Pattern Recognition: Gameplay as negotiating procedural form

Tom Betts
School of Art, Design & Architecture
University of Huddersfield
Queensgate, Huddersfield, UK, HD1 3DH
0148404220288
tom@nullpointer.co.uk

ABSTRACT
This paper will examine the relationship of pattern recognition and Gestalt principles to procedural form in gameplay. It will identify key features of pattern based play mechanics and outline important synergies between programming paradigms and procedural form. In the course of the paper I will examine the formal and aesthetic qualities of procedural structures and discuss how they generate the experience of psychological flow. I will also identify the role of these mechanisms and their effects in current game design.

Keywords
Video Games, Computer Games, Emergence, Ludic, Procedural form, Generative, Digital Sublime, Programming

INTRODUCTION
Video games represent the intersection of many different forms of interaction. In recent years the scope of game design has broadened to incorporate aspects of narrative driven spectacle derived from cinematic media. This approach has entailed the adoption of various film production methodologies and authorial control. However, the player interface for almost all game design still relies on pattern based ludic exchanges inherited from a long history of non-digital gaming. Unlike linear entertainment media, the production of meaning in a game experience is driven by interaction, and framed by evolving patterns of production and negotiation. This paper examines how in an increasing content rich environment, procedural methods and pattern processes are still central to game design, and can drive both complex gameplay and generate an inherent narrative through the players negotiation of emergent procedural forms.

Firstly I will examine the stages of pattern production, recognition and negotiation, as they exist in games. I will discuss how Gestalt psychology frames the processes of perception and recognition, and how procedural mechanisms generate the formal structures that drive pattern based gameplay. I will then identify specific examples of Gestalt features in a range of games and genres. Following this, a section on the constraints of procedurality will explore the ideas of emergence in pattern production. Finally I will examine notions of immersion and authorship in the production and negotiation of procedural form in games.

Proceedings of DiGRA 2011 Conference: Think Design Play.

PATTERN RECOGNITION

Many traditional board game interactions rely heavily on pattern recognition. The hand driven processes of *Chess*, *Go* or *Draughts* are generally mastered in two cognitive stages. Firstly the player exercises their natural pattern decrypting skills in order to understand the visual relationships of objects in the gamespace. Pattern recognition is generally described in cognitive science as a two stage process, a combination of *feature detection* and *Context* or *Connectionist Modelling*. Feature detection is identified as the process of identifying and grouping small fragments of pattern from visual data and then organising those elements into larger forms based on their similarity to internal mental templates, Selfridge(1959). This system is augmented by an additional contextual layer, where confusing similarities in basic groupings can be solved by using the surrounding visual information (Figure 1). In gameplay this process is extended through interaction, with the user able to test assumptions and explore mappingS though trial and error. Innate Gestalt perceptions are blended with game specific rules of organisation and interaction.

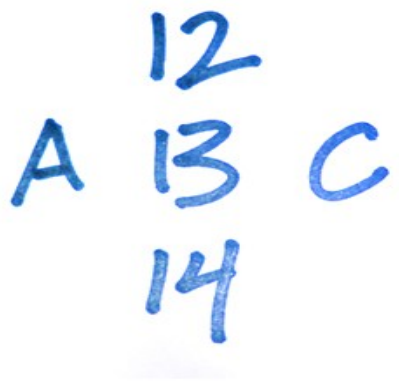

Figure 1: The importance of context in pattern recognition

This process then feeds into a second stage of engagement where players use their developing pattern knowledge to mentally project the potential of subsequent moves. In this way players gradually assimilate the rules of a specific game and generate the best mental model they can of the system and its permutations. They then use this model to anticipate subsequent developments of the system and adjust their interactions accordingly. Generally the goal of mastering this process is the pursuit of a pre-defined win condition. But even within sandbox games (games with no objective goals), there is an innate tendency for players to strive for results that generate harmony, symmetry, simplicity and closure of form. These tendencies are all core elements of Gestalt psychology.

GESTALT PSYCHOLOGY

The Gestalt field of psychology, founded by Max Wertheimer, Kurt Koffka and Wolfgang Köhler in the early 20th century proposed an innate, holistic idea of visual perception called Gestalt Form. In this framework all visual phenomenon are seen as part of a 'whole'. The elements of this whole only gain meaning when all constituent parts are interpreted in relationship to each other and their totality. Gestalt psychology asserts that this holistic perception occurs at a subconscious level and is a primary function of our relationship with the world.

"The organization of the visual field occurs within us essentially without our involvement, and without our explicit awareness of any of its laws." (Metzger, 2006).

Is is interesting to note how Gestalt principles are often extended to imply a transcendental relationship between the viewer and the world being perceived. Wolfgang Metzger's text on visual perception *Laws of Seeing* (2006) is primarily a scientific analysis of human visual processing. However, later in the publication he begins to conflate his scientific findings with a philosophical stance on phenomenology. Metzger uses the universality of Gestalt rules to question whether pattern processing is more than just a psychological effect of an individuals perception of the world. He suggests that the 'mental laws' of cognition and the 'physical laws' of organisation are in effect the same exact processes, simply being executed in different contexts.

"The mental and physical laws come into conflict only if we cling to the opinion that the human spirit is a stranger in this all-too-petty physical world," (Metzger, 2006).

The holistic aspects of this cross boundary proposition have been greatly influential in psychology and philosophy. Metzger is proposing that certain pattern principles underlie the production and recognition of all forms, whether physical or psychological. Although the philosophical ramifications of this a statement are beyond the scope of this paper, it is interesting to note how Gestalt principles can be seen as inherent in both production and perception.

The automatic processes of Gestalt perception are employed seamlessly in much of game design. Equally, the exchange of pattern recognition and pattern production is a vital element of gameplay. The holistic extent of this player-system network can significantly effect the states of immersion and authorship in a game. I will discuss these issues in a later section.

