LEVEL DESIGN PATTERNS

Looking for the Principles of Unified Level Design

By
Simon Larsen

Supervisor: Espen Aarseth
IT-University of Copenhagen

12-week project
April 2006

What they have learned is not just the specific rules intrinsic to a particular system; they've learned abstract principles that can be applied when approaching *any* system. They don't know how to program a VCR because they've memorized the instructions for every model on the market; they know how to program a VCR because they've learned general rules for probing and exploring a piece of technology, rules that come in handy no matter what model VCR you put in front of them.

- Steven Johnson, talking about learning abstract patterns of technology systems in his book *Everything Bad is Good for You* (2005)

Table of contents

1. Preface

The idea for this paper came about when I first heard about design patterns in an object oriented programming class at the IT-University of Copenhagen in the fall of 2004. It occurred to me that many different fields use design patterns, but very few call them what they really are; patterns. Creativity is guided by formal design tools in many fields, such as movie making, music, literature and comics. Even the art of computer game design is much focused on “common practices”, “good advices” and the paradigm of “why-fix-it-it-aren’t-broken”. Never before has the process of creating games been so complex and time consuming. The time for formal design tools is now.

Copenhagen,
April 2006.

_____

Simon Larsen

2. Introduction

My aim with this paper is to take the idea of formal design tools and show how to apply them to the process of creating levels for multiplayer first-person shooters (FPS). The focus of this paper to be on the architectural properties of the levels and not the storyline or other added elements that differ from game to game. Single player games are often very linearly structured, because they need to convey a tightly knitted storyline to the player. The levels themselves are constructed in such a way that they emphasis the storyline and underline the mood and setting that the story is conveying. Multiplayer games must to a much higher degree leave the playing field open, so to speak. To use the words of one of the leading forces behind Epic Games' Unreal Tournament series, Cliff Bleszinski: *"A Level Designer who is building for a Multiplayer-oriented title is much like a playground architect"* (2000a). The story teller is no longer present to the same extent as in a single player game. Multiplayer games are often fast-paced and center on reaching specific predetermined team-based objectives. So when looking for examples that underline the design patterns presented herein the following games are the only ones taken into consideration:

1. Unreal Tournament 2004 (by Epic Games, 2004)
2. Day of Defeat: Source (by Valve, 2005)
3. Battlefield 1942 (by DICE, 2002)

To use a famous quote from pioneering game developer Sid Meier; the aim is to look for "interesting choices" in level design. I wanted to draw the attention towards design patterns for multiplayer level design, since most of the literature available on multiplayer levels design seems to focus solely on collision points (Bleszinski 2000a) (Saltzman, 2000) (Güttler & Johansson, 2005) and I strongly believe that this is just a (small) part of a very larger picture of designing multiplayer levels. Secondly, of all the other works on design patterns for game design (see section 2.3.1 on page 7) none is focused on level design.

2.1 Problem statement

Level design is essentially is a *craft* and therefore you need proven formal tools. By using design patterns as a design tool when creating levels multiplayer games you ensure that the players can seamlessly navigate through your game world. At the same time it will greatly reduce the scope of the design process as you apply tried and tested solutions to your current problem domain. There is no need for reinventing the wheel every time you plan and design a new level. The

question that I will try to answer in this paper is; *how can formalized design patterns be used for creating interesting choices in level design?*

2.2 What are design patterns?

Design patterns are formal tools used for solving known problems. Said in another way; it is a design toolbox. In many fields, ranging from architecture, over software development to creative fields such as literature and movies, people are using some form of formal design tools to help create their work. Some call them design patterns others call them "tools-of-the-trade", but they are essentially the same; formal tools that describe problems (or problematic areas) and proven ways to solve them. If we take movies as example, try to count how many movies you have seen lately that followed a storyline similar to this one: the main character of the story sets out on a quest to undo the wrongdoings that has fallen upon him/her. During this quest, the main character faces many perils and is close to giving up near the ending, but somehow he/she prevails in the end. I would dare say that the large majority of the movies present on IMDb.com's Top 250 list[1] of the greatest movies ever made follow a storyline very similar to the above. Looking at how movies like *Indiana Jones*, the *Star Wars* movies and *The Matrix* trilogy is using the Hero's Journey[2] way of storytelling and then comparing it to the way David Lynch told the story of *Lost Highway* it is easy to spot the difference. Filmmakers such David Lynch, whom is truly artists in their field, makes movies that are not easily understandable. Ask anyone who have seen *Lost Highway* (1997) or *Mulholland Drive* (2001) of what the movie is about and you will properly end up with as many answers as people you ask. The popular movies all revolve around the same story outline. They do it because it works. It is easy to understand for the viewers, because of the familiarity of the storyline. You can argue that this type of storyline is a design pattern. Are they works of art? No, by no means! But they are all using a collection of very effective tools for creating entertainment that is easily recognizable for everyone.

The question then is; do these tools hinder the creative workflow and merely created assembly line produced entertainment that all look the same? When doing level design for multiplayer FPS, the aim is not that the player must play against the environment and solve its architectural puzzles embedded within. They must be able to instantly recognize the navigational patterns and

[1] IMDb.com is short for Internet Movie Database. They have listed the Top 250 movies of all time as rated by the users of the website. The users rate the movies from 1 to 10 with 10 being the highest. Most of the movies in the Top 250 category have more than 100.000 votes. See the chart on www.imdb.com/chart/top.

