Integrating Emergence and Progression

Joris Dormans
Amsterdam University of Applied Sciences
Duivendrechtsekade 36-38
1096 AH, Amsterdam, the Netherlands
+31 20 595 1686
j.dormans@hva.nl

ABSTRACT

This paper investigates how structures of emergence and progression in games might be integrated. By leveraging the formalism of Machination diagrams, the shape of the mechanics that typically control progression in games are exposed. Two strategies to create mechanics that control progression but exhibit more emergent behavior by including feedback loops are presented and discussed.

Keywords

game design, emergence, progression, level design

INTRODUCTION

Games are complex rule based systems that exhibit many emergent properties on the one hand, but must deliver a well-designed, natural flowing user experience on the other. These two aspects of games, often referred to as "emergence" and "progression" respectively (Juul, 2002, 2005), are generally considered be two different ways of creating gameplay challenges. Put simply, with emergence relatively simple rules lead to much variation in gameplay, whereas with progression many predesigned challenged are ordered sequentially. According to Jesper Juul, "emergence is the primordial game structure" (Juul, 2002, 324); and is the result of the many possible outcomes made possible by the rules in board games, card games, strategy games and many action games. Games of this type can be in many different states: the displacement of a single pawn by one square in a *Chess* game can make a huge difference. The number of possible combinations of pieces on a *Chess* board is huge, yet the rules easily fit on a single page. Something similar can be said of the placements of residential zones in the simulation game *SimCity* (Maxis Software, 1989) or the placement of a units in the strategy game like *Starcraft* (Blizzard Entertainment, 1998).

Progression, on the other hand, relies on tightly controlled sequence of events. Basically, a game designer dictates what challenges a player encounters by designing levels in such way that the player must encounter these events in a particular order. The use of computers to mediate games have made this form possible. Progression requires that the game is published with a lot of content prepared in advance, for board games this is inconvenient. As such, progression is the newer structure, starting with the adventure games from the seventies. In its most extreme form, the player is railroaded through a game, going from one challenge to the next or failing in the attempt. With progression the number of states is relatively small, and the designer has total control over what is put in the game. This makes games of progression well suited to games that tell stories.

Most modern games fall somewhere between games of emergence and progression. *Grand Theft Auto: San Andreas* (Rockstar North, 2004) has a vast open world, but also a mission structure that introduces new elements and unlocks this world piece by piece. In the story-driven first-person shooter game *Deus Ex* (Ion Storm, 2000) the sequence of levels motivated by a predesigned storyline dictates where the player needs to go next, but players have many different strategies and tactics available to deal with the problems they encounter on the way. Pure games of emergence and pure games of progression represent two extremes on a bi-polar scale, but most games have elements of both. Yet at the same time, emergence and progression are presented as two alternative modes of creating challenges in games, that might co-exists in a game, but are hard to integrate. This paper questions this perspective and seeks strategies to merge structured level design and emergent, rule-based play more effectively.

One trajectory towards an answer is that emergent behavior thrives somewhere on the border of chaos and order (cf. Salen & Zimmerman, 2004, 155). A true chaotic system will seem random and meaningless to most observers, whereas in games it helps if the player can make sense of what is going on. Where rules push games towards chaos by introducing dynamic behavior, levels pull games back towards order by imposing structure. If games are pulled too far back, they become games of progression where the spatial structure dominates the rules and little dynamic play remains.

This paper investigates what structures in rule-based systems directly contribute to a game's emergent behavior and how these structures can be integrated better with the structures that are typically used to control players' progress through the game.

UNDERSTANDING EMERGENCE

The use of the term emergence in games, which predates Juul's categories (for example Smith, 2001) is often in reference to the use of the term within the sciences of complexity. There it refers to behavior of a system that cannot be derived (directly) from its constituent parts. In games, as in any complex system, the whole is more than the sum of its parts. While the active agents or active elements in a complex system can be quite sophisticated in themselves, they are usually represented with rather simple models. Even when the study is about the flow of pedestrians in different environments, great results have been achieved by simulating them with only a few behavioral rules and goals (Ball, 2004, 131-147). Similarly, the elements that make up games are can be a lot more complex than the elements of a typical system studied by the science of complexity, but at least some games (such as *Go* and *Chess*) are famous for generating enormous depth of play with relative simple elements and rules. The active substance of these games is not the complexity of individual parts, but the complexity that is the result of the many interactions between the parts. The particular configurations of elements into complex systems that contribute to emergence in other systems also cause interesting gameplay. In other words: gameplay is an emergent property of a game system defined by its rules. For game designers this means that understanding the structural characteristics of emergent systems in general, and in their games in particular, is essential knowledge.

In a study of emergence, Jochen Fromm builds a taxonomy of emergence that consists of four types of emergence (types I, II, III and IV). These types can be distinguished by the nature of communication, or feedback, within the system (Fromm, 2005). Feedback is created when a closed circuit of communication exists within a system; in effect, when the current state of a particular element directly or indirectly effects the state of the same element later on. Feedback is considered to be positive when these effects strengthen

themselves, as is the case with guitar feedback where strings are vibrated to produce sound, and amplification of the sound causes the strings to vibrate in turn. Feedback is

considered to be negative when the effect dampens itself. A thermostat is a typical example, a thermometer detects the temperature of the air, when it is to low it will activate a heater, the heater will then cause the temperature to rise which in turn will cause the thermostat to turn of the heater again. In this way negative feedback is often used to maintain balance in a system (DiStefano III et al., 1967).