If we can consider that Gestalt principles are present in the production of form as well as its perception then it is my argument that procedural programming methods are an ideal mechanism for generating and investigating these principles in digital games.

PROCEDURALITY

The terms 'procedurality' and 'procedural form' have various contemporary definitions. Many of these focus on the notion of procedure as a linear process of operations with no room for flexibility 'Just following procedure'. In computer science, the origins of procedural programming lie in the gradual abstraction of linear code into subroutines. In simple computational functions procedural code operates linearly. However, the fragmentation of code, combined with the increasing power of CPUs to perform rapid iterations, soon allowed procedural programming to explore complex nonlinear phenomenon and emergent behaviour. A typical example of this process is the cellular automata game *Life* developed by John Horton Conway in 1970. (Figure 2)

Life is played out across an infinite grid of binary cells that are either alive or dead (full or empty). These cells change state based on a set of simple rules that operates on a local scale yet produces a global transformation. When initally performed by hand Conway could see complex patterns and detailed behavior emerge. But it was only when computers were able to iterate the game at greater scales and rates that the true range of complexity and emergence could be explored. Conways *Life* is a classic example of a ruleset that generates procedural form. It essentially deterministic but produces a such a wide range of permutations that it seems 'alive'. Almost all games are driven by similar procedural mechanisms that allow iterative expression but are governed by the limitations of specific rulesets.

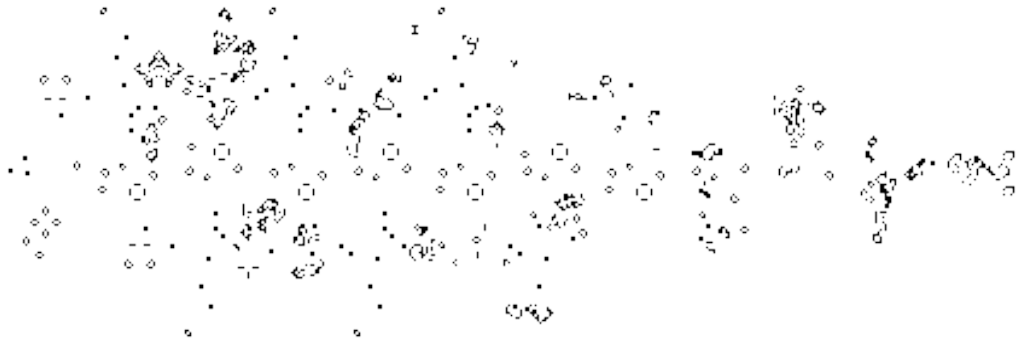

Figure 2: *Life*, showing a 'puffer train' formation

The game of *Noughts and Crosses* is procedural, yet compared to *Life* its permutations are extremely limited. Its simple ruleset can easily be executed with pen and paper, and before long a player can project and control the result of any game. It is only when the calculations involved in a game hit a specific magnitude of speed or scale that the results reach a point beyond human determination. Manuel DeLanda says *"This makes the expression "the behavior of equations" less metaphorical because recursion transforms a static mathematical object into a dynamic computational process"* (2011). Its this dynamic behavior that is key to the generation of interesting procedural form in games.

Ian Bogost also distinguishes the notion of procedurality from static systems by highlighting the use of programming as a autonomous creative proxy *"...one authors code that enforces rules to generate some kind of representation rather than the authoring the representation itself."* (2007). He also acknowledges this generative aspect as a core element of computer programming *"Procedurality is the principal value of the computer, which creates meaning through the interaction of algorithmns"* (2007). In games, procedural systems are 'played out' in real time (in player time). They present evolving dynamic forms and symbolic interactions through a rich audiovisual interface. Players instinctively engage with these abstractions, through which they can explore and manipulate the underlying procedural grammar. The interface through which these exchanges operate is generally one of pattern production and recognition.

Other tiers of context might be built on this layer, adding higher level concepts of political or social motivation, in some cases completely overriding the initial Gestalt interaction. Bogost describes this as the development of a procedural rhetoric, encompassing broader aspects of psychological and cultural exchange. These more linguistic channels of communication generally allow greater authorial control and potential for cinematic and literary immersion, however they frequently limit the expressive range of procedural form and significantly alter relationship between player and designer, I will return to this issue later.

How do procedural forms and Gestalt principles function in digital games? The following section outlines some key principles of perceptual psychology in relationship with familiar game designs.

GESTALT PRINCIPLES IN GAMES

Many Gestalt principles relate to process of grouping visual forms and events. This mental organisation occurs in relationship to key attributes in the scene being perceived. Attributes such as; *proximity*, *similarity*, *closure*, *continuity*, *common fate* and *symmetry* are all organisational cues defined by Gestalt principles. These attributes are key to the audiovisual design of any interactive game system. But, in addition to driving visual

pattern recognition Gestalt principles are also vital in aiding a players participation in the ongoing procedural logic of gameplay. Pattern processing encourages people to mentally decipher systems, but it also prompts them to respond, proactively organising components in the game space. Anyone who has found themselves stacking pennies in neat columns or separating lego bricks by colour, knows that ordering and pattern matching is a natural and subconscious urge. Gestalt processes are therefore not only part of passive 'reading' but also drivers of pattern 'production'.

Figure & Ground

A central concept of Gestalt perception is the notion of the *figure-ground* relationship. This asserts that our visual processing will always try to differentiate between objects and their backgrounds. We can see this typified in the classic face/vase illusion and the Necker Cube (Figure 3), where the graphical readings compete so closely for recognition that the image seems to oscillate. This oscillating equality of figure and ground is also referred to as multistability.