[2] The stereotypical storyline presented above is very close to what Joseph Campbell refers to as The Hero's Journey or Monomyth in his book The Hero with a Thousand Faces (1948). A monomyth is the journey undertaken by the standard mythological hero, as described by. The core concept of the monomyth is: "A hero ventures forth from the world of common day into a region of supernatural wonder: fabulous forces are there encountered and a decisive victory is won: the hero comes back from this mysterious adventure with the power to bestow boons on his fellow men" (1948, p. 30)

move fluidly through the level. The architecture must be created in such a way that the players are working *with* the environment and it is not becoming an obstacle that the player also has to overcome. More Indiana Jones and less David Lynch, so to speak.

2.3 Prior work

The idea of using formal design tools as an approach to solving issues in different fields is not new, not even in the domain of computer game design. Christopher Alexander et al. described design patterns as a formal design tool for use in the field of architecture in the book *A Pattern Language* (1977). In it Alexander writes the sentence that should prove to be the foundation for the entire use of design patterns in software development in the decades to come:

> Each pattern describes a problem which occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice. (Alexander, 1977, p. x)

Software developers Erich Gamma, Richard Helm, Ralph Johnson and John Vlissides[3] based their neo-classical book on software development *Design Patterns: Elements of Reusable Object-Oriented Software* (1995) on Alexander's description of a design pattern. They took the notion of having abstract patterns describing solutions that could be used to solve some very concrete problems in software development. It is properly within traditional software development that you can find the biggest influence of design patterns.

2.3.1 Design patterns in games

Design patterns in games have been a topic of discussion for some time now. Doug Church proposed what he saw a way of overcoming some of the very general problems involved with the process of game development and game design in his article: "Formal Abstract Design Tools" (1999). Harvey Smith has also proposed some formal design tools. During two separate presentations at the Game Developers Conference he outlined the thoughts behind his "Systemic Level design" (2002) and "Orthogonal Unit Differentiation (O.U.D.)" (2003). Common for these two presentations is that he is talking about design patterns, but he never get around to actually calling them that.

[3] They are often referred to as simple, The Gang of Four or just GoF.

Someone who did indeed call the formal design tools for design patterns were Staffan Björk and Jussi Holopainen. They presented in their book *Patterns in Game Design* from 2004 a way of using patterns in the process of designing games. Their book took the call from Bernd Kreimeier's article "The Case for Game Design Patterns" (2002). Björk and Holopainen have with their book made the definitive documentation of how and when to apply design patterns to the process of game design. Björk and Holopainen have continued to work with the aim of making design patterns for games an intricate part of the game design process. Both with "The Game Design Patterns Project" website[4] and with their newer article "Design Patterns and Games" (2006).

Noah Falstein and Bob Bates revived their "The 400 Project" project during the Game Developers Conference in March 2006. Their project is a very ambitious take on rules that makes a good game. The project was originally started by Hal Barwood and Noah Falstein in 2001, and even though they rigidly state on the project website

> No, although there are similarities. Alexander's work grew out of architecture, and is, in Hal's words "A welcomed allied analysis". But it lacks the imperative – the 400 rules are stated in terms of instructions to follow, rather than observations of existing patterns. It also lacks the trumping information that is important to understanding how these rules interact. [5]

the similarities between their project and game design patterns cannot be overlooked. As of writing this they have listed 112 rules of the 400 intended.

2.4 A very *brief history of multiplayer first-person-shooters*

In 1992 id Software released *Wolfenstein 3D* and effectively changed action games forever. The game measured a whopping 700 Kb, a huge amount at the time, but was nonetheless downloaded a quarter million times (Saltzman, 2000, p. 111) in the year it was released[6]. Since then first-person-shooters (FPS), as the genre became known as, have become one of the most popular genres.

Wolfenstein 3D was *technically* not the first action game with a first-person perspective[7], but it was without doubt the game that launched the genre. The second generation FPSs were building

[4] The Game Design Patterns Project is currently located at this web address: http://www.tii.se/play/projects/gamepatterns/index.html

[5] Text taken from the project's FAQ on: http://www.theinspiracy.com/400_project_faq.htm

[6] Notice also that the game was *downloaded* at a time where most people still have not even heard of the internet.

[7] One could argue, as many have, that the old CinemaWare's game *It Came from the Desert* (1989) and *The Terminator* based on the movie of the same name (Bethesda Softworks, 1990) are the first FPSs. Both uses the first-person perspective, they are here for the sake of the argument not considered true first-person shooters as such. More information on these games and games with the first-person perspective in general can be found on MobyGames.com.

upon the foundations id Software made with their first games *Wolfenstien 3D* (1992) and especially the two *DOOM* games (1993 & 1994).
Quake (1996) and *Star Wars: Dark Forces* (1995) showed signs of more advance gameplay, but it was not until *GoldenEye 007* (1997) and *Half-Life* (1998) that the genre really showed its full potential. The FPSs was clearly maturing as a genre and the desire to innovate the gameplay elements, spawned memorable games such as *Medal of Honor* (1999) and *Halo* (2001). This was also the period when the genre finally moved online with various offerings for multiplayer play with *Quake 3* (1999) and *Unreal Tournament* (1999) leading the masses.

