The Ninth Annual Game Design Think Tank
Project Horseshoe 2014

Group Report: Progression Systems

Participants: *A.K.A. "Team Golden Cookie"*
Daniel Cook, Spry Fox
Joris Dormans, Ludomotion
Andrew Friedman, Amplify
Aki Jarvinen, Playdemic
Mario Izquierdo, Linden Lab
Crystin Cox, ArenaNet
Squirrel Eiserloh, SMU Guildhall
Link Hughes, ArenaNet
Ian Schreiber, Rochester Institute of Technology
Facilitator: Jenna Hoffstein, Little Worlds Interactive

Workgroup Objective

To investigate progression systems in pursuit of insights, tools, and frameworks with which to improve them.

Outcome

We identified characteristics and critical vocabulary to describe and classify various types and elements of progression systems in games, and best practices for implementing progression that players will likely find compelling, interesting, and worthwhile.

Problem statement

Progression Systems are a core motivational tool of game designers, and numerous well-examined examples have been typified in a wide variety of games and genres. It is therefore in the interests of game designers to better understand these systems, and embrace a set of best practices in order to be able to better and more deliberately craft them to evoke a desired set of player experiences. But what are these best practices? What types of progression exist? Which forms of progression are more likely to compel a player to play "just one more level" or "one more turn" because she intrinsically wants to? Which are more likely to be received only as a tedious "grind" that players grit their teeth and endure only as payment to access other, more enjoyable parts of the game? For that matter, what *is* a progression system, and does it contain component parts that can be separately discussed and analyzed? These are the questions addressed by this workgroup.

Overview

We tentatively defined "progression" as the perceived differences between iterations of some game loop that bring the player towards one or more goals.[1]

We identified four primary best practices for making progression systems that seem more compelling to players:

- **Investment**: progression systems are more engaging when they require direct interaction and effort from the player in order to move forward.
- **Movement**: progression systems earn more attention if players are aware of their current status, and changes in that status, such that they can perceive a sense of progress as well as understand how to make further advances towards future goals.
- **Anticipation**: use of proper pacing so that the player feels there are many steps forward that are each satisfying, with appropriate rewards for progression, can excite the player and make the progression feel meaningful.
- **Motivation**: the player should have some reason to choose to engage with the progression system. Designers should be aware of ways that player motivation can be enhanced (or eroded).

In the remainder of this report, we will first deconstruct the concept of a progression system into its component parts – which we call “progression atoms”, of which we identify eight primary types – along with the types of motivation that might drive a player to engage with these systems. With critical vocabulary in hand, we then zoom in to a single iteration on a progression system to suggest a model for analysis and understanding, and then zoom out to show how these iterations can link together, and offer some principles for what might cause progression systems to succeed or fail.

Types of Progression

What are the elements of progression systems and how do they relate to one another other?

Progression systems typically begin with a motivator which incentives the player to enter a progression arc (one-time progression from A to B) or progression loop (repeating cycle of progression); we refer to these collectively as progression atoms. Each individual atom has attributes that modify how it is experienced by the player. When completed properly, the progression atom typically provides a reward that (ideally) satisfies the original motivator.

There are many different types of progression, as identified below. These types are not mutually exclusive; a single game event may impact progress along several of these at once (for example, catching a new Pokémon works towards completion of the “Gotta Catch ‘Em All” accomplishment arc, as well as increasing character stats/capabilities, and possibly introducing new mechanics and lowering perceived difficulty).

We first subdivide progression into two broad categories: individual and social.

Individual progression

Affects the player or character relative to the game environment.

Progression arcs reflect unidirectional progress towards the completion of a segment of the game experience. We identified two basic types of arcs.

- **Accomplishment arc**: progression towards the completion of tasks, quests, collections, achievements, badges, etc. – i.e. filling up a progress bar (actual or theoretical) or crossing items off a to-do list.
- **Discovery arc**: progression towards experiencing the game’s narrative, traversing levels, exploring game areas, getting to know NPCs through dialogue systems, etc. – i.e. exploration.

Progression loops reflect progression that continues indefinitely, where completion of one loop may simply serve to enable or magnify the next iteration of that same loop. We suggest four types of progression loops.

- **Difficulty loop**: actual or perceived changes in the difficulty of the game over time, such as increased enemy health or damage, number or speed of foes, and so on.
- **Complexity loop**: unlocking or enabling new game mechanics and features (spells, powers, tools, etc.) that open up new gameplay options, styles of play, or strategies/tactics.
- **Power loop**: progression in the power of the player's avatar(s) via changes in character attributes or stats ("virtual skill").
- **Cosmetic content loop**: progression towards obtaining cosmetic or status items that do not have direct in-game effects.[2]

Social progression

Affects the player relative to other players.

- **Influence**: increase in followers, shares, likes, friends list, and similar ("Reach").
- **Ranking**: increase in standing in permanent or tournament leaderboards or other ranking schemes, whether global, local, or among friends.
- **Guild**: gaining entry to a player-created group, then gaining additional rank/responsibilities /privileges, and ultimately rising to a position of ongoing leadership and influence within the group and the overall player community.

Motivations for Progression

There are many reasons a player may wish to enter or advance a progression arc or loop. It is worth examining self-determination theory[3] in this context, which suggests a motivation can either be intrinsic (the player is motivated by her own internal drives and desires[4]) or extrinsic (the player is motivated by an external reward or punishment). Of note, introducing extrinsic motivators (carrots, sticks, or social pressure) can displace the more powerful intrinsic motivators; if the extrinsic motivation is later removed, the intrinsic motivation may be extinguished. Thus, designers should be careful about rewarding players for good behavior and punishing them for bad behavior; this works well to motivate players to do rote tasks, but long-term is actually anti-motivational for nontrivial tasks or goals.