It has been recognized that feedback plays a role in emergent behavior for games, too. During his 1999 lecture at The Game Developers Conference Marc LeBlanc introduced feedback loops to the game design world (LeBlanc, 1999). Since then, feedback loops have been discussed by a number of influential designers, including Salen & Zimmerman (2004), Adams & Rollings (2007) and Fullerton (2008). A classic example of feedback in games can be found in *Monopoly* (Darrow & Parker, 1933) where the money spent to buy property is returned with a profit because more property will generate more income. In this case the feedback loop is positive because investing money will generate more money. Positive feedback can be applied to 'positive' game effects but also to 'negative' game effects, as is the case with loosing pieces in *Chess*, which increases the chances of losing more pieces, and which will eventually make you lose. LeBlanc suggests that positive feedback drives the game to a conclusion and magnifies early successes (see also Salen & Zimmerman, 2004, 224–225). This is certainly the case for *Monopoly*: a player that by chance gets more money or good property early in a game is very likely to win.

Negative feedback stabilizes a game by diminishing differences between players, by applying a penalty to the player who has done something that takes him closer to his goal and winning the game, or by giving advantages to the trailing players. A lot of racing games use negative feedback to keep a race close, either by giving trailing players more advantages or by hindering leading players. This effect is often described as 'rubber-banding'. It can be implemented by blatantly giving trailing players better acceleration and more grip, or as is the case in *Super Mario Kart* (Nintendo, 1992) by having the most effective weapons in the game affect cars in front of the player that uses them. LeBlanc points out that in most multiplayer games that allow direct interaction some sort of negative feedback is already in place, as rational players will target the leader more than any other player. As one might expect, negative feedback can prolong a game and magnifies late successes.

The concept of a game's *internal economy* (Adams & Rollings, 2007) to model activity, interaction and communication between game parts within the game system, helps to further understand feedback loops and emergent behavior in games.[1] A game's economic system is dominated by the flow of resources. In games resources can be anything: from money and property in *Monopoly*, via ammo and health in a first person shooter game, to experience points and equipment in a role-playing game. Even more abstract aspects of games, such as player skill level and strategic position can be modeled through the use of resources. A game's internal economy consists of these resources as well as the entities or actions that cause them to be produced, consumed and exchanged. In general, a game starts to exhibits emergent behavior when somewhere between two and four feedback loops operate in the game's internal economy (Dormans, 2009).

UNDERSTANDING PROGRESSION

Despite the importance of mechanics and emergence in games, no professional game designer can turn a blind eye towards the level design and the way levels control players' progression through the game. To subject yourself to game rules is to cross the boundary of the magic circle and to immerse yourself in the game's fictional space. Within that space the player starts to explore the game and its possible states. The number of rules, interface element and gameplay options of a modern retail video game is usually larger than most players can grasp at once. Even smaller games found on the Internet frequently require the player to learn a multitude of rules, to recognize many different objects and to try out different strategies. Exposing a player to all these at the same time can result in an overwhelming experience, and players will quickly leave the game in favor of others. The best way to deal with these problems is to structure the game experience with clever level design that teaches the player the rules in easy-to-handle chunks. In many cases games include special tutorial levels to introduce a player to the core concepts, and even then they will introduce new concepts with care.

The use of tutorials and level design to train the player is an illustration of one of the strengths of the medium of games: the use of game space to structure player experience. Unlike literature or cinema, which are well suited to depict events in time (histories), games are well suited to depict space. Henry Jenkins places games in the tradition of spatial stories, an honor they share with traditional myths and hero's quest as well as modern works by J.R.R. Tolkien (Jenkins, 2004). Simply by traveling through the game space, a story is told. A similar sentiment is found in Ted Friedman's 1999 essay on *Civilization* (Meier, 1991) where the drama of that game directly stems from the players journey through and conquest of a virtual world.

Daniel Cook's skill atoms constitute one of the most concrete theoretical perspectives on this aspect of level design (Cook, 2007). He analyses the individual steps a player goes through in learning a new game skill, and the way individual skills are hooked up into chains. Once the design team has decided on the final mechanics to be included in the game, levels can be structured in such way that the player is taught these mechanics. The most straightforward approach is to spread out the chains of skill over the level and to organize the level accordingly. In this case the chain of skills is integrated into the level's mission structure, and related to the level's spatial layout. However, levels are not there to teach the player the required skills only; there is usually more to a level than just a tutorial. Levels are also structured to facilitate exploration. Once the player has learned the basics of playing a particular game, levels provide the player with opportunities to display their mastery. During this stage, the mechanics become a means towards the goal of exploring the level or completing an interactive story.