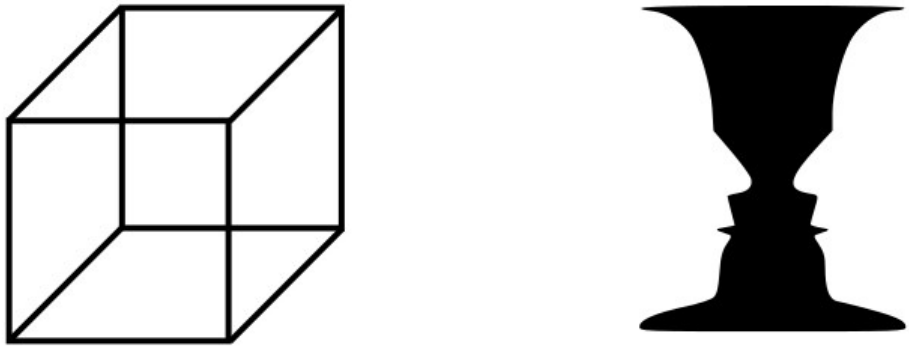

Figure 3: Figure-Ground Multistability

Most videogames deliberately avoid confusions of figure and ground since the clarity of the visual field helps the player to identify the correct moves to make. However certain games specifically exploit this dichotomy. *Ikaruga* (Treasure, 2001) for example is a shmup (Shoot Em Up) designed around a polarity system where the player can flip between two distinct yet integrated visual layers, one black , the other white. In this scenario the gamer is forced to play within a face/vase world The player has to learn to negotiate a complex set of procedural bullet patterns while also tracking which of the games two colours is the figure and which is the ground. Other games which echo this style of oscillating figure/ground relationship are titles such as *Shift* (Armor Games, 2008) (Figure 4) and *Echochrome* (Game Yarouze, 2008).

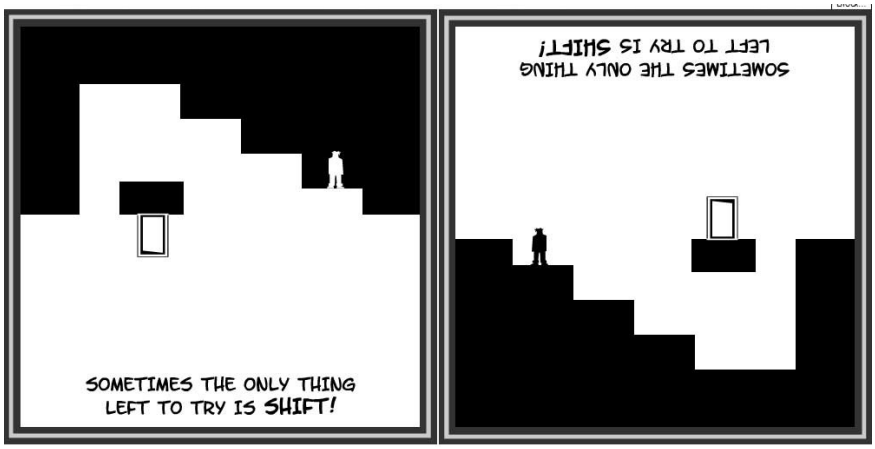

Figure 4: *Shift*, demonstrating the black/white dual plane system

The perception and separation of figure is often reliant on the visual grouping of objects and how shared elements of design differentiate such groupings. There are a number of key Gestalt principles that help define these groupings.

Proximity

The principle of proximity describes how objects close to each other are seen as single units. This a key element of games like *Every extend extra* (Q Entertainment, 2006) where the player must calculate the distance between objects in order to anticipate the distance to which his detonation chain might grow.

Similarity

The principle of similarity identifies that individual forms which share visual characteristics are seen as a group, even if separated in space. This is one of the most prominent Gestalt principles and is used in countless mechanisms such as; match three puzzle designs *Bejewelled* (PopCap, 2007), bullet colouring in shmups like *EspGaluda*, (Cave, 2003), unit designs in RTS (Real Time Strategy) games like *Pikmin* (Nintendo, 2001) and many more.

Continuity

The principle of good continuity reflects the desire to follow smooth, continuous patterns and progressions rather than abrupt changes or dislocations. This principle is key to defining the ideal racing line in most games. It is also part of the aiming procedure for any gravity modelled projectile, either in 2D as in *Worms,* (Team 17, 1995) or 3D with the FPS (First Person Shooter) practice of 'leading' your aim to account for travel time and arcs. We can also see the desire to rectify broken continuity in the 'repair' of broken lines in puzzle games like *Bejewelled*.

Symmetry

This principle recognizes that we see forms in symmetrical configurations as single groups, with a focus at their axis of reflection. This is another aspect frequently found in puzzle games, where it is markedly easier to see a symmetrical solution than an asymmetric one . Shmups commonly use mirrored patterns which enable players to gauge the movement of whole systems simultaneously and react accordingly. Designers frequently use symmetry in environments to guide player movement, or create balanced opportunities (such as in competitive multiplayer scenarios). Even in sandbox games where goals are player generated, users tend to build symmetrical forms where they can.

Closure

The principle of closure demonstrates that we see forms as complete objects even if there are voids in their layout. This phenomenon is again a core element of puzzle games, but is also prevalent in RTS or Tower Defence games where the player mentally connects the effective ranges of towers or units to form barricades or perimeters. The notion of surrounding and capturing a territory is key to the strategy of *Go, Risk* and many of their digital equivalents. In a more visually direct manner 'capturing' by closure is the principle gameplay mechanics of *Qix* (Taito, 1981) and *Loop* (GameLab, 2001) .

Common Fate

The principle of common fate indicates that individual objects moving together are seen as related. This is a feature frequently used in shmups where unique enemies following similar trajectories are seen as a a single ‘wave’, grouped together in contrast to other forms on screen. This is also visible in RTS games and racing titles where individual entities traveling in the same direction and velocity are perceived as a grouped threat. For example players can easily distinguish enemy takedown cars in *Burnout Paradise* (Criterion, 2008) as a separate visual layer due to their similar speed and motion.