As such, we theorize that intrinsic and not extrinsic motivators are more powerful (or, at least, more desirable), and that the best progression systems focus on the kinds of things that people are driven to do naturally, or are engaging and/or enjoyable in their own right.

We also noticed that each motivation seemed to have a corresponding reward: in other words, if the player has a certain type of motivation, it is because she is anticipating a corresponding payoff in the future. The motivation and the reward are just two sides of the same coin; the motivation causes the player to enter the progression system, and the reward is what they get when they complete an iteration.

Some examples of motivations and their corresponding rewards include:

Individual motivators:

- Control (reward: increased avatar power)
- Mastery (reward: increased player skill)

- Curiosity (reward: increased player knowledge)
- Individuality (reward: self-expression)
- Closure (reward: completion)

Social motivators:

- Relatedness (reward: inclusion within a group or community)
- Superiority over others (reward: competitive victory)
- Community value / leaving a legacy to others (reward: recognition)
- Watching the world burn (reward: seeing the results of a successful griefing)

Relationships Between Motivations, Types of Progression and Rewards

We believe there are some clear relationships in the previous two sections, suggesting that not only are motivators and rewards strongly linked, but that each type of progression is driven by a unique motivator, and yields a unique reward:

Motivator	**Progression Atom**	**Reward**
Control	Character Stats	Power
Mastery	Difficulty, Complexity	Human/Player Skill
Curiosity	Discovery Arc	Human/Player Knowledge
Individuality	Cosmetic Content	Self-Expression
Closure	Accomplishment Arc	Completion
Relatedness	Guilds	Cooperation, Social Inclusion
Superiority	Leaderboards	Competition, Recognition
Community value	Influence	Recognition
Look, some people are just jerks, okay?	(We rarely create systems specifically to support griefing. Rather, this behavior arises from systems that allow or enable it, among certain players – usually unintentionally.)	Griefing

Resistance Curves

What progression looks like on the micro scale.

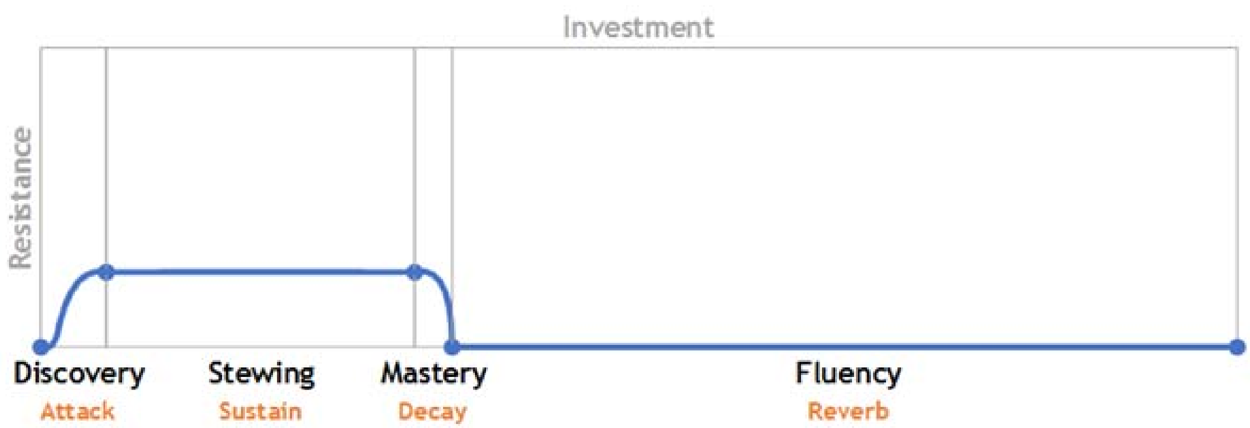
By its nature, progression necessarily requires that the player overcome some form of resistance. Otherwise, the rewards would be given to the player immediately, without her having to engage in any kind of progression loop. The resistance may come in many forms, generally requiring some combination of player skill, luck, and time investment in order to overcome. In this way, resistance can be considered as a generalization of the "difficulty" of the game.[5]

Progression can be represented, then, by a graph of resistance vs. investment, where investment is the generalization of player time and effort. In effect, this becomes a way to visualize the pacing of a game. An individual unit of resistance (a single challenge or puzzle found in a single iteration of a progression mechanic) goes through a rising and falling curve that manifests in a set of four stages. The length and amplitude of each stage varies, based not only on the type and complexity of the challenge, but also contextually and subjectively based on player skill, avatar power, and game state. Each of these units is a single “building block” out of which the larger progression curve of the game can be derived.

The stages of a resistance unit appear to relate closely to the traditional mathematical (or musical) envelope:[6]

1. **Discovery**: the process of revelation of the existence, nature, and scope of the particular problem space. During this period the player transitions from ignorance to awareness. Analogous to the “attack” of a musical envelope.
2. **Stewing**: the process of investing in the now-defined problem space, interacting with it, pushing its boundaries, cogitating upon it. Analogous to the “sustain” of a musical envelope.
3. **Mastery**: the period in which a player’s understanding of the problem approaches critical mass, and the pieces start falling into place. The process accelerates as the player unravels the problem until it finally gives way. Analogous to the “decay” of a musical envelope.
4. **Fluency**: after successful navigation of a challenge, the player internalizes the lessons learned, generalizing and abstracting them, and ultimately adding the new skill to her skill set / toolkit. Analogous to the “reverb” of a musical envelope.