This structure of introducing game mechanics gradually and have these act as locks and keys can be found in a very pure form in certain smaller independent games. *Knytt Stories* (Nygren 2007) and *Robot Wants Kitty* (Hamumu Software 2010) are good examples of such games. Both of these games are platform games where the player's goal is to reach a particular location (even though these games' story might frame it a little bit differently). Both, basically consist of one large level where the player gathers a number of power-ups that act as locks and keys. But also the challenges the player meets get progressively more difficult. Where, for example, the double jumping ability allows the player to jump longer distances in both games, the gaps the player needs to jump across do get wider, and the penalty for failing a jump increases from, having to replay a little part of the level, to dying and/or replaying longer parts of the level.

Lock and key mechanisms such as the power-ups encountered in *Robot Wants Kitty* or *Knytt Stories* are typical for action-adventure games and many other games of progression (Ashmore & Nietsche, 2007). Many of these games disguise their locks and keys, or have them serve dual functions. A key might be used as a weapon and way to unlock a new area. Keys might involve the mastery of a skill in order to use it effectively. Conventional design wisdom dictates that it is generally better to have players encounter these lock before the key. There are three reasons for this. 1) When keys are encountered first, players will simply be forced to collect everything they encounter without discrimination, which makes rather simplistic gameplay. 2) With obstacles and items that act as locks and keys but are represented with something else, it is easier to recognize the key if players know what the lock is; players then usually realize where they can proceed; they will actively formulate the intention to return to the lock. 3) When players can negotiate obstacles they were unable to get past earlier, they will experience progress and accomplishment.

MISSION VERSUS SPACE

Level design has not been studied as extensively as game mechanics. Yet, it is generally acknowledged that levels benefit from having a relatively simple gestalt, especially when this gestalt matches the intended gameplay, rhythm and pacing. To this end, a number of scholars of games and interactive stories categorized spatial structures frequently found in games (Ryan, 2001; Byrne, 2005; Adams & Rollings, 2007; Schell, 2008). These typologies include structures such as "linear plots", "braided plots", "branching trees", "networks" and "open worlds". Of these typologies, Marie-Laure Ryan specifically concerns herself with interactive story structure, while the others concern themselves with game levels. Still, the similarities between many of the structures they describe is striking. In fact, it is a common observation that in games, stories are, at least partially, structured spatially instead of temporarily (Jenkins, 2004). This causes some confusion whether these categories concern themselves with level geography or topology. As Ryan focuses on interactive storytelling, her categories are clearly topological, but the other three typologies are much more geographical in nature. This confusion suggests that in level design topological structures and geographical structures are frequently isomorphic, which is not always the case.

Separating a level's topology from its geography will help to create much clearer perspective on their respective structures and their effect on players' experience. In this respect, I consider the layouts discussed above to be geographies. A level topology would focus more on the logical structure of the player's tasks. These tasks or challenges are usually fairly straightforward tests of the player's abilities. They might take the form of puzzles, fights, traps or hidden objects than need to be collected. Ed Byrne suggests structuring these tasks in cell-diagrams outlining the game's flow and highlighting the different player tasks. These cell diagrams are simple informal structures that read almost like a storybook that help design a level's layout or rhythm. Cell diagrams focus on a game's logical and temporal structure instead of its spatial layout (Byrne, 2005). Other models that focus in player tasks can be found in analysis of Ben Cousins (Cousins, 2004) and with the "hierarchy of challenges" described by Ernest Adams and Andrew Rollings (2007) and that was directly inspired by Cousins' work.

The two different approaches, focusing on the geographic layouts of a level on the one hand and on the sequence of tasks on the other, suggest that in a level both structures exist at the same time. These structures are superimposed onto each other and as a result it is all too easy to confuse one with the other and to take their interrelation for granted. I

argue that a level consists of both a space and a mission (Dormans, 2010); it has a particular spatial layout and a series of tasks that need to be performed in that space. For many games the mapping between the mission and the space is quite direct and their structures might be quite similar, even isomorphic. However, this does not need to be the case. Games might reuse the same space for different missions, as is the case in *System Shock 2* (Looking Glass Studios 1999) where the player traverses the same areas of a spaceship multiple times. *System Shock 2* shows that the same space can accommodate multiple missions (assuming that the individual mission structures do not resemble each other too closely). Reuse of game space in this way is often economic: the developer does not have to create a new space for every mission in the game. It has gameplay benefits as well. For example, the player can use previous knowledge of the space to her advantage, adding to the player's sense of agency and the depth of the gameplay.

Mikhail Bakhtin's notion of the *chronotope* provides an interesting parallel for the combined mission and spaces in games. The chronotope refers to the artistic relation between time and space in literature. Bahktin observes that in literature time and space are "fused into one carefully thought-out, concrete whole" (1981, 84), and that their particular artistic intersection is important to the literary form. He also observes that particular chronotopes correlate to particular literary genres. For games it the artistic intersection between mission and space that has a big impact on the experience and quality of the game. What is more, it also seems that particular configurations of missions and spaces do correspond to particular game genres. The mission-space configurations typically found in action-adventure games are quite different from mission-space configurations typically found in computer role-playing games, for example.

