The Science of Level Design

Kenneth Hullett
University of California, Santa Cruz
Santa Cruz, CA 95064
khullett@soe.ucsc.edu

ABSTRACT

Level designers create gameplay through geometry, AI scripting, and item placement. There is little formal understanding of this process, but rather a large body of design lore and rules of thumb. We have taken a first step in improving understanding by identifying design patterns in first-person shooter (FPS) levels, providing cause-effect relationships between level design elements and gameplay. The next step is to validate this theory with a series of experiments that test the strength of these relationships. Analysis data collected from subjects can improve understanding and provide designers with scientifically verified tools for creating gameplay in FPS levels.

Categories and Subject Descriptors

D.2.2 [**Software Engineering**]: Design Tools and Techniques, H.1.2 [**User/Machine Systems**]: Human factors, K.8.0 [**Personal Computing**]: Games

General Terms

Design, Human Factors

Keywords

level design, game design, design patterns, user testing, data mining, player modeling

1. INTRODUCTION

FPS games are combat-oriented games where the player engages other characters with a variety of projectile and melee weapons. The player navigates a 3D world while looking through the eyes of the main character (i.e., a first-person point of view). A screenshot from a typical FPS is shown in Figure 1.

Play in FPS games is divided into discrete levels, built by level designers. They construct the geometry of the level, place items in it, and populate it with both friendly and enemy NPCs for the player to encounter. They use level design to create gameplay, but there is little formal understanding of how the design elements they employ create gameplay.

While numerous books have been written on FPS level design [3][4][5], they mainly convey design lore without providing a deep understanding of how to create gameplay. Experienced level designers draw from their knowledge of existing games and have an intuitive feel for what features create different types of gameplay. However, they lack a language to describe their design ideas. There is a need for a structured way for designers to communicate their intentions and to pass their knowledge on to less experienced designers.

To address this problem, we have begun identifying design patterns in first-person FPS levels. Our use of design patterns is inspired by their use in the domains of architecture [1], software engineering [4], and game design [5][2]. The design patterns we have identified are described in terms of how they are used by designers, the concerns the designers must address when they use them, and the gameplay they create.

Figure 1: A screenshot from *Halo 3,* a popular FPS game.

The taxonomy of design patterns is a useful tool for improving designers' abilities to communicate design ideas and as a reference for possible features to incorporate into levels. However, the process by which it was created is necessarily subjective. Designers' intentions in using certain features may vary, and how players react to the patterns may vary.

To improve understanding of the relationships between design patterns and gameplay, we propose to conduct a series of experiments. A combination of qualitative and quantitative assessments will be used to determine if patterns have the expected effects on player behavior. If significant deviations from the expected results are found, we can adjust the theory, improving our understanding of the relationships, and increasing the usefulness of the taxonomy as a tool for level designers.

2. RESEARCH QUESTION

The goal of this research is to gain a deeper understanding of FPS level design and how it creates gameplay. The main research question is: Do design patterns create the intended gameplay effects?

To validate the cause-effect relationship between design patterns and gameplay, a series of user tests will be conducted. If the expected behavior occurs when a player encounters a design pattern in a level, then the theory is validated. For example, if a

FDG 2010, June 19-21, Monterey, CA, USA

pattern is expected to lower the tempo of a level, then upon encountering it, the player should begin moving more slowly and cautiously. If the expected behavior does not occur, we can adapt the theory to match the observed results.

Validating the design patterns will increase their usefulness to level designers by proving the connections between level design elements and gameplay. To do this, we will have to show the effects of adding or removing a design pattern, as well as the different effects of variations within the pattern.

3. DESIGN PATTERNS

The use of design patterns to better understand levels is inspired by their use in software engineering [7], which were in turn inspired by design patterns in architecture [1]. Kreimeier was among the first to adapt the concept of design patterns to the domain of digital games by identifying game design patterns [8], and the idea was extended by Björk et al. [2].