As an example, here is what a resistance unit might look like for a simple **lock-and-key** mechanic.



Note that the resistance is never particularly high, because the process of searching for a key is not typically a great challenge in its own right per se. The Discovery phase here is short: it doesn’t generally take long for a player encountering a lock to realize she must search for a key. The Mastery phase is also short: once the player has the key she needs only return to the lock and use the key. The Stewing phase is longer, and consists primarily of exploring the area to search for the key. Also note the Fluency phase has no resistance; once the player owns a key, she can bypass all matching locks without further effort.

By contrast to the graph above, a resistance unit for mastering a new skill (say, learning to use a new unlocked unit in an RTS game) has a more articulated curve, creating a better feeling of progression in the player:

Here, Discovery - simply learning the nature of the problem - is itself an extended task, where the possibility space is not clear up front and the player must do some experimentation or exploration to find the boundaries of the new unit: how best to direct it, discovering its strengths and vulnerabilities, identifying contexts in which it is most useful, and so on. The Mastery phase is likewise extended, as it takes some practice and trial-and-error to learn the best uses of the unit (strategies and tactics, matchup advantages, avoiding weaknesses, etc.). Even in the Fluency phase, there is still some amount of resistance, as it requires some continued base amount of player skill to execute.

Resistance Projection

Players only perceive the resistance graph relative to their current trajectory, and extrapolate linearly what their progress will be. During the Stewing phase and the earliest part of Mastery, the projection is near-horizontal, potentially leading to player frustration. Midway through Mastery, the player is more able to see the light at the end of the proverbial tunnel. In most cases, though, completion is closer than it appears:

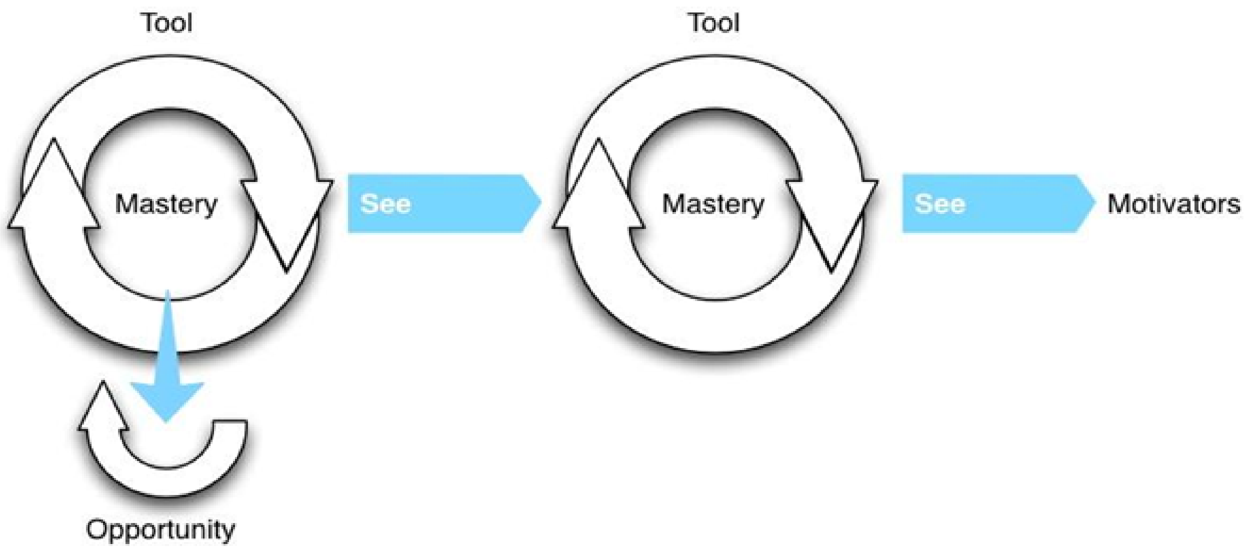
One possible takeaway here is that designers should be aware of the fact that players may perceive resistance as being much greater than it actually is, and thus the relatively flat "Stewing" period - especially a protracted one - can produce player frustration due to the lack of perceived progress (even though the actual rate of progress may be quite high). Consider seeking ways to put more of the challenge into the Discovery and Mastery phases instead.

Interaction Loops

Linking resistance units together to form Progression Systems.

As with many aspects of game design, player perception can diverge from reality. In the case where the player receives little feedback as to her progress, she may actually be getting closer to a goal but not realize it, and thus project even farther out. Designers should build sufficient UI communication into the game to give the player an understanding of when she is progressing.

Resistance curves are drawn from the perspective of measuring the player's progression through a challenge over time. Another way to envision this same progression is through the lens of the player's actions as she iteratively explores and masters the game's systems.



The player takes an action in the game, resulting in an execution of game rules that in turn yields feedback that helps the player update her mental model of the game, which in turn informs the player's next set of actions taken. For example, a player might learn how to time a stun and a smash attack to do much more damage than a single attack.

In addition, these interaction loops can generate in-game resources and tools that provide increased avatar power (what Jesse Schell refers to as "virtual skills"[7]). For example, a loop might result in the player's avatar earning a +5 sword that improves her ability to navigate the system.