One problem in integrating games of progression with games of emergence is that a lot of advances have been made in developing emergent game mechanics and also in highly detailed and immersive game spaces while advances in the development of game missions and stories have straggling behind. Over the past few decades the game development community has accumulated much experience with creating compelling game spaces with interesting rules. From the early limited spaces from the seventies and eighties to the vast virtual areas found in modern games, games spaces have grown into highly detailed constructions with near analogue qualities. Traversing the space of a contemporary game is no trivial task, especially for those games that involve movement in their core mechanics, as is the case with most action games. But also for strategic games this evolution has been fast. On needs to simply compare open free world of *StarCraft II* (Blizzard Entertainment 2010) to the tile based combat found in *Civilization* (Meier 1991) or indeed classic board games such as *Chess*, to appreciate the strategic depth that freely positioned units and more 'analogous' terrain features allow. Seeing these huge strides in the development of game spaces towards structures with a high granularity, it is curious to observe, as Noah Wardrip-Fruin does, that game stories and quests have not grown as much; game missions usually work with a very limited set of possible states, all of which are known in before play (Wardrip-Fruin, 2009, 59).

One cause for the mismatch in granularity between mission and space Noah Wardrip-Fruin points out, is the quest logic and dialog trees common to most games albeit on a smaller scale. The player's progression through a mission is simply tracked by setting up a few bottlenecks or gates to act as milestones in a story. Once the player or fulfilled the task associated with a milestone, the game advances to the next state. The implementation is as simple as keeping track of a few simple Boolean flags that control the visible entries for the in-game journal that records the game story Wardrip-Fruin (2009).

The common implementation of dialog trees can further serve as an example of the problems faced by designers of interactive missions and stories in games. Dialogs feature in many games, and while certain games do not even bother to make their dialog interactive, those that do often resort to using dialog trees. A dialog tree offers players a few optional lines to advance the dialog. Reaching certain nodes in a dialog tree changes the game's state. Many dialog trees are not really trees, but are more akin to directed graphs as often different paths through the tree will take players to the same node in the dialog. One problem commonly associated with these tree-like structures is that they quickly become inefficient and overly complex; the number of options that need to be created is much larger than the average player will ever see, and without proper editing tools an writer of a dialog tree might quickly loose track over all the options. Worse, dialog trees do a rather poor job of really creating the illusion of freedom or agency. With only a few options available at a time, chances are that players will feel constraint in their options (Wardrip-Fruin, 2009, 56). In all likelihood players will recognize the tree like structure, and it is not uncommon for them to traverse the entire tree in order to explore all possible gameplay consequences, which mostly is a trivial yet tedious task. In short, at the micro-level of the dialog, these tree-like structures often constitute poor gameplay (Dormans 2006). Still, at the macro-level of mission or game, they are quite common.

For Noah Wardrip-Fruin the problem ultimately lies with the shape of the underlying processes: the processes that underlie both the dialog tree and larger interactive mission implementations is rather uninteresting. He suggests a new approach to game fiction is warranted and that this approach should be fundamentally different from the quest flags and dialog trees that govern most missions in games. (Wardrip-Fruin, 2009, 76). I propose that a closer inspection of the mechanics of game missions offers plenty opportunities to arrive at a better shape for interactive missions.

MISSION MECHANICS

The mechanics that govern missions and player progress can be represented with the use of Machinations diagrams. These diagrams have been developed to represent the internal economy of games and to foreground feedback structures that exists within it (Dormans 2009). They model resources (small colored circles) that are collected on pools (open circular elements). Pools might be passive, or interactive. Interactive pools are represented with a double outline and can be activated through certain player actions. Arrows indicate how resources flow through the diagram, not unlike tokens in Petri-nets. Dotted arrows indicate how a pool's state (the number of resources on a pool) affects the strength of the flow elsewhere (called state connections), or how certain elements are activated when certain conditions are met (called activators). State connections have markers that indicate change ("+", "-", "+2"), activators have markers that indicates a condition ("<3", ">0", "==3"). Figure 1 represents a rough representation of the mission in the Forest Temple level in *The Legend of Zelda: Twilight Princess* (Nintendo 2006).[2] In this representation fights and test of skills are omitted, the focus is on the items that must be collected to finish the level. The diagram might seem a bit overwhelming at first, but it consists of the same basic mechanics lock and key mechanic, with a few variations, that are strung together. This lock and key mechanic is isolated in figure 2.

From the discussion in the section on emergence it became clear that game mechanics benefit from having feedback loops. The lock and key mechanics discussed above do not involve any feedback. In a Machinations diagram feedback needs a closed circuit that consists of at least one state connection that is not an activator; none of the mechanics above fulfill those requirements as all connections are either flow connections or