Some games intentionally subvert or disrupt the players perceptual readings. One common technique to achieve this is the of inversion or re-orientation a key axis of interaction. In *VVVV* (Terry Cavanagh, 2010) this is done by making the players jump button reverse gravity for their character. The simple perversion of a familiar interaction in game vocabulary causes the normal interpretation of the gamespace to collapse. The player must adjust their mental reading of the environment accordingly. Similarly the *Portal* (Valve, 2007/2011) series uses a science fiction gun to exchange the cartesian axis of gameplay, forcing the player to learn a new patterns of recognition and interaction (Figure 5). Even when game designs disrupt our natural pattern processing, most players adapt quickly to the new rules of the world. It is this process of mastering new patterns that drives skill development and player progression in many games.

Many of these key Gestalt principles should be automatically incorporated into level design and GUI (Graphical User Interface). Aspects of harmony, symmetry, closure, proximity and continuity are all important elements in the design of game spaces. Used correctly they can guide and support player interaction, greatly improving the immersive experience. The usage of techniques like these is a vital part of the production process for any game and is reflective of good general design principles. However, it can manifest itself as a ‘top down’ integration, where Gestalt psychology is used to support and augment a predefined interaction framework, rather than being part of the emergent game experience. There is no doubt that the negotiation of audiovisual space via Gestalt processes significantly increases player immersion, but I would argue that contemporary mainstream production models often undervalue the potential of pattern processes as the driver for actual gameplay itself.

Figure 5: *Portal*, demonstrating axis switching via portals

As suggested earlier in this paper, I propose that Gestalt principles are not only embedded in our recognition system but are also a key element in the computational and conceptual construction of form. Computationally derived structures can be closely associated with the same Gestalt principles as both procedural programming and psychological perception deal with the emergence of complex structure from simple formal rules. Many games are based entirely on the relationship between these emergent procedural forms and the players reaction to them. Although this may be easier to see in abstract game genres it is an underlying mechanism in almost all types of gameplay.

Gestalt patterns in game genres

Game design often exploits our subconscious desire to negotiate and organise patterns by presenting the player with visual puzzles to solve. These puzzles are frequently based on a disturbance of harmony or disruption of Gestalt principles which the player is tasked with repairing. Incomplete forms - *Tetris* (Alexey Pajitnov, 1984), disordered matrices - *Bejewelled* and disorganised clutter – *Peggle* (PopCap, 2007) are all visual elements that prompt gamers into 'tidying' mode. In these formats, the player manipulates architectural elements of the game-world (from an abstracted gods-eye perspective), in order to achieve specific formal resolutions. Another approach is to challenge the user to navigate procedural form more literally, from the embodied position of a game avatar. In these cases the puzzles can become more representative of physical spaces; chaotic mazes - *Mr Driller* (Namco, 1999), skateable objects - *Tony Hawk's Series* (Various Developers, 1999-2010) and bullet patterns – *Ikaruga*. In either scenario procedural forms are allowed to evolve through iteration and interaction in order to produce complex emerging challenges with variable solutions.

Simulation games are often designed around the management of discrete entities and proximity based interactions - *Sim City* (Maxis, 1984-2011), *Theme Park* (Bulldog, 1994). Even though the graphical representation of theses elements may be part of a photo-realistic fiction, their actual gameplay functions are highly symbolic. In many simulation games it is even possible to view colour coded maps that show the underlying interactions of these elements in a more iconic fashion. RTS games use similar graphical techniques via HUD (Heads Up Display) systems and strategic overlays to display unit interactions in a more direct and symbolic manner. In some cases, strategic gameplay actions can be entirely performed on a symbolic/abstracted layer, while the cinematics and storyline play out in an entirely different part of the graphics engine - *Supreme Commander* (Gas Powered, 2007).

Racing games also require a high degree of pattern negotiation, forcing players to make quick navigational decisions based on rapidly evolving visual information. This is

particularly true in the case of open world games such as *Burnout Paradise* , *Grand Theft Auto* (Rockstar, 2001-Present) where the removal of laps reduces the players need to learn routes by memory. Instead players have to recognise and negotiate the procedural patterns of traffic, road intersections and incidental obstacles. In order to distinguish important gameplay objects from the general background ambiance, key features are often telegraphed with virtual HUD icons and grouped with the simple gestalt principles of common fate, continuity, and similarity. The notion of the perfect 'racing line' is essentially an trained form of pattern recognition, where the driver 'reads' the terrain ahead and calculates the best route available. *Mirrors Edge* (EA Digital Illusions, 2008) is a interesting hybrid of platformer and racing game that clearly demonstrates a Gestalt approach to visual design. Key architectural elements are colour coded in real time to indicate the navigational options available to the player (Figure 6). Shortly after the games commercial release a series of entirely abstract time trial maps were made available. These tracks made no reference to the original games storyline and neither were they based on realistic architectural design. Instead, the extra tracks were modeled in an abstract visual aesthetic, providing a purer pattern negotiating challenge.