Each loop contains an immediate goal that crystallizes as the player interacts with the system. Players are motivated to reach this goal, either intrinsically (players naturally seek mastery and increased agency) or extrinsically (in-game or out-of-game rewards). Note that this is a key differentiator for player enjoyment: individual loops can be pleasurable and motivating on their own; otherwise, if a player is just doing a current loop in order to advance some other loop, it's experienced as a "grind" (in the negative sense).

Individual interaction loops can be related to each other and feed into one another. From these linkages, a sense of overall progression through the game emerges.

Components of the system that a player is aware of at any given time:

- **The current loop**: what the player is doing Right Now.
- **Foresight in this loop**: the player's vision into the future outcome of mastering the current loop.
- **Next loop**: the interaction loop(s) that become available once the player gets better at the current loop. This is often fuzzy.
- **Future loops**: a series of additional steps the player will have to do, but does not have foresight into… yet.
- **Anchoring goals**: long-term goals that the player is aware of and desires. The player's base

motivation to go through all of the atomic interaction loops in the game (in addition to any added motivations to participate in any particular loop for its own sake).

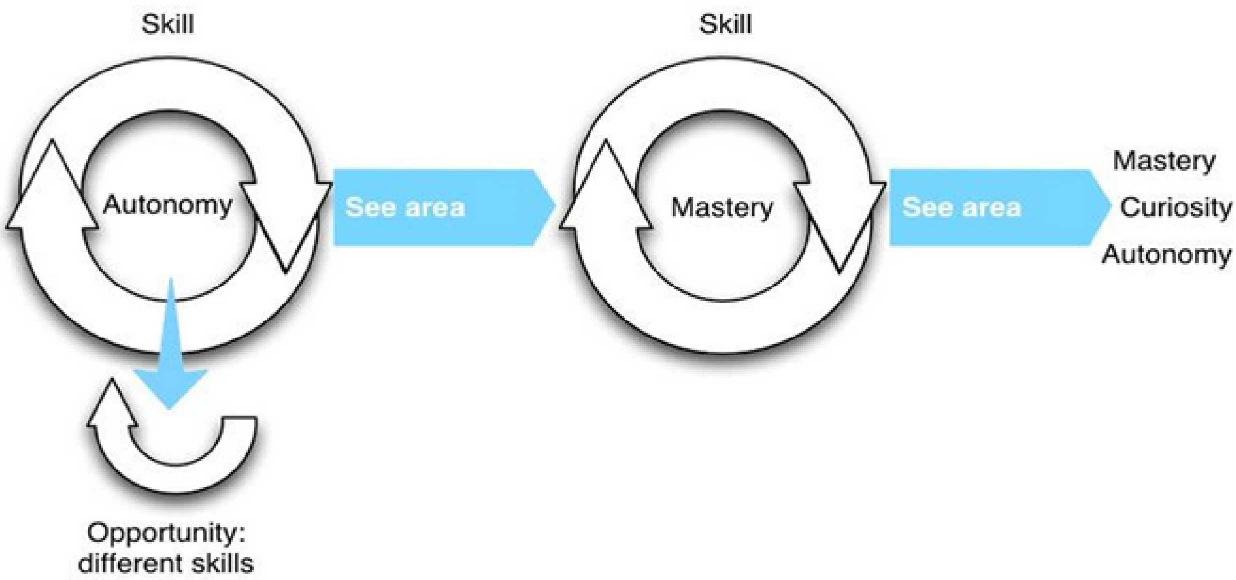
- **Exits**: ways for the player to leave the current loop (stopping their current behavior) to do something else. This may be to pursue other interaction loops in the game, although leaving the game entirely is always an option.

For example, the player's view of the present and future interaction loops could be like this:
(1) Steal Ed's pants, (2) ???, (3) Profit. Where step (1) is the current loop, step (2) represents the future loops, and step (3) is the anchoring goal.