The descriptions of patterns explain how they can be used, the concerns designers must address, and the gameplay created. The fields are listed below:

Description – A high level description of the pattern and the major design considerations

Affordances – Aspects of the pattern that the designer can vary

Consequences – A description of the gameplay the pattern creates

Relationships – How the pattern interacts with other patterns

Examples – Some examples from popular commercial games that illustrate the pattern

We have initially identified ten patterns, grouped into one of four categories based on the type of gameplay produced. The categories are:

- Patterns for Positional Advantage
- Patterns for Large-scale Combat
- Patterns for Alternate Gameplay
- Patterns for Alternate Routes

Below we present an example of the *sniper location*, one of the patterns for positional advantage. To see the complete pattern collection, visit the authors' website at:

eis.ucsc.edu/LevelDesignPatterns

3.1 Example Pattern

Description. Sniper locations are one of the most common patterns. A character in a sniper location can attack other characters with long-range weapons while remaining protected. Any elevated position that overlooks some portion of the level is potentially a sniper location. They may be intended for use by either players, NPCs, or both.

Creating a sniper location for use by an enemy rather than the player requires additional consideration. Enemies positioned in the sniper location may require special scripting to create the desired behavior; they should remain in place, using cover if available, and engage the player with long range weapons.

Affordances.

- The height of the sniper location over the main part of the level
- How large of an area is available for the sniper
- The amount of cover available for the sniper
- The size of the area that the sniper can cover from the sniper location
- How accessible the sniper location is from the area overlooked

Consequences. When confronted with an enemy sniper location, the player is forced to make careful use of cover or seek alternate routes to avoid being exposed to fire. This can increase the tension and slow the pace of a level while creating a challenge for the player.

A player sniper location generally slows the pace of a level while lowering tension as the player is able to engage enemy NPCs without being exposed to enemy fire. However, if the sniper location is not isolated from the rest of the level, the player will have to defend the access point as well, increasing tension.

Relationships. Sniper locations interact with many other patterns. They may be placed to cover an arena or a choke point. Most stationary turrets are also sniper locations. A gallery is specialized type of sniper location.

Examples. In the level "Route Kanal" of *Half-Life 2*, the player encounters an enemy sniper location, shown in Figure 2. It is high above the player's position, but has very little cover. The player can engage the enemy NPCs, but is exposed and needs to be cautious.

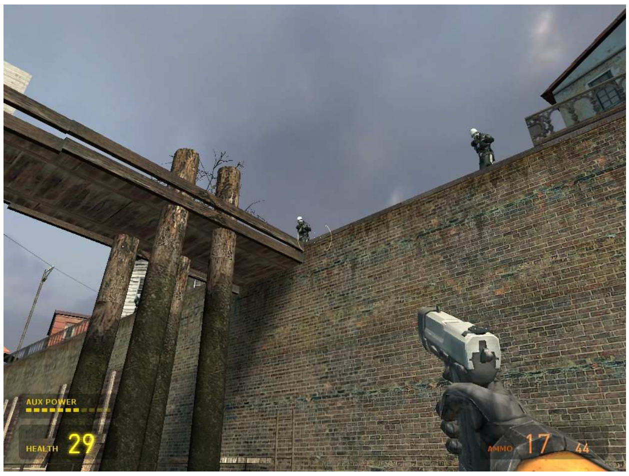

Figure 2: Sniper location in *Half-Life 2*

There is a sniper location in the level "Corinth River" of *Killzone 2*. The player is on an elevated walkway overlooking a medium-sized area containing enemy NPCs. Both the player and enemy NPCs have cover, but by looking down from above, the player is able to locate the enemy NPCs and engage them.

4. VALIDATING CAUSE-EFFECT RELATIONSHIPS

The relationships between level design elements and gameplay suggested by the design patterns will be validated by user testing. We propose experiments where subjects will play a series of levels designed to show the effects of design patterns. While it is unfeasible to test every possible variation of every design pattern, we can create representative levels that use a variety of patterns in typical ways. A combination of qualitative and quantitative data analyses will be used.

4.1 Quantitative Analysis

All data about in-game actions will be logged while the subjects play the levels, including:

- Movement – frequency and speed of movement, amount of time spent standing still, frequency of jumping
- Items – what items are picked up and how often
- Weapons – which types are used and how accurate the player is with each one
- Exploration – percentage of the map explored
- Combat tactics –average distance to enemy NPCs, average distance engaged from, number of enemy NPCs eliminated, amount of damage taken

Quantitative analysis will include applying data mining techniques to the collected data. Techniques such as k-means clustering and emergent self-organizing maps (ESOM) have been used to find player models from recorded gameplay data [6][9]. These techniques find clusters in the data, grouping subjects based on similar patterns of behavior exhibited throughout a level. For example, if subjects generally move quickly and engage enemy NPCs at a short range, they would be classified as aggressive players. The percentage of subjects in each class, as well as what classes will be found differs from level to level.