One game really opened the eyes for the potential of multiplayer FPS was *Counter-Strike*. It did not start out as a stand-alone game, but as a mod for *Half-Life*. But the influence this mod had (and still very much have) on multiplayer FPSs is not to be overlooked. When it was first released in June 1999, it quickly became the most popular game to play over the internet. You can in fact talk about the eras before and after *Counter-Strike*. It is still to this day the undisputed king of online FPSs. According to Wikipedia.org *Counter-Strike* accounted for 70 percent of entire the online FPS audience in 2004. Valve, the developer of *Half-Life*, also saw the commercial potential of *Counter-Strike* and bought the rights for the mod and later turned the once free mod in to a commercial product in November 2000. Valve released a much anticipated updated version of the game called *Counter-Strike: Source* in 2004. This version utilizes the much more powerful game engine *Source* that also powers Valve's other products such as *Half-Life 2* (2004) and *Day of Defeat: Source* (2005).

Other multiplayer FPSs worth mentioning are the *Tribes* series (covering three games from 1998 to 2004), *Battlefield* series (covering three games 2002 to 2005 and numerous expansion of each game), and lastly the free *Wolfenstein: Enemy Territory* (2003) is worth mentioning for it's addition of experience points to the class system.

The sole reason that *Counter-Strike* is not included as an example is this paper is the mere fact that it is the most analyzed FPS game around. Adding one more analysis to the pile would not accompany much. Instead I have opted to look at the before mentioned games (see page 5).

3. Level design patterns

"Few things are harder to put up with than a good example"

- Mark Twain

In the following section I will present a small collection of design patterns applicable for multi-player FPS level design. The collection is by no means exhausted but should provide you with a starting point as to what design patterns could look like in connection with level design. The design patterns have come about from analyzing several levels looking for common ways of solving problematic elements of level design. The design patterns themselves will abide loosely by the Game Design Pattern Template as set forth by Björk & Holopainen (2005, p. 38-39);

Name:	A short, specific and idiomatic name whose main purpose is to be mnemonic, after reading the description.
Core definition:	A brief sentence describing the core idea of the pattern. Highlighted in bold letters under the name.
Icon:	An icon representing the design patterns. Used for quick reference when doing comparable analysis (added to Björk & Holopainen's template by this author).
General description:	Contains information of how the pattern can affect the level, in which games it was identified and examples of levels where it can be found. Because of the visual nature of level design this section is rather large in comparison with the other sections due to the amount of screenshots needed to illustrate the examples. The general description must also contain some form of motivation for the name of the pattern, if not evidently clear.
Using the pattern:	Providing examples of how the pattern could be useful in solving level design related issues.
Consequence:	Using a pattern to solve something might cause other problems or prevent other patterns from being applied. This section explains the pros and cons of using this pattern as a solution, and if any, what alternatives there are.
Relations:	Here relationships between patterns are defined. How the different patterns intertwine and interconnect. Some patterns exclude the use of others and others still need the presence of other specific patterns to work properly.
Reference (if applicable):	List the related previous works that either has been a direct inspiration for the pattern or contain descriptions of the main aspects of the pattern. This section is mostly compromised of quotes from other literary source concerning level design or design in general.

3.1 Multiple paths

Each path must be supplemented by one or more paths in order to overcome bottlenecks.

General description: In multiplayer levels, it is crucial that all main routes going from and to central objectives have alternatives. If this pattern is left out, and there is only one way to the central objectives they becomes easy to defend and almost impossible to attack, taking the pacing out of the play. The dynamic gameplay of raging battles between two teams are put to a standstill because of the bottlenecks in the architecture of the level, if this pattern is not implemented or not implemented properly.

In the Kalt level from *Day of Defeat* has a clearly defined underground path that runs under the entirety of the level. Image 1 shows the ladder leading down to this underground system of tunnels.

Image 1: The ladder leading down to the alternative under ground path.

The Avalanche level from *Day of Defeat* is a very small and compact level, but boasts an impressive number of alternative paths. Image 2 shows the main path (blue) and some of the alternative paths converging on it (red). It take no more than one minute or so to run from one end of the level to the other but the many alternative paths adds some very interesting choices in routes.

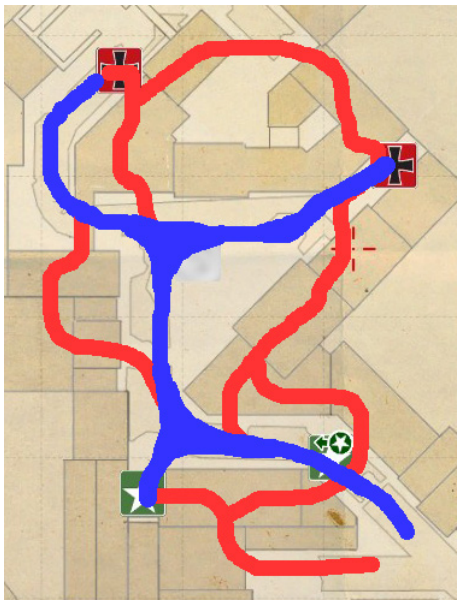

Image 2: Multiple Paths in the Avalanche level.

In the Electric Fields level from *Unreal Tournament 2004* (Image 3) the use of the *Multiple Paths* pattern is very visible. At any given place on the level the player is being offered at least two paths to choose between.