Figure 6: *Mirrors Edge*, demonstrating colour coded navigational cues

Shoot-Em-Ups or Shmups are another genre of gaming where pattern generation/recognition is very pronounced. Although a niche format, derived from space invaders, Shmups have developed over 30 years with an increasing focus on the generation and navigation of complex patterns. In games such as *Mushihimesama*, (Cave, 2007) *Ikaruga* or *Geometry Wars* (Bizarre Creations, 2003) (Figure 7), mathematical functions are used to generate interlacing fields of enemy fire and choreographed waves of enemies. Players must negotiate aspects of negative space, pattern formation and colour coding in order to dodge the evolving maze of particles. In many Shmups, bullets are coloured to help the player read the visual field, for example; pink for geometric spreads, blue for aimed shots. Often, the garish look of Shmups can be attributed to this design decision, distinguishing patterns through coloration and shape. In these examples, the games visuals are developed to service gameplay, as opposed to cinematic games where graphics are generally used to drive character or atmosphere. Just as players of *Go* can see tactical opportunities in the back and white matrix of a game board, so Shmups reward and develop a players strategical reaction to the visual expression of their internal algorithms. Expert players can 'herd' clouds of bullets around in complex formations and predict minute gaps in forthcoming patterns. For many gamers the mental acrobatics and manual dexterity to perform these tasks is too much. Even so, the zen-like state achieved when playing well has ensured that shmups still have a place in contemporary gaming culture.

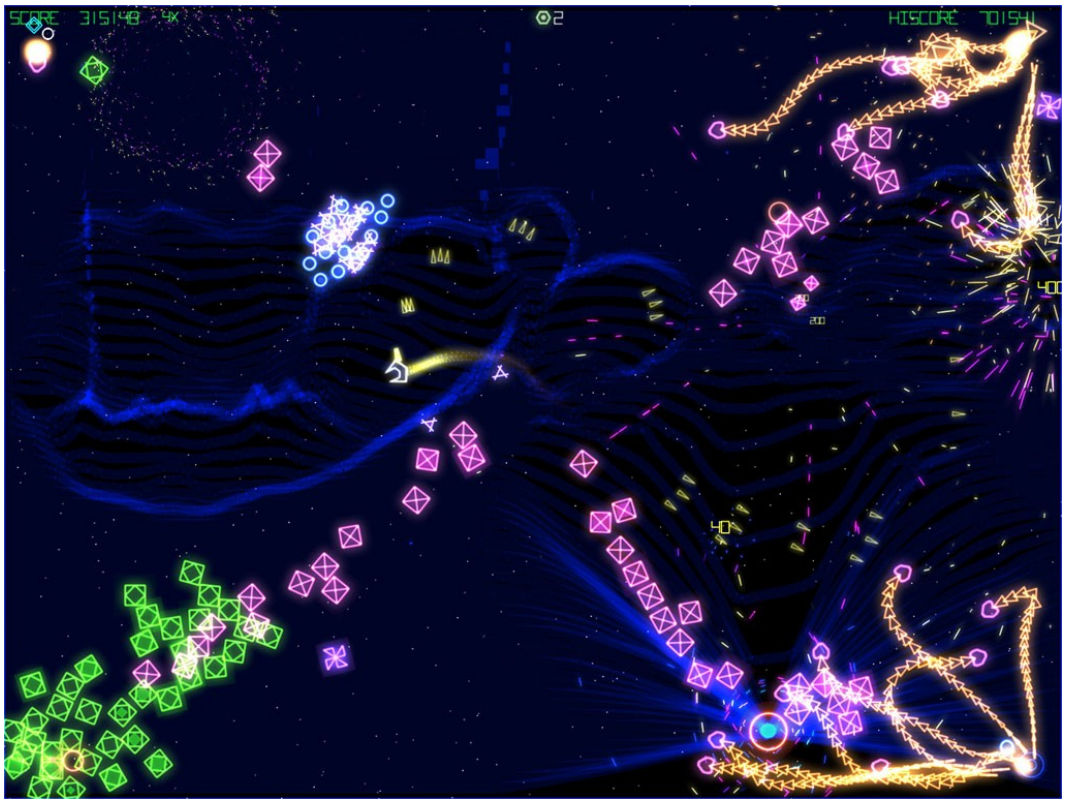

Figure 7: *Geometry Wars*, demonstrating various Gestalt cues

It is clear then that unlike static gestalt designs, interactive games allow the player to influence the visual field, controlling the formal elements that constitute the games symbolic vocabulary. A well designed game will manage this vocabulary in such a way that our patten processing instinct is kept stimulated with emerging problems and unpredictable opposition. When using evolving procedural systems designers must decide where to place restrictions on expressive range of their algorithms, to promote or control different degrees of emergence.

DEGREES OF EMERGENCE AND AUTONOMY

When procedurality is foregrounded in game design it produces a dynamic and interactive field of evolving audiovisual content. This field represents the designers guiding principles (through game mechanics), but also expresses some of its own autonomy (as emergence through code). This autonomy can be agent or AI (Artificial Intelligence) based, or it can emerge from environmental factors. The characters in *The Sims* (Maxis, 2000-Present) are partly autonomous agents, going about their own business with often unusual results. The player's pet in *Black and White* (Lionhead, 2001) is even more unreliable, learning by observation and simple AI training. The behaviour of autonomous agents or live procedural systems (such as realtime plant growth, weather simulation) can make game worlds unique and unpredictable. In many cases players can also reshape the gamespace themselves by destroying, re-ordering or constructing forms. Inevitably, the way gamers read patterns in existing game spaces influences the way they create their own patterns in response. While some designers restrict this behaviour in order to maintain tighter control, others rely on such emergence as part of their gameplay aesthetic. The degree of freedom given to both players and game mechanics is invariably linked to the type of game being designed and the key features of its genre.

Simple rules like those of *Bejewelled* or *Puzzle Bobble* (Taito, 1994) are designed to generate slight but potentially infinite variations of play. Their rulesets are limited by the

range of possible interactions and configurations, but can generate a large number of different gameplay scenarios. This design framework is close to non-digital single player games such as *Solitaire* or *Mah Jong*. Although the number of permutations in these games may be beyond practical determination, they are still thematically restricted. There is little chance for surprise events to occur or new gameplay structures to emerge. Pattern processing in these conditions quickly settles into a muscle memory response, only perturbed by changes in the games tempo or the occasional introduction of a new piece of functionality.