For each level used in testing, player models will be created from the recorded data. Based on the classes in the player model we can say something about the level – for example, if most subjects are classified as exhibiting cautious behavior, then the design elements used in the level are creating cautious behavior. This behavior can be correlated with the expected gameplay from the design patterns used in the level.

Player models from different levels can also be compared. Again, this can be correlated with the design patterns used in the levels. If a level contains patterns that are intended to increase pace, then the player models for that level should show a large percentage of subjects in a class exhibiting behavior associated with a fast pace (i.e., moving quickly and frequently) when compared to a level containing patterns that slow the pace.

4.2 Qualitative Analysis

Video recordings of the subjects' play-throughs can be used for qualitative analysis. The videos will be coded with observed patterns of behavior and instances where the subjects' behavior changes noticeably. For example, if a subject begins moving more quickly or engaging enemy NPCs more aggressively, we will note this change and where in the level it occurred. These changes in behavior can then be correlated to the expected behavior resulting from the design patterns.

Common behavioral patterns we expect to observe include:

- Movement speed – Does the player move slowly or quickly?
- Aggressiveness – Does the player engage enemy NPCs directly and at close range, or do they move slowly, use cover, seek the path of least resistance, and only engage enemy NPCs when opportune?
- Weapon use – Does the player prefer certain weapons, or do they switch weapons depending on the situation and the type of enemy NPC encountered?
- Navigation – Does the player follow the intended path, or do they get deviate from it frequently?
- Exploration – Does the player explore the level thoroughly, or do they move directly from objective to objective?

The coded play-throughs will be analyzed with respect to the design patterns contained in the level to see if there is a correlation. If a change in behavior typically occurs when subjects encounter a design pattern, we can say there is a correlation. If, over a large percentage of the subjects, the change in behavior does not match the theory presented in the taxonomy, or no change occurs, we can update the design patterns with the observed behavior.

4.3 Example

Imagine a level featuring a large arena containing a large number of enemy NPCs. For the experiment we create two variant levels – one where the player enters directly into the arena, and one where the player enters the arena in a strategically places sniper location. Since the sniper location design pattern is intended to slow the pace of the level, we would expect our analyses to show this occurred in the 2nd version.

In our analysis of the play-through video, we would expect to note the subject exhibiting behavior corresponding to a slower pace upon encountering the sniper location, i.e., moving very little, using cover, and engaging enemy NPCs with long range weapons. In the player models for the levels, we would expect to see a higher percentage of subjects in a class of more cautious play when compared to the player model for the 1st level.

5. GAMES CITED

Halo 3, Bungie Software, 2007.

Half-Life 2, Valve Software, 2004.

6. REFERENCES

[1] Alexander, C. "A Pattern Language." Oxford University Press. 1977.

[2] Björk, S., Holopainen, J. "Patterns in Game Design." Charles River Media. 2004.

[3] Bryne, E. "Game Level Design." Charles River Media. 2005.

[4] Clayton, A. "Introduction to Level Design for PC Games." Charles River Media. 2003.

[5] Co, P. "Level Design for Games: Creating Compelling Game Experiences." New Riders. 2006.

[6] Drachen, A., Canossa, A., and Yannakakis, G. N., "Player Modeling using Self-Organization in Tomb Raider: Underworld." in Proceedings of the IEEE Symposium on Computational Intelligence and Games. 2009.

[7] Gamma, E., Helm, R., Johnson, R., and Vlissides, J. "Design Patterns: Elements of Reusable Object-Oriented Software." Addison-Wesley. 1995.

[8] Kreimeier, B. "The Case For Game Design Patterns." Gamasutra. March 12, 2002. www.gamasutra.com/features/20020313/kreimeier_03.htm.

[9] Pedersen, C., Togelius, J., and Yannakakis, G. N., "Modeling Player Experience in Super Mario Bros." in Proceedings of the IEEE Symposium on Computational Intelligence and Games. 2009.