activators; they do not involve any state connections. There are two strategies to include more feedback in the game mechanics that control progression through a level: 1) designers could develop lock and key mechanics that do involve feedback, or 2) progress itself could function more like an abstract resource in the game's internal economy and can be gained (or might be lost) through mechanics that operate on a fairly large scale. The first option is explored in this section, the other option is explored in the next section.
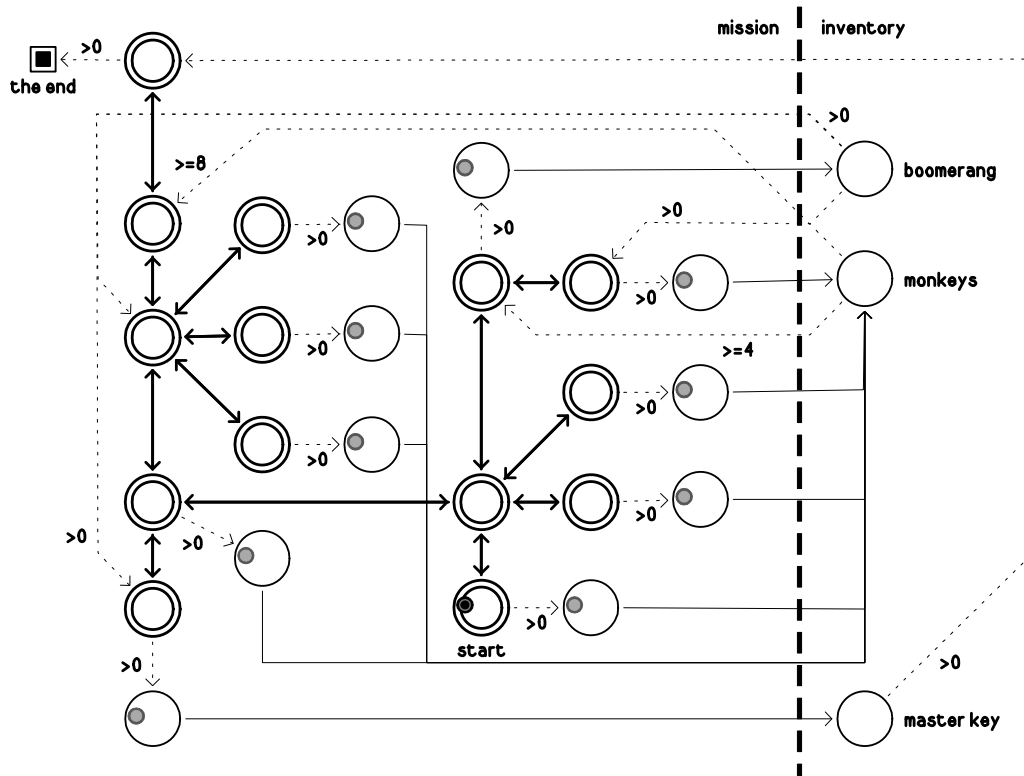
Figure 1: The mission of the Forest Temple level in a Machinations diagram. The player's location is represented with the black resource that starts at the pool marked "start". In the interactive version of this diagram, the player, represented with a black token, can be move by clicking the interactive pools adjacent to the player's current location. The player's location might activate pools containing other resources, represented with gray tokens, which are automatically transferred to the player's inventory on the right, and which in turn unlocks new locations for the player to travel to. In this way the player can progress towards the final node in the upper left. These resources represent the eight monkeys the player needs to gather, the master key, and "Gale Boomerang" which can be used to fight enemies and activate certain switches to open new passages.

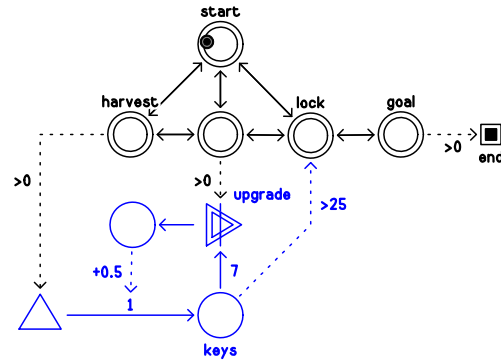**Figure 2:** The lock and key mechanic in isolation.

To create lock and key mechanisms that involve feedback, a good starting point is treating the keys more as a resource that can be produced and consumed. For example, figure 3 represents a mechanism where players need to "harvest" ten keys before they can open the lock. Negative feedback is implemented through the application of dynamic friction on the number of keys players have collected. The more keys that are collected, they quicker the keys are drained. This makes it somewhat harder to estimate how many keys need to be harvested to get past the lock. Obviously, this is more difficult to estimate as the distance between the location where keys can be harvested and the lock increases. However, the mechanic is not very interesting in itself: it boils down to harvesting enough keys and then make a run for the door, there is little strategy involved.

Figure 3: Simple feedback applied to a lock and key mechanism. In this diagram the upward triangle indicates a source: an element that produces resources, whereas the downward triangle indicates a drain: an element that consumes resources. The percentages indicate the chance a key is drained every time step.

In an attempt to create a more interesting mechanic, we can apply a common feedback pattern encountered in many games called the dynamic engine pattern (Dormans 2009). Figure 4 represents such a mechanic. This time players needs to collect more than 25 keys in order to proceed, but this time they have the option to invest 5 keys to increase the harvest rate by 0.5. However, this mechanic is probably too simple, too. It is not very difficult to find out what number of upgrades is ideal for this scenario.[3] But even that is not necessary, players can achieve the goal without needing to upgrade at all. These weaknesses should not come as an surprise: as I indicated earlier, one feedback loop is generally not enough to create an interesting dynamic mechanic. The particular strategies are the direct result of the use of the dynamic engine pattern. Games that do mostly rely on just a dynamic engine as their sole, or single-most important, feedback loop, such as *Monopoly* usually include random factors to make it more interesting and unpredictable, but that is not direction I want to explore here.