Image 3: The many paths leading to and from the location of the red flag (bottom left).

Using the pattern: In game mode such as Capture the Flag[8] it is vital that there is more than one way into and out from the position where the flag is located. You ensure the re-playability of the level by adding more paths. There will rarely emerge a "waterproof" way of traversing the level is the number of paths is sufficient.

[8] Capture the Flag or CtF is a game mode in many multiplayer FPSs were the objective is for two teams to steal the opposing team's flag and return this flag to their own base to score a point.

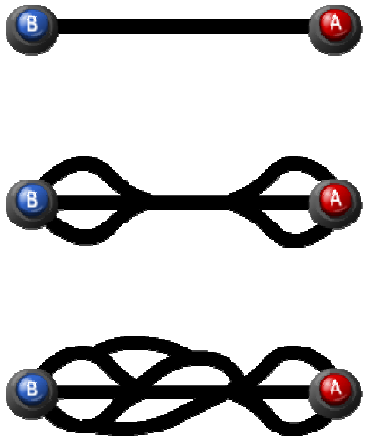

Image 4: Adding more paths to the level ensure playability.

Consequence: You avoid bottlenecks that can in worst cases lock the gameplay in a gridlock, resulting in none of the teams being able to win. Overuse of the pattern on the other hand can result in too many paths, where no players ever meet, because all players are moving of paths that seldom cross. This can although be countered with the use of the *Collision Point* pattern, and the *Local Fights* pattern.

Relations: There is a close relationship with the *Collision Point* pattern. If talking about super- and sub patterns, *Collision Point* must be considered a sub-pattern of *Multiple Paths*.

Reference: The game design literature is abundant with advises pointing out the importance of giving the players more choices. Game designer Sid Meier coined the famous quote about game being "a series of interesting choices", and continues in a recent interview:

> We maintain the philosophy that a game's a series of interesting choices, of directions that you give the player, and we always try to ensure there's a couple of alternative paths - one will be the choice that looks the best in your current circumstance, but you're always thinking: 'Next time I play, I might try that instead: it sounds interesting as well'. (Sid Meier interviewed in EDGE - #159, February 2006)

3.2 Local fights

Break up the level in smaller areas that are more or less closed of the rest of the level.

General description: This is the "Think Global – Act Local" of level design patterns. Large levels have to broken up into smaller parts than can sustain themselves. Large areas might seem impossible to cross if it remains *too* open. There must be places to hide or to change direction that will leave eventual pursuers bewildered.

The Flash level from *Day of Defeat* is a good example of a long main route going from the Allied spawn point to the Axis spawn point (Image 5). The design of the level forces the players to

engage in close combat city battles without knowing whether the enemy is right around the corner or not.

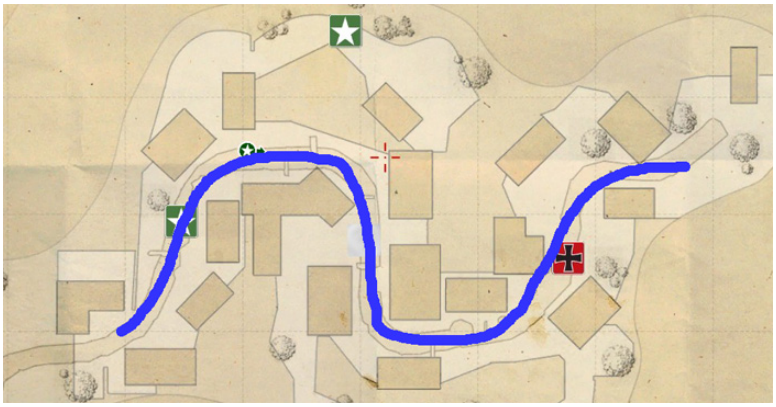

Image 5: The main route is broken into smaller parts (the route is highlighted with blue).

The Grassy Knoll level from *Unreal Tournament 2004* takes another approach to the same problem. Instead of having the route from the two bases go by small roads the level has been opened up and a large area sits between the two bases. But the area is not entirely open. It has been broken up into smaller areas by adding large trees, stones and hills as it can be seen in Image 6.

Image 6: Grassy Knoll level from Unreal Tournament 2004.

The Bridge of Faith level from *Unreal Tournament 2004* (Image 7) consists of some very large rooms that the players have to traverse getting from base to base. But none of these rooms seems overwhelmingly open due to the strategic placement of some large columns.

Image 7: Large columns divide the very large rooms into smaller areas. (A clear reference to the scene from "The Mines of Moria" from Peter Jackson's movie *Lord of the Rings* (2001)).

Even in very large and very open level such as the Toburk level from *Battlefield 1942* the areas are broken up into smaller areas by making the terrain hilly as seen in Image 8.

Image 8: The hilly terrain in the Toburk's vast open desert breaks up the level into smaller areas.

The name and icon from *Local Fights* comes from fact that the closer you are to your opposing players the more fun the game becomes. There is absolutely no fun in running around a level and constantly being fired upon by some faraway located sniper. Or as army general William Prescott said leading his men in the famous *Battle for Bunker Hill*: "Don't fire until you see the whites of their eyes"

Using the pattern: When creating a corridor, break the corridor in two or more parts. This maintains the felling of going down a long corridor, but removes the feeling of being too much out in the open. The risk of crossing a long open corridor or area might seem too high for most players. Breaking the area into smaller areas gives the players a feeling of having only to cross smaller and less risky areas to get to their location.