Whereas basic puzzle mechanics can be played out on boards or cards, more complex procedural techniques can only be processed by computer, due to considerations of magnitude, complexity and timescale. The geometric abstraction and mazelike detail of *Ikargua's* bullet patterns or the variable infinite worlds of *Minecraft* (Mojang, 2009-Present) are only possible due to the power of computers to iterate over procedural code. We can observe the complexity levels of structural form increase from simple puzzle games all the way to Will Wrights *Spore* (Maxis, 2008). This increase of detail is a direct result of the extended usage of procedural code and techniques.

As procedural complexity is allowed to increase, there is the potential for more emergent phenomenon and expressive range. If a game world displays the capacity for autonomous activity, players are often more inclined to believe in its fiction. Equally, allowing players a greater degree of influence on the world through procedural systems can give them an improved sense of control and agency. Players negotiate the patterns produced by these more complex systems (*Minecraft, Spore*) in the same way they would for simpler games. However, the forms they engage with are more dynamic and are likely to produce scenarios that require more active re-interpretation than learned muscle memory responses. Open procedural systems encourage this aspect of player autonomy and experimentation (as opposed to reinforcing muscle memory performance or narrative control). This is typified in the differences between *Wipeout* (Psygnosis, 1994) and *Burnout Paradise* both ostensibly mainstream racing games. The former concentrates on replaying tracks selected from a traditional menu system in order to achieve the perfect racing line and record times, whereas Burnout Paradise rarely repeats any specific route and rewards players for their exploration of its world as much as for winning races. It is clear that the developers at Criterion take advantage of the range of gameplay scenarios offered by the random interactions of the autonomous world and the players own gameplay goals. This form of ‘open world’ design provides more opportunities for a player to engage in emergent forms of gameplay, simply because it allows the development of procedural form to occur with some autonomy (traffic incidents, random cars to chase, player defined goals). The city itself has been designed with Gestalt principles in mind, offering patterns of ramps to climb, causeways to negotiate and concrete chasms to jump. The city in *Burnout Paradise* is a predefined world, planned in much the same way that Tony Hawks digital skateparks were designed. There are other developers using procedural processes to both produce and maintain their game worlds in real time.

Minecraft is an open ended game of exploration and construction, based on a mathematically generated world. Both the world itself and the play mechanics are heavily reliant on the frameworks of procedural form. The terrain in the game is generated from various noise algorithms and is effectively infinite. There is no hand placed content in the entire gamespace, every element of the landscape is calculated from deterministic rules, to the extent that a world generated from the same numerical seed will be identical for every single player. The players negotiation of the worlds produced is essentially the exploration of procedural algorithms drawn in three dimensions and filtered by simple game rules. It is the careful superimposition of these rules on the underlying form that give purpose and direction to the players interactions The rarity of different block types, the altitudes at which they occur and the conditions necessary for creatures to spawn all

give a wider expressive range to the underlying mathematical world. Players begin by simply negotiating the terrain, becoming familiar with the structures that are possible and the mechanisms for navigating them. Next they become familiar with the distribution of elements and their uses. Finally they understand the construction of the world in such a way that allows them to engage with it on an almost subconscious level.

Minecrafts grid based terrain follows simple procedural rules in the same way that Conways game *Life* does. *Minecrafts* rules contain more localised conditions and its gamespace is realised in a chunky 3D visualisation. Within this pixellated world, the permutations of underlying cellular automata are given virtual-physical form; array values become blocks of sand or dirt, noise functions become mountains and caves. The procedural form becomes less algorithmic and more tangible through the embodiment of the player in this space as an FPS avatar. Players develop a familiarity with the patterns of the world and proceed to re-order its material in a way that is both game directed (the progression of metals to tools etc) and player driven (personal construction ambitions). The open nature of *Minecrafts* procedural processes and the transparency of its visualisation are vital in generating scenarios for emergent narrative and sandbox style gameplay. Systems of open procedural form can often dissuade gamers who are used to more precise guidance or linear exposition. However for many players the potential for endless replay and exploration is the most enticing factor of the game. *Minecraft* is also unusual in its reluctance to offer any real tutorial or guidance to new players, instead they have to engage with the world immediately and essentially negotiate its procedural grammar through direct interaction.

As with the cognitive stages of pattern recognition the process of negotiating procedural form within gameplay can generally be seen as operating in two stages. Firstly there is an initial phase of pattern recognition, where players learn the mechanics of a specific ruleset from interaction with an audiovisual representation. Secondly, established patterns will become embedded in a form of muscle memory, where they operate almost subconsciously. Both modes of interaction are closely aligned with the processes of Gestalt perception and are frequently associated with the experience of what psychologists call *Flow.*

PATTERN PROCESSING AND FLOW

Psychologist Mihaly Csikszentmihalyi coined the phrase 'Flow' (*Flow: The Psychology of Optimal Experience,* 1990) to describe the sensation we experience when we are fully immersed in any activity where challenge and performance are perfectly balanced. If a participant finds the task at hand too difficult, he will drift from this state into sensations of anxiety or confusion. Conversely if the task is too easy the participant will enter a state of boredom or apathy. Csikszentmihalyis frequent references to sport and problem solving make his concepts particularly applicable to the world of game design. This connection was later explored by game designer Jenova Chen in order to discover how games might offer adaptive levels of challenge (*Flow in Games*, 2006).