Figure 4: The dynamic engine pattern applied to a lock and key mechanism. The sideways triangle represents a converter; an element where one resource is converted into another resource. In this case 7 keys are converted into 1 upgrade.

To order create a more interesting lock and key mechanic, we can complement the dynamic engine pattern by some form of dynamic friction (see figure 5). In this case enemies spawn that will consume the harvested keys. Now players have to balance between three tasks: harvesting, upgrading and fighting the enemies to keep their numbers down. This is not a trivial task, playing the interactive version of the Machinations diagram is already a fairly interesting challenge.[4] Simply harvesting will not probably not bring the player very far, and although it is possible to achieve the goal by switching between harvesting and fighting, this requires players to maintain a delicate rhythm of switching between the two for a long time; it is very hard to accomplish. Players need to find a balance between the three actions in order to reach the goal. When the fighting is made skill based, then the most effective balance can actually vary between individual players. In its essence the mechanic is very similar to the basic gameplay mechanic of the real-time strategy: players need to balance between, harvesting raw materials, fighting and upgrading.
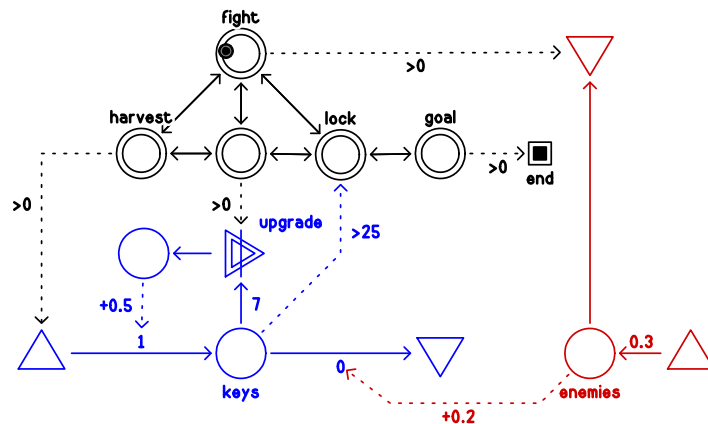

Figure 5: The dynamic engine pattern and dynamic friction applied to a lock and key mechanism.

One critical aspect of this type of lock and key mechanics is the presence of a dominant strategy: a particular course of actions that always trumps other strategies. Dominant

strategies effectively narrow the broad possibility space set up by emergence into a single confined path. When creating feedback mechanisms for locks and keys designers should be very aware of this problem. However, diagrams and simulations such as suggested by the Machinations framework can help designers preventing dominant strategies. Single feedback systems almost always suffer of dominant strategies, this is the main reason why the Machinations frameworks suggest applying multiple feedback loops to create truly emergent gameplay. Another very effective counter is to introduce unpredictable elements from outside the mechanics. This could be player skill and preferences, or even random generators. These elements will often obscure dominant patterns in the gameplay. However, randomness in general must be applied with caution, too much randomness will often lead to lack of agency and even frustration.

PROGRESS AS A RESOURCE

In many games that integrate emergent gameplay with progressive level design, the goal is to reach a certain location. This goal can mode represented abstractly with a very simple diagram (see figure 6). Essentially this is the core mechanic of *The Game of Goose* is similar. The main elaborations this game implements are the use of dice to determine how much progress the players are making each turn and the chance that a player might gain extra progress, loose turns or lose all progress. More advanced games elaborate more: the most common strategy for action-adventure type games to make the production of progress non-trivial and interesting in itself. The experiments with lock and key mechanism that involves feedback, discussed in the previous section, fall under the same strategy. This section seeks to go one step further, it explores the possibility to involve progress itself in mechanism to make the progression more dynamic.
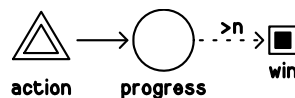
Figure 6: A simple progression mechanism

An interesting and fairly abstract implementation of a progress mechanism can be found in the latest edition of *Warhammer Fantasy Roleplay* (Fantasy Flight 2009). The rules of this tabletop roleplaying game include the concept of a "progress tracker" as a generic tool to manage the players' progress towards a single goal, competition for conflicting goals by multiple parties, or even the players' party's internal tension and friction. The progress tracker takes the form of a track that can be built from individual track pieces This allows the 'game master' to build tracks with lengths that suit the current situation. Markers on the track indicate the progress of individual characters or parties. The rules suggest a number of ways a progress tracker can be used to facilitate scenes that involve races, chases, investigations. The tracker can also be used represent a time limit by forcing the player to accomplish the players to complete a certain task before a marker on a progress track reaches the end, or to create tension by using it to track the built up of some "looming danger" unknown to the players.

Crucially, progress tracks in *Warhammer Fantasy Roleplay* do not only track progress towards some goal (or danger), they might affect gameplay as well. For example, in the scenario that is published with the rules, the progress track to represent the players' investigation into some secret cult includes a special position. Once the players' marker reaches this position, the game master should provide the players with an extra hint in order to speed up their progress. This occurrence creates a one-off, positive feedback

loop. Similar events on the party tension meter, a progress track that is part of the core rules, can cause the player characters to suffer additional stress, fatigue or wounds, causing positive but destructive feedback.