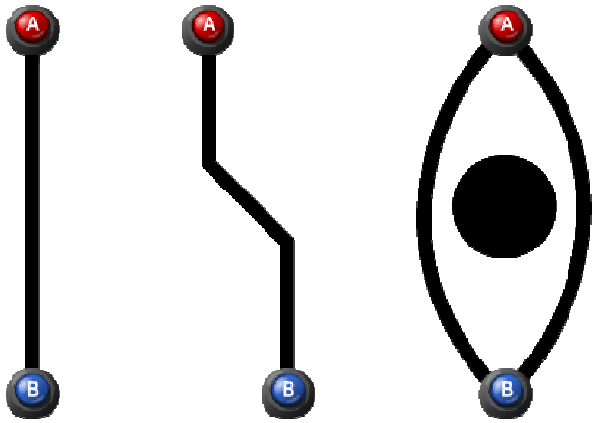

Image 9: Breaking the path between to points by turning the path or adding obstacles.

This design pattern also becomes viable when designing large open outdoor levels. Make the outdoor areas into hilly terrain by breaking the level apart by adding hills, mountains, lakes and forests to the scene and thereby creating smaller sub areas.

Consequence: If large areas are included in a level and not broken up using the design patterns, the areas might become a "sniper paradise", making the area impossible to traverse for players. A potential conflict might occur with the *Risk Incentive* pattern since that pattern is partly based on the fact that some objects need to be located in open areas to provide balance to the level.

Relations: This pattern can have a relation to the *Reference Point* pattern – The lake or trees used to break up the level might serve a double purpose as a reference point (as described in the pattern of the same name).

3.3 Collision points

The paths of opposing players must cross at some point to create tension in the level.

General description: When playing multiplayer games one of the key elements is meeting other players and playing against them. The paths that lead from one team's area to another, or to and from an important objective in the level, must cross so that members from both teams will face each other at some time.

The overview map for the Kalt level from *Day of Defeat* (Image 10) clearly marks the *Collision Points*. At least two places (center) the routes for both teams will collide head on. This provides some very interesting gameplay as both teams try to push the other back and conquer the objectives that are located on both sides just beyond these collisions points.

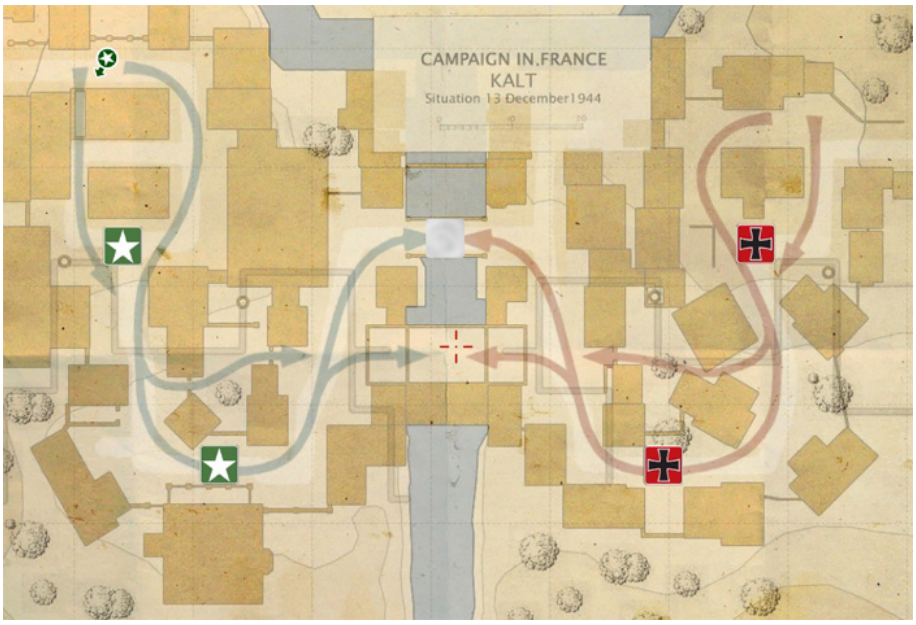

Image 10: The collision points in the Kalt level are clearly visable.

Another way of creating interesting collision points it by make the only way to enter the opposing teams base go through some very narrow spots as in the Maul level from *Unreal Tournament 2004* (Image 11). Here two rather small holes in the dividing wall serves as clear collisions points that makes up for most of the battle in this level.

Image 11: The two holes in the wall create unsurpassable collisions points.

Using the pattern: If you are making a Capture the Flag level, construct the level in such a way that all players must go through a central area (be that either a room or a specific outdoor area). By doing that you ensure that the players will eventually run into each other at some point.

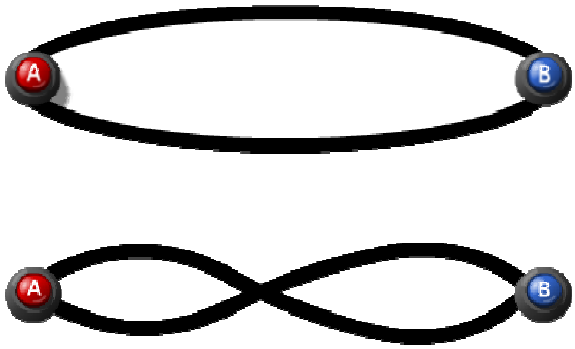

Image 12: Making the main paths cross produces interesting situations for the players.