The sensation of Flow is often marked by an almost subconscious process of pattern recognition and production. Metaphors of navigating structures and understanding underlying forms are common in descriptions of the experience. There are clear parallels with Gestalt perception and procedural rules. The Flow state, as experienced in games, can come from many sources (mastery, immersion, feedback), but it is frequently driven by the interplay of procedural form and the players interaction. Being in 'the zone' while playing a game may result from simple operations such as clearing lines in *Tetris* or ordering cards into suits in *Patience*. But it may also be a result of more complex rulesets and their permutations. In promoting Flow in games it is important to ensure that the interface between player and system is as fluid as possible. Understanding Gestalt principles and procedural pattern generation is vital to the development of this interface, since both processes occur in an emergent and often subconcous fashion. Many gamers

may see the experience of Flow as a deeply immersive, transcendental experience, yet on the other hand sustained Flow could be described as a state of fugue or even stupefaction. It is important to understand that not all examples of a Flow state are equal in terms of their production or results. Game designers can reinforce or undermine various lines of passage into immersive Gestalt states, depending on the intended end state. Immersion in a narrative driven game represents a different mode of engagement than immersion in a more abstract game system might.

MODES OF IMMERSION

I propose that in relation to the procedural rhetoric of narrative driven games and the pattern negotiation of procedural form, we can distinguish two modes of play; narrative immersion and pattern immersion. These modes are not mutually exclusive but do rely on different forms of engagement with game systems and consequently require differently production methodologies.

Narrative immersion is typified as 'escapist', it represents a form of self projection into generally pre-defined characters and fantasy worlds. Players 'escape' into theatrical settings where they act out roles within a pre-authored script. This form of projection is driven by exposition and verbal/textual interaction. It operates on a linguistic level of consciousness, reliant on an deep understanding of cultural and social tropes. There may be flexibility in the way the story plays out but the experience is often geared towards providing a well crafted narrative 'ride' as is demonstrated by cinematic or literary media.

In contrast, pattern immersion operates on sub-linguistic Gestalt level, relating to a participants evolving response to procedural form. In this exchange the underlying frameworks of game mechanics are more prominent. Gameplay scenarios are driven by emergent mathematical phenomenon and produce experiences in a 'bottom up' manner rather than the 'top down' design of narrative approaches. Immersion in games driven by these procedural mechanics tend to lead to a suspension of personal identity and character. There may still be underlying arcs of challenge, failure and success, but these elements occur in the context of a user developing non-linguistic skills and progressive mastery of the game system.

In essence both modes involve the transformation of identity in a consensual exchange with the player. In the narrative mode, the self is overlaid onto a pre-existing game framework of story and character (the gamers identity is augmented or projected). In the second mode, the sense of identity is dissolved or sublimated by immersion in a system of formal non-linguistic rules. Both are escapist tendencies, but operate towards different ends of the ego spectrum.

I would argue that procedural pattern immersion can only be achieved through active gameplay experience. It is an iterative process, formed through repeated cybernetic exchanges, to be understood it has to be enacted. On the other hand, story driven games rely more on a players prior experience of literary, cinematic and cultural tropes. This focus on the cross-media nature of narrative also enables them to be described more easily 'outside of the game' through reference to other formats. Perhaps it is this ease of communication that leads to certain game genres appearing more prominently in media coverage and critical discussion. Describing *Call of Duty 4: Modern Warfare* (Infinity Ward, 2007), through its narrative setting, communicates much about the game experience, even to a non gaming audience. Story has a relevance outside of active gameplay and can be experienced in a way that pattern negotiation can not. This is not surprising as narrative games follow in the tradition of re-combining, or re-enacting story experiences, where procedural games are concerned with the realtime variation of scenarios. A narrative description of *Minecraft* is much less informative of the actual game experience. The two forms of immersion outlined above also relate closely to how game designers decide to expose mechanics and allocate authorship in their games.

AUTHORSHIP - MAKING/PLAYING/EXECUTING

Narrative games tend to maintain a traditional author-reader relationship. Although the player may be allowed some flexibility in their exploration of the story, the main path of exposition is usually tightly controlled. In games based on negotiating procedural form it is generally the game mechanics, rather than the storyline, that constitute the main channel of communication. This creates a different relationship between the user and the game (and between the player and designer). The exchange is less about delivering a clearly authored experience and more focused on the exploration of particular game mechanics. Storyline or character elements are occasionally imposed on games of procedural form in order to motivate players or present some fictional excuse for the task at hand. However, unlike narrative in driven games, they are rarely the main point of player interaction or gameplay progress (Figure 8).

It could be argued that traditional methods of narrative delivery are by their nature incompatible with games based on procedural forms, but there is increasing evidence of games actively exploring the idea of emergent narrative. An example of this can be found in *The Sims,* where the AI driven character relationships and desires can combine to generate ongoing story lines and narrative drama. In a more experimental framework *Façade* (Procedural Arts, 2005) by Michael Mateas and Andrew Stern uses natural language processing techniques to produce a generative interactive fiction. Even mainstream RPGs like *Fable* (Lionhead, 2004) and *Oblivion* (Bethesda, 2006) attempt to produce emergent behaviours through agent based emotional interactions.

Figure 8: *Might & Magic: Clash of Heroes* (Capybara Games, 2011) Demonstrating narrative framing of abstract puzzle mechanics.

None of these examples explores the idea in as much depth as the ASCII simulation game *Dwarf Fortress* (Tarn Adams, 2006-Present). In *Dwarf Fortress* a combination of system autonomy, player interaction and designer intent result in complex emergent forms of gameplay and narrative. The role of authorship in this scenario inevitably becomes more fluid. In an interview with designer, Tarn Adams he was asked:

Who are the 'authors' in a typical DF (Dwarf Fortress) game? How much of this role is held by you, the system and the players?

"My brother and I create the systems (including some fixed content), and the choices made at that stage are influenced by our preferences, worldviews, talents and flaws, and then the system creates the content. The players are exposed to the content and can manipulate it using the tools we (and others) create for them. How they use the tools is up to them, and how the content reacts is up to our systems. There's a common happening in computer games where a player's mind will fill in the gaps in mechanics with stories of their own, either consciously embellishing or by assuming mechanics are more complicated than they actually are. This is something that a game designer can be aware of, and it influences how I do things. It is one of the cool things about computer games that I can't find an exact analog for in other more traditional creative endeavors — perhaps magic shows or similar theater, though it isn't about misdirection so much as it is about providing enough raw material for the imagination to work with."