As *Warhammer Fantasy Roleplay* illustrates, progress mechanics can be used to cause feedback, and despite it lacks fine granularity, it is an excellent way to involve progress in the dynamic behavior of the game. More sophisticated forms of feedback can be used to evolve this further. A suitable pattern to accomplish this is the escalation pattern (see figure 7). With this pattern, which is found in simple, emergent games like *Pac-Man* (Namco 1981) or *Space Invaders* (Taito 1980), the player's goal is to complete a number of tasks. In *Pac-Man* this task is to eat all available dots; in *Space Invaders* it is to destroy all alien invaders. The task is getting progressively harder as the player is making progress. The dots get harder to reach in *Pac-Man* while the alien invaders start to move faster and faster as their numbers decrease. A variation on the same pattern can be found in *Tetris* (AcademySoft 1986). In this variation some form of complexity is created and it is the player to task to keep this complexity under control. However as complexity increases this task gets progressively more difficult, usually another mechanism ensures that complexity is produced at an increasing rate (see figure 8). In *Tetris* the Tetris blocks cause the complexity, and game speeds up every time the player reaches a new level.
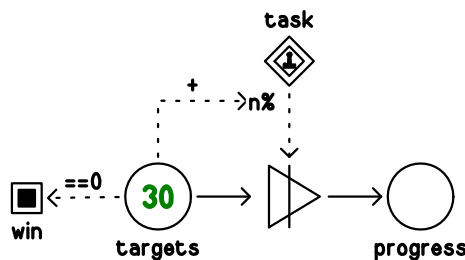

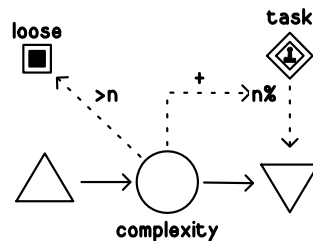**Figure 7:** Escalating complications

Figure 8: Escalating complexity

Applied to a game that includes progression, it is possible to model the progress towards a certain goal and have that progress affect the mechanics. In a sense, games of progression have always mimicked this effect by ordering a fixed sequence of challenges roughly from easiest to the most difficult. Non-linear missions with alternative branches can be built using a similar principle in order to create dynamic levels with more replay-value, but as has been argued before, this strategy is not very effective. Most of the time many more branches and challenges need to be created than an average player is every going to see. By creating a system where story-like progression emerges directly from the game mechanics, endless possibilities can be created efficiently. When the mechanics are

set up to produce enough variety, this could lead to games where interactive experiences, and perhaps stories gain a whole new dimension.

CONCLUSIONS

In games structures of emergence and progression as implemented through mechanics and levels cannot be separated. Games use level structure to teach the mechanics to players, and use special mechanics to control players' progress through a level. However, games that find a good balance between the two are rare. One would expect missions to be the place where levels and mechanics converge. But as games have evolved over the past few decades, and ways have been found to create detailed spaces and articulate mechanics, the evolution of missions has mostly stood still. As a result it is rare to find game that truly integrates emergent gameplay and progressive level design.

In this paper I have explored the relation between mechanics and levels in order to find a better balance between these two elements of game design. Leveraging the Machinations framework to investigate the underlying shape of the mechanics involved, I have explored how mechanics that control level progress could benefit from implementing feedback loops, but also how progress itself can be integrated better in the internal economy of a game. These suggestions are preliminary; apart from some promising prototypes they have not been implemented and have not been thoroughly tested. My intention in this paper was to illustrate the potential of games that integrate emergence and progression, and to inspire further research in this area.

The main point, however, is that emergence and progression need not be two separate dimensions of game design. Using the right tools designers can shape emergent mechanics to produce progressive experiences, and by having a clear perspective on a game's internal economy and mechanics designers can structure levels in such way that learning to play the game comes natural to the player. In this way notions of emergence and progression can be combined to create compelling game experiences that offer great freedom to the player at the same time.

BIBLIOGRAPHY

AcademySoft (1986) *Tetris*, AcademySoft

Adams, E., & Rollings, A. (2007). Fundamentals of Game Design. Upper Saddle River, NJ: Pearson Education, Inc.

Ashmore, C., & Nietsche, M. (2007). "The quest in a generated world". In Situated Play, Proceedings of DIGRA 2007 Conference.

Bakhtin, M.M. (1981). The Dialogic Imagination. Austin, TX: University of Texas Press.

Ball, P. (2004). Critical Mass: How One Thing Leads To Another. New York, NY: Farrar, Straus and Giroux.

Blizzard Entertainment Inc. (1998) *StarCraft*, Blizzard Entertainment Inc.

Blizzard Entertainment Inc. (2010) *StarCraft II*, Blizzard Entertainment Inc.

Burke, T. (2005). "Can a table stand on one leg? critical and ludological thoughts on star wars: Galaxies". GameStudies.org.

Byrne, E. (2005). Game Level Design. Boston.

Cook, D. (2007). "The chemistry of game design". Gamasutra. URL http://www.gamasutra.com/view/feature/1524/the_chemistry_of_game_design.php

Cousins, B. (2004). "Elementary game design". Develop.