If used correctly the level can rise from being confusing and mediocre to being a tension filled experience. This is also one of the best design patterns for accentuate to the players that they are playing in a multiplayer environment, since they are hereby guarantee to run into other players.

Consequence: If the map only contains one collision point, it is imperative that the "time-to-contract" for each team reaching these points is the same. Locating the collision point too close to one of the teams' main defense objectives renders the level unbalanced, hence unplayable.

Relations: There is a relationship between this patterns and the *Local Fights* pattern, the *Collision Points* patterns being a sub-pattern of the latter. Using the *Collision Points* pattern is one way of ensuring that at least one *Local Fight* will be present if implemented correctly.

Reference: Christopher Alexander talked about adjusting the layout in city planning so that you would create areas of the community that would concentrate the activity in so-called *nodes* (1977):

> Pattern #30: Activity Nodes: Create nodes of activity throughout the community [...]. First identify those existing spots in the community where action seems to concentrate itself. Then modify the layout of the paths in the community to bring as many of them through these spots as possible. This makes each spot function as a 'node' in the path network. (p. 166)

But the most vigorous analysis of collision point in level design comes from Güttler & Johansson (2005) in their article on the topic:

> The play patterns are formed on the foundation of the spatial design of the level and the behavior of the players. [...] These tactics are all based around the collision points of the level; points, that are noted by the two teams mission oriented paths verge on or cross each other on one or more locations through the level. (*Own translation*, p. 156)

3.4 Reference points

Always provide reference points in your level to help navigation.

General description: The player must never feel lost. *"A player will never accuse you of being* too *helpful, unless you are blatantly telling him how to beat each challenge before he has had a chance to work it out alone."* (Byrne, 2005, p. 61). The aim of a multiplayer level is to produce a fluid game playing experience for all the players. The players should be challenged by the level but never to such an extend that the players are fighting *the level* and then the opposing players. It should always be the other way around.

The navigational aides do not always have to be large buildings or monuments but can be implemented as in the Kalt level from *Day of Defeat* where the piping in the underground looks distinctively different accordingly to whether you are under the Allies' or the Axis' area of the level (Image 13).

Image 13: The piping in the underground shortcut looks different under the Allies' and Axis' area.

The bridge in the Anzio level (Image 14) and the church in the Avalanche level (Image 15) from *Day of Defeat* serves as a great way of communicating enemy moment to follow team members, "The Germans are moving over the bridge" or "He's sitting in the church tower". Since these reference points are *unique* in the levels they occur, all players would instantly recognize the location of such messages.

Image 14: The bridge provides a recognizable reference point.

Image 15: The church ruin works well as a reference point.

Using the pattern: There is no harm in applying interesting architectural elements to your level if you just always keep in mind that the players must be able to move about in the level without too much frustration. Use the power of reference points to aide navigation. When using the pattern it is also imperative that the reference points providing are *unique*. You cannot navigate through a level that has five similar blue roads and much less give directions to another player.

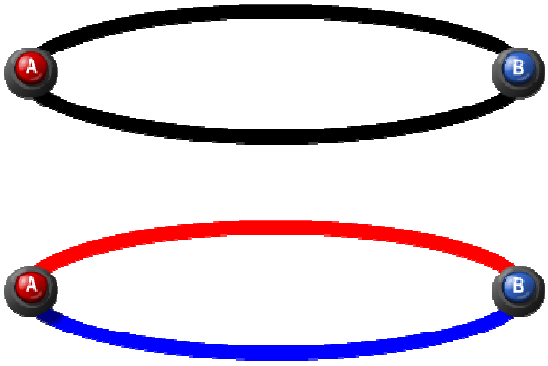

Image 16: It is not always sufficient to refer to the different routes as "left" and "right". Make the routes unique by adding reference material directly into the scenery.

Consequence: Overuse can lead to the players experience loosing their willing suspension of disbelief. The aim is to add these reference points as subtle as possible.

Relations: This pattern has a loose relationship with the *Multiple Path* pattern. As important as it is to provide more paths for players it is just as important to make the paths distinctive.

Reference: Although this refers to the architectural challenges in single-player levels the advice given can be directly transferred to multiplayer levels.

> If your goal is to create an environment that is totally alien, it pays to periodically give your audience something familiar to anchor them themselves to. [...] If you can periodically give them some reference point... such as, "Oh, I am in a spaceship" or "Hey, this must be the engine room" you will be doing them a great favor. (Carson, 2000).

Ed Byrne writes in his book *Game Level Design* (2005) about the importance of keeping continuity in your visual style:

> Although the player may not be scholar in historical architecture, mixing different decorative periods, such as Art Deco [...], Gothic [...], and Megalithic [...] will simply confuse the player and make him wonder if he skipped to the next level by accident. (p. 284)

Marc Saltzman (2000) makes the case for well placed landmarks as navigational cues:

> Landmarks can serve a dual purpose in well-designed maps: They can add to the immersion factor while offering navigational cues to players, helping them know their direction and location in the level. (p. 126)

3.5 Defense areas

Aide the players or team defending objects by making the architectural layout of the level work to their advantage.