Examples of interaction loops chained together

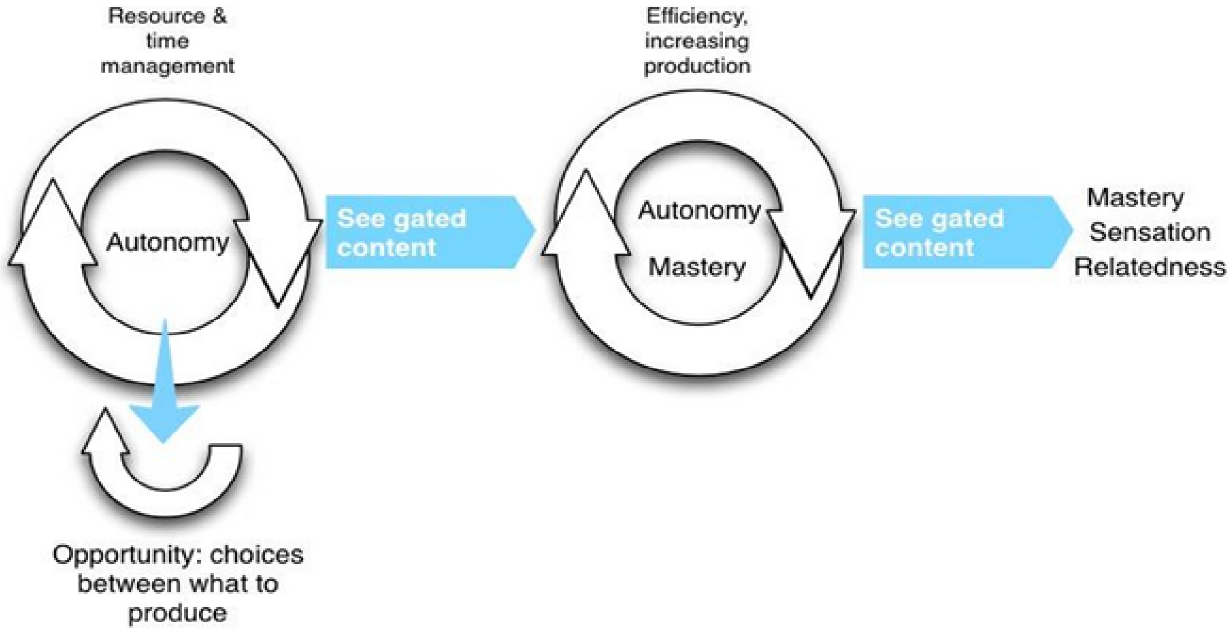
Zelda/Metroid Content Progression

In Zelda and Metroid type games, the player is motivated by a desire to gain access to new areas and new skills, which leads to a desire to master the most recently acquired skill or area. This loop is repeated with each successive skill/area throughout the game.



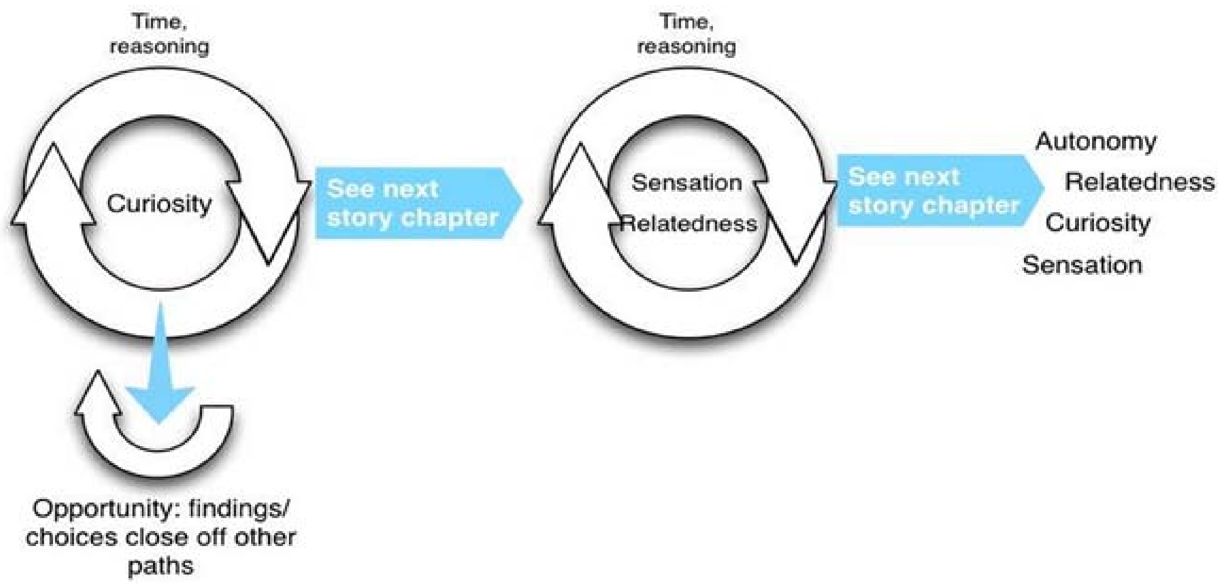
Progressive Resource Management

In a game with progressive resource management mechanics such as FarmVille or StarCraft, the player is motivated by the desire to have more options for resource acquisition, which in turn affords the ability to create an efficient engine. In the case of PvP games with an end condition, this continues until one player's engine ultimately allows them to dominate the game and triumph over their opponents for the win. In the case of games that have no defined endpoint and only a transition to some sort of elder game, the loop repeats until a player chooses to exit the loop, either out of boredom with the mechanic or due to sufficiently diminishing returns.



Narrative-driven Progression

In narrative-driven games such as Walking Dead or Heavy Rain, the player is motivated by seeing the story develop and resolve. On successive iterations of the loop, the player gains a greater understanding and relatedness to the characters.



Some things worth pointing out about these examples:

- Vastly different games at their core can still share similar progression models.
- Even single interaction loops may have several motivators.
- Progression of entirely linear narrative (or gameplay) is not an ideal fit for this model, as interaction loops assume some amount of player agency and motivation to drive them forward.
- As a design exercise, putting your game's progression design into this model can help in refining the design, spotting potential issues, or pinpointing the source of known issues.
- This model is scalable and can operate at any level of abstraction, so it can be used to dig deeper into player motivations for the most granular parts of a design.

Note that executing an interaction loop has a cost to the player. This includes player costs (cognitive and time costs spent to execute the loop), sometimes resource costs (spend of in-game resources to interact with the loop), and opportunity costs (choosing to not interact with other loops – or other activities outside of the game – while progressing through the current loop).

A player's costs become a kind of investment in the system over time; inside the player's brain at the faster-than-conscious-thought level, is an economic calculation being evaluated constantly that takes into account both perceived future payout of the current interaction loop and the sunk cost of previous actions, versus alternate actions that will yield more than continued investment. Choosing to continue along the current path or switching to an alternate strategy or tactic is the beating heart of many interesting game designs.