(Interviewed by the author, http://www.nullpointer.co.uk/content/?p=388, 2010)

Allowing procedural form to support the production of narrative in this way spreads the task of authorship between designer, games system and player. Andrew Wilmott, lead project engineer on *Spore*, expresses a similar view.

Who are the 'authors' in a typical Spore game? How much of this role is held by you the developer, the system and the players?

"..we're facilitating the player in authoring their own experiences. Clearly we're providing the mechanisms and basic game world in which those experiences are set. However we're relying on the player to supply their own internal narrative, and project their own thoughts onto the game world. We're in the business of making lego sets rather than movies, as it were.."

(Interviewed by the author, unpublished, 2010)

The metaphor of *Lego* is an interesting analogy, as it can be understood as both a procedural system and an expressive tool. Although the development of games and their consumption occur in separate environments, designers and programmers engage with procedural forms in a similar way to players. The development process may have a wider set of verbs compared to the final product but there is still a sense of 'play' within the creation of the system. The evolving dialogue between system and designer is an integral part of the development process where the act of programming a game, is often itself a game.

CONCLUSION

The relationship between Gestalt psychology and the processes of pattern negotiation in games is a vital area of research. The formal principles shared by these two frameworks are crucial for both design and analysis of the medium. The subconscious drive to process and predict patterns is a major constituent in our desire to play games, but it is also the guiding principle behind much of ludic game design. I argue that the immersion produced by negotiating procedural form is entirely different from the sensation of narrative driven engagement. Pattern immersion operates on a holistic and sub-linguistic level of interaction which can only really be understood through direct experience. This factor reflects the capacity of procedural form to generate a wide range of permutations for any scenario, including many that are unpredictable by either player or designer. With less linear control over exposition or narrative, games based on procedural form also change the relationship between game designer and player. Gamers may be exposed to the procedural mechanics of a system in a more direct manner, and given enough agency within those rulesets the role of authorship can become shared between the player, the system and the designer. Procedural form also represents a potentially infinite field of permutations for replay and exploration. Negotiating this field is an abstract

psychological challenge that brings its own reward and is an essential force in driving both the production and consumption of gaming experiences.

BIBLIOGRAPHY

Adams, T(2011) “Tarn Adams Interviewed by the author” http://www.nullpointer.co.uk/content/?p=388

Bogost, I, “Persuasive games : the expressive power of videogames” Cambridge MA: MIT Press, (2007).

Chen, J (*Flow in Games*, 2006). http://www.jenovachen.com/flowingames/thesis.htm (Accessed July 2011)

Csikszentmihalyi, Mihaly (1990). “Flow: The Psychology of Optimal Experience” New York: Harper and Row.

DeLanda, M. “Philosophy and Simulation: The Emergence of Synthetic Reason”Continuum Publishing Corporation (2011)

Gardner, M, “Mathematical Games -The fantastic combinations of John Conway's new solitaire game "life"”, http://ddi.cs.uni-potsdam.de/HyFISCH/Produzieren/lis_projekt/proj_gamelife/ConwayScientificAmerican.htm (Accessed July 2011)

Metzger, G. “Laws of Seeing”, MIT Press (2006)

Selfridge, O.G. (1959) “Pandemonium: A paradigm for learning”. “In the mechanism of thought processes”. London: H.M. Stationary Office

Wilmott, A (2011) “Andrew Wilmott Interviewed by the author” unpublished

LUDOGRAPHY

Alexey Pajitnov (1984) Tetris [Gameboy]Armor Games (2008) Shift [PC Computer]

Bay 12 Games/ Tarn Adams (2006-Present) Dwarf Fortress [PC Computer]

Bethesda (2006) Oblivion [PC Computer]

Bizarre Creations (2003) Geometry Wars [Xbox360]

Blizzard Entertainment (2004) World of Warcraft [PC Computer, Online Game]

Bulldog (1994) Theme Park [PC Computer]

Cave (2003) EspGaluda [Arcade]

Cave (2007) Mushihimesama [Arcade]

Criterion (2008) Burnout Paradise [Xbox360]

EA Digital Illusions (2008) Mirrors Edge [PC Computer]

GameLab (2001) Loop [PC Computer]

Game Yarouze (2008) Echochrome [PSP]

Gas Powered (2007) Supreme Commander [PC Computer]

Infinity Ward (2007) Call of Duty 4: Modern Warfare [PC Computer]

Lionhead (2001) Black and White [PC Computer]

Lionhead (2004) Fable [PC Computer]

Maxis (1984-2011) Sim City [PC Computer]

Maxis (2000-Present)The Sims [PC Computer]

Maxis (2008) Spore [PC Computer]

Mojang (2009-Present) Minecraft [PC Computer]

Namco (1999) Mr Driller [Arcade]

Neversoft (1999-2007) Tony Hawk's Series [Playstation]

Nintendo (2001) Pikmin [Gamecube]

PopCap (2007) Peggle [PC Computer]

PopCap (2007) Bejewelled [PC Computer]

Procedural Arts (2005) Façade [PC Computer]

Psygnosis (1994) Wipeout [Playstation]

Q Entertainment (2006) Every extend extra [PSP]

Rockstar (2001-Present) Grand Theft Auto [Playstation]

Taito (1981) Qix [Arcade]
Taito (1994) Puzzle Bobble [Arcade]
Team 17 (1995) Worms [PC Computer]
Terry Cavanagh (2010) VVVV [PC Computer]
Treasure (2001) Ikaruga [Dreamcast]
Valve (2007/2011) Portal 1 &2 [PC Computer]