General description: Most levels for multiplayer action oriented games revolve around one team attacking another team's location or skirmish over specific control points. In either case team-members frequently needs to defend these areas or objectives. Because the defenders do not know when or where the attackers might come from they have a disadvantage. This can be counter by giving them objects that can help them defend.

In the Anzio level from *Day of Defeat* the defense of the bridge can be done from a ruin located nearby (Image 17). Here the defenders can partly unseen watch all the traffic crossing the bridge and can quickly duck for cover if under enemy fire.

Image 17: The defense of the bridge, as seen from the attacker's point of view (left) and the defenders (right).

More open areas, as in the Avalanche level from *Day of Defeat* (Image 18) and the Iwo Jima level from Battlefield 1942 (Image 19) can be aided with the addition of sandbags or sandbags-like structures that the defenders can cover behind.

Image 18: The sandbags defense of the German flag.

Image 19: Defense of the hill in the Iwo Jima level.

Alternatively the defenders can be given extra hardware to help with the defense. This can be either stationary guns or special buttons that can close of areas or shut doors. In the case of the El Alamein level from *Battlefield 1942* (Image 20) the defenders have been provided with a heavy anti-aircraft gun and a high powered stationary machine gun with unlimited ammunition.

Image 20: The left image shows an anti-aircraft gun and to the right is a mounted machine gun.

Using the pattern: Create areas surrounding important objective in the level with elements that can help the defenders of the object defend. That being either providing elements that they can seek cover behind or adding hardware that aide in the defense.

Consequence: If the defense area becomes too powerful it will effectively bring the level to a standstill with the attacker having no way of overrunning the defense area. So this is one pattern that should be used with a lot of thought. A good way of using this is to combine it with the *Multiple Paths* pattern, making the defense only cover one entrance to the objective and then have one or more alterative paths leading into the area surpassing the defense.

Relations: There should be a relation between the use of *Defense Areas* and the *Reference Points* patterns. For both the attackers and the defenders using the *Defense Area* it is important that it is easily recognizable so that communication about events taking place at these areas can be easy conveyed to follow team members.

3.6 Risk Incentive

Access to wanted objects in a level must be connected with some element of risk

General description: High powered weapons, health packs or shortcuts are very common in levels, but in order to keep the level balanced there must be some risks connected with the goal obtain of these. In many levels the use of shortcuts will fit into the domain of this design pattern. They are often very high risk but will get you to your destination much faster.

The shortcuts provided in the Donner level from *Day of Defeat* (Image 21) and the Double Damage level from *Unreal Tournament 2004* (Image 22) both give the players an advantage if they success in using them, but there are also some considerable risks connected with them. In the Donner level, you can not turn back without surely getting a bullet in the back if meeting an opponent in this narrow passage and in the Double Damage level taking the shortcut through the shallow water makes you very visible to other players and an easy target for skilled snipers.

Image 21: This short cut quickly becomes a death-trap if a player meets an opponent.

Image 22: The route through the shallow water is much faster but leaves the players vulnerable to enemy fire.

The placement of the Double Damage power-up (not to be confused with the above mentioned level of the same name) in the Bridge of Faith level from *Unreal Tournament 2004* (Image 23) is another good example of the *Risk Incentive* pattern. The power-up is located at one of the most visible place in the entire level making any player seeking to take it easy target for all others.

Image 23: There is a high risk connected with trying to get the power-up.

Using the pattern: If you have a route going from A to B, make a straight route go from the two points. Place this shortcut in a highly visible place, so that it becomes clear for all the players when one is trying to take advantage of this shortcut. Similarly place power ups and powerful weapons in very open areas making obtaining them a risky undertaking.

Consequence: Using the patterns gives the players choices, they will have to individually decide whether or not it is worth the risk trying to traverse a shortcut or obtain a power-up.

Relations: This pattern is loosely related to the *Multiple Paths* pattern. If the multiple paths in a level provide some form of shortcut the *Risk Incentives* pattern must also be applied to the level. Another relation is to the *Defense Areas* pattern, using the hardware provided to aide the defense should also be connected with some form of risk, such as it takes time to enter and exit an anti-aircraft gun or that the visual area is limited when using a stationary machine making you easy target to anyone trying to flank you.

Reference: Cliff Bleszinski (2000a) elaborates on "the beauty of risk incentive", and the power of giving the players the choices instead of taking it for them:

> The player weighs the risk; he assesses the challenge, and gets to make a decision. He feels like he's in control, and the designer provides him with a choice. [...] In a Deathmatch style game a player will have the choice of going for an ass-kicking weapon, only if he risks his neck by going into an extremely open and well guarded spot. Never underestimate the usefulness of this technique.

Ed Byrne (2005) makes the same argument:

> Allowing the player a choice if they need something badly enough to risk more than they would gain to get it is the basis of many gameplay systems and events in game level design. The judicious but intelligent use of different levels of risks coupled with large and small rewards can make all the difference. (p. 103)

Björk & Holopainen presented their *Risk/Reward* design pattern in their book *Patterns for Game Design* (2004). The patterns is very similar to the one presented here, the key difference being the area of application. Theirs is belonging to the domain of game design and this pattern to the domain of level design. Their description of the pattern:

> Interesting choices in games must have the potential for both advantageous and disadvantageous effects. Although players do not normally strive to have the disadvantageous effects, these may be unavoidable, and performing the actions at all can depend heavily on the chance of gaining the Reward and the risk of gaining the Penalty. This kind of decision-making is based on Risk/Reward choices (p. 376)

4. Conclusion and future research

"It's all very well in practice, but will never work in theory".