Completion and Closure

The Russian psychologist Bluma Zeigarnik noted that people remember incomplete tasks more easily than completed tasks. This Zeigarnik effect[8] (what gamers and game designers know as "completionism") involves players being motivated to complete known unfinished tasks, feeling a sense of persistent cognitive load for as long as they do not complete a task, and experiencing a sense of relief when completing a long-standing task.

Sid Meier's Civilization is well known as a highly compelling game; players may sit down for just a few minutes, only to find that an entire evening has passed because they kept playing "just one more turn" ad infinitum. In the context of progression and interaction loops, we can gain insight into how and why this happens.

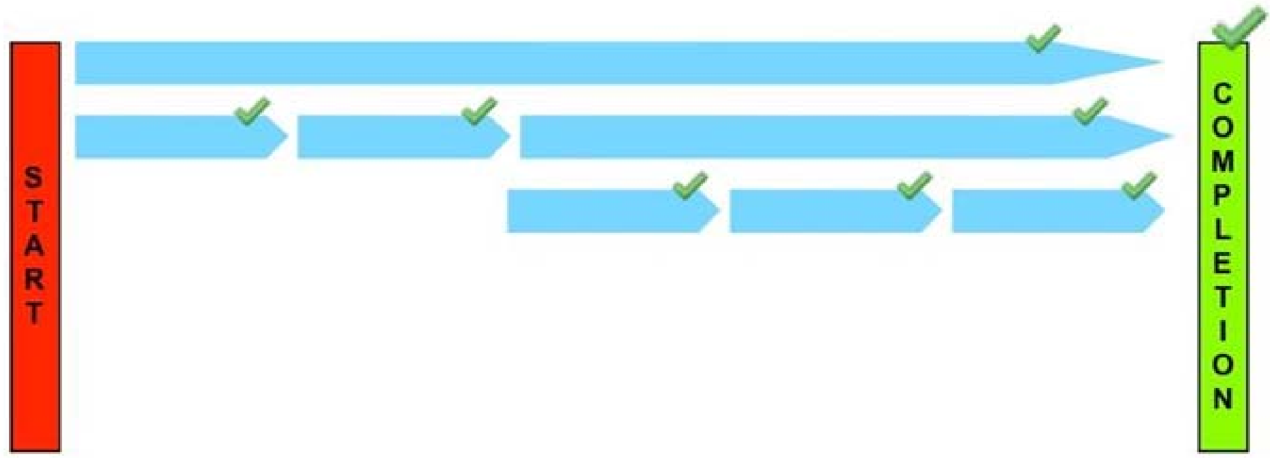
Civilization is designed to have many interaction loops active at any point in the game. Your research is about to complete; a stack of units is about to get to an enemy town; a building in one of your cities is about to finish; and so on. Some loops are short-term and others are long-term, but the pacing of the game is such that these loops share two qualities:

1. They are open loops; they are not complete, yet, and completing a loop just tends to open up the next iteration of that same loop so that it remains open indefinitely.
2. They are overlapping; the player is almost never more than a turn or two away from completing one or more goals.

Hence, a Civilization player is compelled to take one more turn just to close a loop in order to save themselves the cognitive load of having an almost-but-not-quite-completed goal or task, only to find yet another near-complete goal to complete on the next turn, until the entire game ends or the player is forced to walk away against their cognitive desires.

As if this were not enough, the pattern of feedback for the closing of interaction loops is varied so that it acts like an intermittent reinforcement schedule, the most compelling of known psychological conditioning models.[9]

By contrast, total completion is when all open loops in a game close down at the same time. There can be design benefits from planning the game to have natural points like this. The ongoing cost of maintaining active skills and working knowledge about the game drops off dramatically, allowing players to "let the game go," freeing up their mental resources and allowing their cognitive load to drop dramatically. There is a wonderful feeling of completion on the part of the player. It provides a natural breakpoint for the player to step away, in order to deal with other life tasks outside of the game. If these moments are spaced closely enough together, players will not feel that the game is holding them hostage.



Open loops that lead into other loops prevent a sense of total completion. The solution to this is to synchronize the various loops in the game so that they all come to a climactic conclusion at about the same time.

It is worth distinguishing completion from closure. Completion is when one or more interaction loops are finished. Closure is a feeling of the player that part or all of the game has come to a satisfying conclusion. Completion is all about moving forward; closure is about leaving things behind. For completion to generate closure, it is important that the player have the opportunity to look back on the play experience so far. Closure is facilitated by longer interaction loops, and by closing off possibilities of revisiting earlier parts or phases of the game, either through a sense of loss (the player grieves over a lost party member) or a sense of relief (player realizes they have grown to the point that they do not need a certain thing anymore). The holy grail here is to provide players with a sense of closure (i.e. an interesting play session or episode) even if the player fails to accomplish their goal.

Random Thoughts

During our time at Horseshoe, the group discussed many ideas that couldn't quite fit into the main document, but are still worth mentioning. Those are some of those thoughts, in no particular order:

Non-meaningful choices can still feel like meaningful interactions

For example, in "Cookie Clicker", or "Dear Esther", pacing and progression can act as a substitute for player agency and meaningful choice.

Increased player agency can still improve the meaning of the interaction. In Cookie Clicker, the player experiences the same progression model with or without the Golden Cookies that appear on the screen occasionally; but their presence greatly improves player retention.

