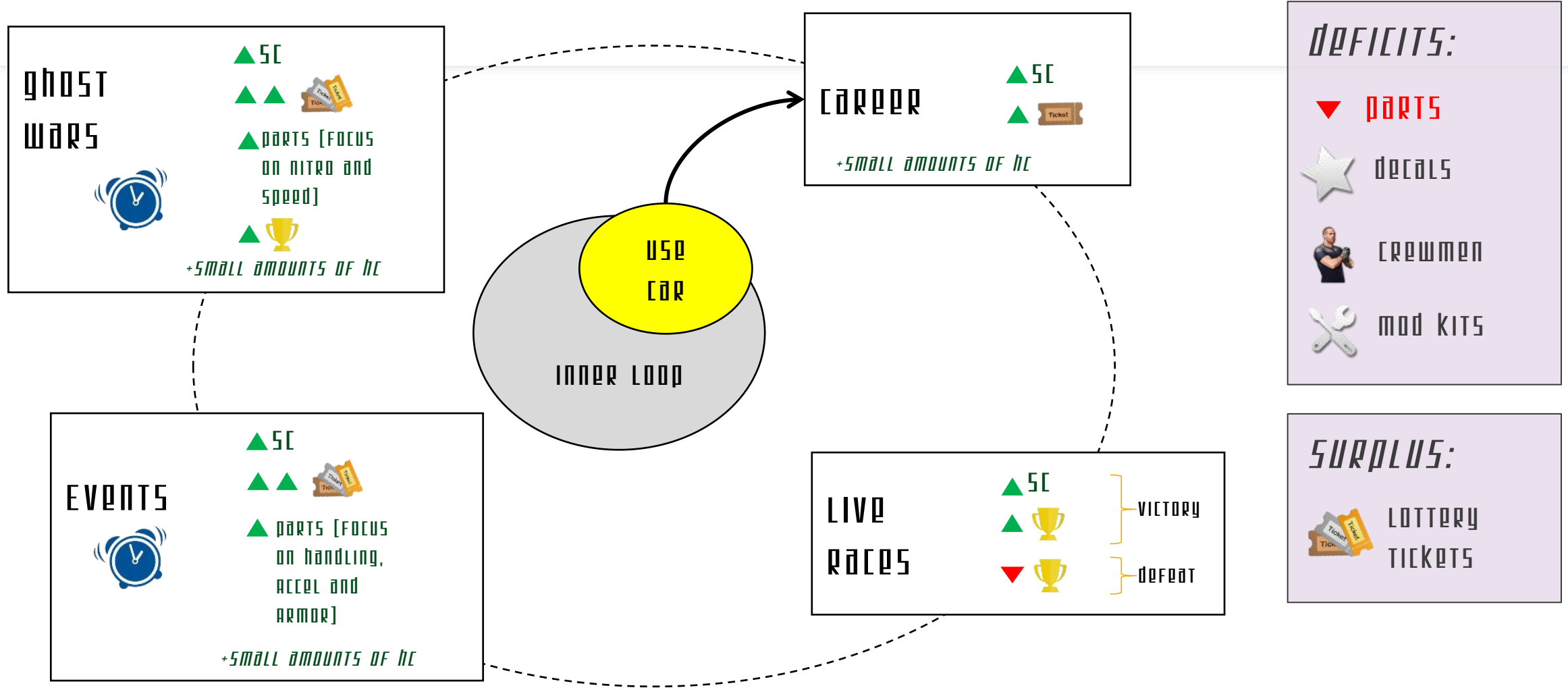
🔧 MOD KITS

🏆 LEAGUE POINTS

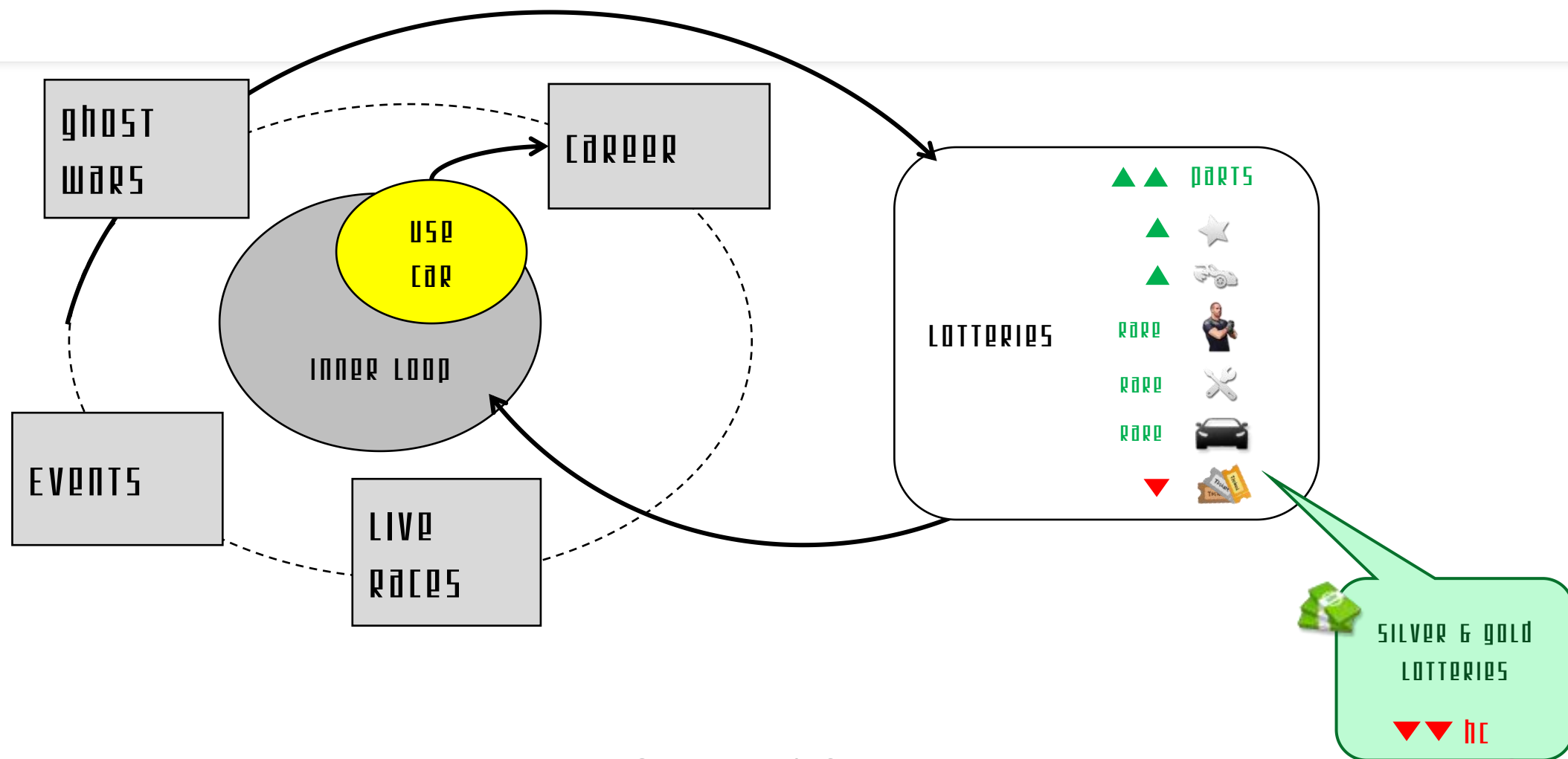
Inner Loop + Hard-currency hooks



Loop with Play Modes



Full Loop



Assignment



Q&A