Darrow, C. & Parker, G. S. (1933) Monopoly, Parker Brothers

DiStefano III, J. J., Stubberud, A. R., & Williams, I. J. (1967). Theory and Problems of Feedback and Control Systems. New York: McGraw-Hill.

Dormans, J. (2006). "Talking to games: Gameplay as expressive performance". In Proceedings of the International Digital Games Conference 2006 , (pp. 133–143).

Dormans, J. (2009). "Machinations: Elemental feedback structures for game Design". In Proceedings of the GAMEON-NA Conference.

Dormans, J. (2010). "Adventures in level design: Generating missions and spaces for action adventure games". In Proceedings of the Foundations of Digital Games Conference 2010, Monterey, CA.

Fantasy Flight Games (2009) *Warhammer Fantasy Roleplay,* Games Workshop, Ltd.

Friedman, T. (1999). "Civilization and its discontents: Simulation, subjectivity,and space". In G. Smith (Ed.) On a Silver Platter: CD-ROMs and the Promises of a New Technology, (pp. 132–150). New York, NY: New York University Press.

Fromm, J. (2005). "Types and forms of emergence." RL http://arxiv.org/abs/nlin.AO/0506028

Fullerton, T. (2008). Game Design Workshop: A Playcentric Approach to Creating Innovative Games. Morgan Kaufman, 2nd ed.

Hamumu Software (2010) *Robot Wants Kitty*, Max Games

Ion Storm Inc. (2000) *Deus Ex,* Eidos Interactive Ltd.

Jenkins, H. (2004). "Game design as narrative architecture". In N. Wardrip-Fruin, & P. Harrigan (Eds.) First Person: New Media as Story, Performance and Game, (pp. 118–130). Cambridge, MA: The MIT Press.

Juul, J. (2002). "The open and the closed: Games of emergence and games of Progression". In F. M̈ayr̈a (Ed.) Proceedings of Computer Games and Digital Cultures Conference, (pp. 323–329). Tampere, Finland: Tampere University Press.

Juul, J. (2005). Half-Real: Video Games between Real Rules and Fictional Worlds. Cambridge, MA: The MIT Press.

LeBlanc, M. (1999). "Formal design tools: Feedback systems and the dramatic structure of completion". Presentation at the Game Developers Conference. URL http://algorithmancy.8kindsoffun.com/cgdc99.ppt

Looking Glass Studios, Inc. (1999) *System Shock 2*, Electronic Arts, Inc.

Nintendo EAD (1992) *Super Mario Kart*, Nintendo Co., Ltd.

Nintendo EAD (2006) *The Legend of Zelda*: Twilight Princess, Nintendo of America Inc.

Maxis Software Inc. (1989) *SimCity*, Infogrames Europe SA.

Meier, S. (1991) Sid Meier's Civilization. Microprose

Namco Limited (1981) *Pac-Man*, Atari, Inc.

Nygren, N. (2007) *Knytt Stories*.

Rockstar North Ltd. (2004) *Grand Theft Auto: San Andreas*, Rockstar Games, Inc.

Rollings, A., & Adams, E. (2003). On Game Design. Berkeley, California: New Riders.

Ryan, M.-L. (2001). Narrative as Virtual Reality: Immersion and Interactivity in Literature and Electronic Media. The John Hopkins University Press.

Salen, K., & Zimmerman, E. (2004). Rules of Play: Game Design Fundamentals. Cambridge, MA: The MIT Press.

Schell, J. (2008). The Art of Game Design: a book of lenses. Morgan Kaufman.

Simpson, Z. (2000). "The in-game economics of ultima online". Presentation delivered at the Game Developers Conference, San Francisco, California, March 2000. URL http://www.mine-control.com/zack/uoecon/uoecon.html

Smith, H. (2001). "The future of game design: Moving beyond Deus Ex and other dated paradigms". Retrieved September 8, 2008. URL http://www.igda.org/articles/hsmith_future.php
Taito Corporation (1980) *Space Invaders*, Atari, Inc.
Wardrip-Fruin, N. (2009). Expressive Processing. The MIT Press.

ENDNOTES

[1] The history of the notion of internal economy within games is somewhat fragmentary. It appears in a chapter title in (Rollings & Adams, 2003) which is a precursor to (Adams & Rollings, 2007), but the notion itself is not really discussed in that book. There are a few examples of the use of the term or the synonym "in-game economy" in the context of games, such as in (Simpson, 2000) and (Burke, 2005). In these two cases the term is reserved for trade and inventory systems (as opposed to, for example, combat systems). It is not until the notion was discussed in more detail in (Adams & Rollings, 2007) that it starts encompass types of resources (such as health) than more than a strict interpretation of "economic" would allow and that a game's internal economy can actually include combat systems, leveling systems, etc., as well. Since then, the term appears in lectures and syllabuses that follow Adams and Rollings book. It is in this wider sense that the term is used here.

[2] For an interactive version of this diagram where the mission can be "played" see http://www.jorisdormans.nl/machinations/wiki/index.php?title=Mission_Mechanics.

[3] As it turns out, the ideal number is actually one *or* two upgrades, both arrive at 26 keys at exactly the same time, while taking no upgrades or more upgrades turns out to be slower.

[4] Also available at http://www.jorisdormans.nl/machinations/wiki/index.php?title=Mission_Mechanics.