Attributes of Progression Systems

We identified two attributes that could be used to modify the experience of a particular progression system. One is the *acceleration rate*: how fast does the player progress, and in what pattern (e.g. linearly, with a positive or negative feedback loop, and so on).

The other is the *juiciness of the system*: the type and intensity of sensory feedback (e.g. change in music, particle effects, screen shake, or other audio or visual effects that happen during progression).

There is not necessarily a "correct" acceleration rate or juiciness that applies to all systems; rather, the acceleration affects the pacing of the game, while the juiciness affects how much emphasis the game places on the system.

Disagreement over whether ProgressQuest is an example of progression

After getting sucked into a debate on this that lasted far too long, we agreed that (1) it is best described as an edge condition, and (2) it was not such a compelling experience that we should hold it as any kind of paragon of the field. There are better games to study as positive examples for progression mechanics.

Slot machines are an interesting case

Slot machines are mathematically balanced to trigger a false sense of progression in players. An example of this is near-misses on a slot machine that make the player feel like they're "getting closer" to hitting the jackpot. This was also seen in Destiny, where a loot drop that fails to give the player something better than what they have doesn't actually bring the player any closer to their goal in any way; and yet, it can give a perception of progress, in the sense that a player may feel they have to get an unknown number of drops before they get the next useful drop, and each "failed" drop is one more closer to their goal.

However, this false feeling of progress is not guaranteed. Hope is a necessary component when the outcome is partly random. If the perceived chance is too low, the player may feel a sense of futility.

MMOs tend to get around Destiny's issue because there is a sense of fairness in social groups ("Need vs. Greed"), so players who have not received useful drops in awhile may be able to get one from their guild.

Ability to trade between players reduces the feeling of progression. In Guild Wars 2, any item can be bought or sold on the trading post; rare items might cost more gold, but all feel attainable to players. However, there is no perceived movement towards a goal here; there is no granularity, you either have enough to buy the thing you want or you don't, so there isn't a compelling resistance curve here.

We also encountered an interesting paradox: both randomness and control can satisfy a player's desire for certainty. "I want lots of gold so I get better at the game and improve my drop rate" vs. "I'll grind and wait for the RNG to give me sufficient amounts of loot, eventually."

Randomness on its own isn't a motivation, but it is related to motivation

The underlying motivation here might be curiosity, surprise, anticipation, a desire to experience something new, or the gaining of knowledge rather than skill. Randomness is something a player must submit to; progression is something a player takes control of.

Perceived challenge of a game

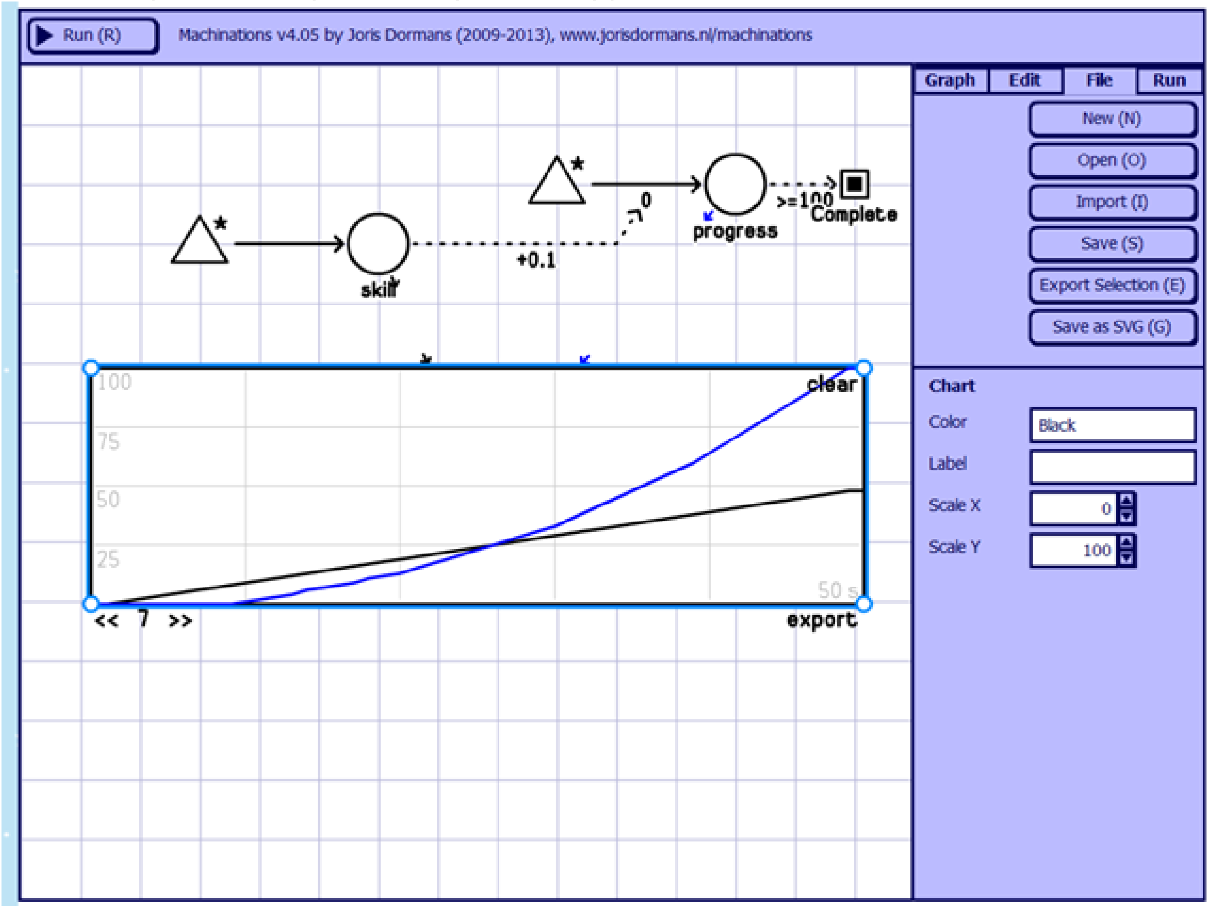
The perceived challenge of a game is a function of the player skill, avatar power, skill requirement of the challenge, and power level of the challenge. Beating a boss may be perceived as easier if the player becomes more skilled at the game, or if the player's avatar gets a damage boost; it may be perceived as harder if the boss is given extra attack or movement patterns that require a greater degree of skill to avoid, or if the boss is granted more health or does more damage. Perceived challenge does not necessarily change at a constant rate over time; it can go up or down.