- Anonymous

The craft of designing levels have in the resent decade, along with the rise of FPS, become a full-fledged profession comparable to programming and art direction. All professions need formal design tools and it is time that level design got its. The presentation of the level design patterns herein merely scratches the surface. There are many more patterns waiting to be formalized and many more to discover. My aim with this paper was show the usefulness of design patterns both in connection with design levels but also in connection with analyzing them. They can provide a much needed vocabulary to the discussion. You could easily state "This level is boring… add more *Multiple Paths*" or "This area is too open and wide, add some more *Local Fights*". It is the same with movies (and other non-interactive narratives); if the main character seems dull or un-believable you give him an inner conflict. That is an established way of making him much more interesting. The same goes with music. Keep the lyrics but changes the beat or add another verse. If the players of your level complain about it being monotonous to play then go back through these pages and try and add one or more of the design patterns offered. It will most certainly enhance your level.
It is not about streamlining all levels and only creating levels after the same template. That would result in indistinguishable generic levels a looks and play alike. The level design patterns presented herein should instead be looked upon at as tools. As paint brushes to help you paint you canvas. There is no silver bullet to be found for designing levels.

The work ahead is now to look for additional and more complex patterns that can be used in the analysis and design of future levels. There is a need for level design patterns for single player games as well. My hope is just that the level design patterns can remain as abstract as possible so they do not become watered down iteration of the same patterns presenting the same solution to the same problem domain in different ways. Increasing the sheer number of patterns achieves nothing. The aim is to keep them as hands-on and relevant as possible.

5. References

All links checked as of April 2006.

Alexander, Christopher, et al.: *A Pattern Language: Towns, Buildings, Construction*. Oxford University Press, 1977. ISBN: 0195019199

Bjork, Staffan & Holopainen, Jussi: "Game Design Patterns". 2003. Available online at: http://www.nokia.com/library/files/docs/Game_Design_Patterns.pdf

Bjork, Staffan & Holopainen, Jussi: *Patterns in Game Design*. Charles River Media, 2004. ISBN: 1584503548

Bjork, Staffan & Holopainen, Jussi: "Games and Design Patterns" in Salen, Katie (ed.) & Zimmerman, Eric (ed.): *The Game Design Reader*. The MIT Press, 2006. ISBN: 0262195364.

Bleszinski, Cliff: "The Art and Science of Level Design". Game Developers Conference 2000. Available online at: http://www.gamasutra.com/features/20010110/cliff_01.htm

Bleszinski, Cliff: "Unreal Tournament Level Design Musings". Game Developers Conference 2000. Available online at: http://www.cliffyb.com/rants/ut-ld-tips.shtml

Brown, Duncan & Chen, Steven: "The Architecture of Level Design". Gamasutra.com, 2001. Available online at: http://www.gamasutra.com/resource_guide/20010716/chen_01.htm

Byrne, Ed: *Game Level Design*. Charles River Media, 2004. ISBN: 1584503696

Campbell, Joseph: *The Hero with a Thousand Faces*. HarperCollins, 1993. ISBN: 0586085718

Carson, Don: "Environmental Storytelling: Creating Immersive 3D Worlds Using Lessons Learned from the Theme Park Industry". Gamasutra.com, 2000. Available online at: http://www.gamasutra.com/features/20000301/carson_01.htm

Church, Doug: "Formal Abstract Design Tools". Gamasutra.com, 1999. Available online at: http://www.gamasutra.com/features/19990716/design_tools_01.htm

Gamma, Erich et al.: *Design patterns: elements of reusable object-oriented software*. Addison Wesley, 1995. ISBN: 0201633612

Güttler, Christian: *Gennemtænkt leveldesign - en nødvendighed?*. Master thesis from The IT-University of Copenhagen, 2002.

Güttler, Christian & Johansson, Troels Degn: "Principper for level design af multi-player first-person shooters" in Jessen, Carsten (ed.) & Walther, Bo Kampmann (ed.): *Spillets verden*. Danish University of Education Press, 2005. ISBN: 8776130665

Kreimeier, Bernd: "The Case for Game Design Patterns". Gamasutra.com, 2002. Available online at: http://www.gamasutra.com/features/20020313/kreimeier_01.htm

Saltzman, Marc (ed.): *Game Design - Secret of the Sages*, 2nd edition. Brady Publishing, 2000. ISBN: 1566869870.

Smith, Harvey: "Systemic level design for emergent gameplay". Game Developers Conference 2002. Available online at: http://www.gamasutra.com/features/slides/smith/index.htm

Smith, Harvey: "Orthogonal Unit Differentiation". Game Developers Conference 2003. Available online at: http://www.planetdeusex.com/witchboy/gdc03_OUD.ppt

5.1 Games

Unreal Tournament 2004 developed by Epic Games., 2004. Published by Atari.

Day of Defeat: Source developed by Valve, 2005. Published by Valve.

Battlefield 1942 developed by DICE, 2002. Published by Electronic Arts (EA).

Counter-Strike: Condition Zero developed by Ritual Entertainment & Turtle Rock Studios, 2004. Published by Sierra Entertainment.