Beyond progression systems

There are many, many things in games that are pleasurable or meaningful to players that are not related to progression at all. While this workgroup report is a deep dive into the topic of progression systems in games, keep in mind that this is just one tiny element of game design, which may or may not be appropriate to use on any given project.

Areas for future work (New Game +)

- We found Joris Dormans's Machinations[10] to be a highly useful tool for modeling the flow of resources in a game, and suspect it can have value in modeling progression systems as well, by correlating the characteristics of resistance graphs with the structure of the underlying mechanics. An oversimplified example of one possible approach:



- In many games that have progression as a primary focus, the player finally perfects their own progression engine in game, right around the time they win and the game ends. Deckbuilding games like Dominion are often guilty of this, as are many CRPGs such as the Final Fantasy games. It can often feel anticlimactic, and this is a design problem in need of better solutions in the future.
- Some games (notably but not exclusively MMOs) transition from their progression mechanics to an "elder game" where progression loops either stop, or shift drastically in their nature. It is worth exploring best practices for maintaining player interest after total completion.
- Is there a way to define, measure or compare progression between games? What makes progression in one game "better" than another?
    - Player retention in multiplayer games, or percentage of players who finish a single-player game, are useful metrics. However, they are probably not something that can be relied on exclusively; it is possible to make a punishing game that has a high level of retention through the threat of pain if players leave.
    - We compared the progression in Diablo 2 and Diablo 3, for example, and suggested that 2 felt better because it had a big social component, and interactions with the player community acted as a primary motivator to progress through the loops at the endgame. In 3, by contrast, it was too easy to either buy the best loot at the auction house or receive it in-game as a drop; since the game was flooded with epic items (at least on initial launch) the progression felt less motivating since it seemed too trivial to progress quickly.
- One of the most challenging aspects of designing progression systems is to design them in such a way that failure is still interesting.
    - One common design pattern is to let the player keep their progression after failure, e.g. a player who is defeated in an RPG can restart from the last save point but with all XP and loot they had accumulated up to their death. The design challenge is to do this in such a way that

overcoming a challenge doesn't feel meaningless: if a player can just keep bashing their head into a brick wall and eventually pound their way through, progression can feel like a function of time more than player skill.

- An interesting pattern, seen in some roguelike-likes such as FTL and Dungeon of the Endless, is to unlock more (sometimes better, sometimes just more varied) starting options for the next playthrough. This addresses the issue of losing all progress in a game with permadeath, and still gives a sense of overall progress even if the player lost. This method may be particularly suited to games with a lot of randomness where progress could otherwise feel futile.

1. As well understood as the term "progression" may be to most game designers, we found that actually trying to give a working definition was challenging. Even within our group, the definition presented here was controversial, but we chose to focus primarily on best practices and not definitions, so we moved on.

2. Note the difference between items earned as rewards, and those that are just given (e.g. earning a new hat from a loot drop, vs. earning a new hat given out to all players as a holiday present). Either one can feel good, but the latter requires no motivation and is not part of an ongoing loop.

3. For more information, see http://www.youblisher.com/p/7435-Self-Determination-Theory/ and http://www.selfdeterminationtheory.org/SDT/documents/2006_RyanRigbyPrzybylski_MandE.pdf

4. Some types of intrinsic motivators described here: http://www.leadership-central.com/types-of-motivation.html#ixzz3IUgOsfXU. And factors that promote these intrinsic motivations here: http://education.purduecal.edu/Vockell/EdPsyBook/Edpsy5/Edpsy5_intrinsic.htm.

5. Note that we view the concept of "resistance" very broadly: exposition in a story can be framed as resistance to plot advancement, for example. Lock-and-key type puzzles also offer resistance without necessarily any additional challenge. There is a huge opportunity for games to look carefully into other sources of resistance other than difficulty. Still, difficulty is the most common source of resistance in games, and the two can be thought of as equivalent for most practical purposes.

6. http://en.wikipedia.org/wiki/Synthesizer#ADSR_envelope

7. Schell, J. (2008). The Art of Game Design: A Book of Lenses. Burlington: Elsevier.

8. http://en.wikipedia.org/wiki/Bluma_Zeigarnik#The_Zeigarnik_effect

9. http://en.wikipedia.org/wiki/Reinforcement#Intermittent_reinforcement

10. http://www.jorisdormans.nl/machinations