Examination of Player Enjoyment and Learning with Explicit Versus Implicit Tutorials

Campbell Crapsey, Myque Ouellette

***Abstract*—The study demonstrates the important differences between Explicit and Implicit tutorials and how they affect players. Through examining player enjoyment, the thesis seeks to prove that players prefer Implicit tutorials over Explicit tutorials. In addition, the hypothesis states that the player learns new mechanics from Implicit tutorials at the same speed that they learn from Explicit tutorials. The thesis demonstrates the effect on player learning from the application of a variety of tutorial techniques. Testing also reveals some unexpected effects on player behavior such as emergent gameplay and player experimentation. This thesis attempts to codify the best tutorial practices in the video games industry.**

***Index Terms*—Games, Learning, Gameplay, Tutorial**

I. Introduction

THERE exist two dominant approaches within the games industry on the subject of teaching new players how to play a game. The first approach uses specific in-game tutorials to teach players new mechanics in elaborate detail (explicit communication) and the second method focuses on the idea of in-game tutorials that utilize implicit communication. The general distinction between the two methods comes from the way in which the game delivers information to the player. Explicit tutorials pause play or control forward progress by using instructional text and audio, as well as overt HUD elements whereas implicit tutorials teach new mechanics without instructional text, audio, or overt HUD elements. The game teaches the player how to play using explicit tutorials while players teach themselves how the game works through the aid of implicit tutorials.

The importance of the tutorial divide in the industry comes down to how players interact with a game from the very beginning. No game wants to risk a poor introduction since that might lose player interest before the player encounters the gameplay highlights. However, the player must learn how to play before they can progress and the balance between those two desires can cause games to succeed or fail. Many studios prefer a tutorial based on explicit communication that walks the player through every action required to play, even at the risk of boring the player at the very beginning of the game. However, there has been a consistent push to refine tutorials to use more implicit communication that teaches players while still affording them an open gameplay experience. The movement rests on the notion that both the desire to start with strong gameplay and the need to teach the player can complement each other without explicit step-by-step instruction.

To increase common knowledge on what makes a successful tutorial, this paper and the associated research artifact examine both sides of the tutorial debate. The first focus of this paper explores how well players learn a new mechanic from an implicit tutorial versus an explicit tutorial. Next, the research examines the degree of enjoyment that the player experiences from each type of tutorial. This thesis seeks to challenge the belief that the player requires explicit direction in order to learn a new mechanic. It proposes that players can not only teach themselves a new mechanic but that they enjoy the process more if they are not explicitly instructed on what to do. To test this hypothesis, testers play one of three versions of the same tutorial: one tutorial that has explicit tutorial techniques, one tutorial that uses tutorial techniques to implicitly direct the player, and one version that displays no tutorial cues whatsoever (to serve as a control group). This thesis replicates the most successful implicit and explicit tutorial techniques used in the gaming industry. Those techniques emerged from extensive research on tutorial techniques throughout the industry.

This thesis uses both common explicit and implicit tutorial techniques to evaluate which approach better teaches gamers as well as which approach gamers find more enjoyable. The explicit techniques include interrupting text pop-ups, lock-stepping (gating player progress as they move through the tutorial), explicit critical path markers, and non-interactive video to teach the player new game mechanics. The implicit tutorial technique set utilizes popular methods such as down-time, necessity of action, checkpointing, NPC foreshadowing, adaptive messaging and positive reinforcement. All these tutorial techniques aim to teach the player how to play the game quickly, but differ on the subject of preserving player immersion at all times. The thesis seeks to demonstrate that players have strong positive feelings for implicit tutorials and strong negative feelings for explicit tutorials while learning the new mechanics at the same rate.

Manuscript received September 23, 2012. This work was supported in part by the effort of Professor Myque Ouellette of the Guildhall at Southern Methodist University.

Campbell Crapsey was with Guildhall at Southern Methodist University, Plano, TX 75024 USA.

Myque Ouellette is with the Guildhall at Southern Methodist University, Plano, TX 75024 USA.

II. Research Review

In order to better understand industry tutorials the research focused on identifying the most popular and successful techniques within tutorials. By better understanding the components that make both implicit and explicit tutorials successful it became far easier to create tutorials using those techniques and test the success of both styles. Subsequently, that body of research reduced the list of techniques to only those used in the most successful tutorials. To determine best practices, extensive research evaluated critical response to game tutorials and subsequently focused on those games that were widely viewed as most successful.

Explicit techniques evaluated for this thesis include several industry standard tutorial components. The list of both Explicit and Implicit techniques emerged from analysis of a variety of games (listed below), as well as from several summit presentations from various Game Developers Conferences in both the United States and in Europe, including "*Limbo*: Balancing Fun and Frustration in Puzzle Design," by Jeppe Carlsen, "Teaching Players: Tutorial and Opening Mission Design for *Company of Heroes*," by Neil Jones-Rodway and Aldric Sun, "10 Tutorial Tips from *Plants vs. Zombies*," by George Fan, and finally "Techniques for In-Level Storytelling," by Steve Gaynor.

- **HUD Popups:** Helpful information appears on the HUD that acts as a reference guide for the player. The pop-up does not pause gameplay but remains until the player either uses the information or the pop-up disappears after a timer expires.
- **Lock-stepping:** The player encounters interrupt messaging that introduces the new mechanic and then allows them to perform the first step of the mechanic. The tutorial then presents a congratulatory message on successful completion coupled with instructional text for the next step of the mechanic.
- **Breadcrumbs:** Highlighting the critical path leads the player in the correct direction while allowing them to move about under their own power. The player possesses the freedom to explore, but are clearly guided in a particular direction. Using this technique to show pickups or critical parts of level layout ensures that the player knows how to play the game as intended.
- **Non-Interactive Video:** Walks the player through an action without giving them control during the sequence. The player observes the game perform the actions at the right time within the context of the situation. While the game becomes a passive experience, the goal remains that the player knows how to play once the video completes.

Implicit techniques evaluated for this thesis include several common teaching tools. As mentioned previously, both Explicit and Implicit techniques emerged from the same sources (listed above the Explicit techniques as well as in the Bibliography at the end of the document).

- **Down-Time:** The player encounters a gameplay environment that does not hurry or threaten them, which encourages experimentation and exploration.
- **Necessity of Action:** Each level section requires the player to successfully learn and apply a new mechanic.
- **Checkpointing:** Player progress remains preserved throughout the entire level, which reduces the risk of death. This encourages the player to experiment and explore. Death becomes a learning tool instead of a prohibitive concern.
- **NPC Foreshadowing:** The NPC tutorial fits into gameplay to preserve the player's immersion. An NPC shows the player how to perform a complex action and the player then attempts the action themselves. NPC foreshadowing occurs within gameplay, and does not take control away from the player or break immersion.
- **Adaptive Messaging:** Visual help that only displays if the player continues to struggle with a mechanic. Players who successfully complete the mechanic never encounter the message while players who are struggling see the help tip and avoid further frustration.
- **Positive Reinforcement:** Aural or visual congratulatory messages let the player know the precise moment they do something correctly. This reinforces the action and encourages the player to perform the action again in a similar situation.

A. Explicit Techniques

Arc System Works . *Persona 4 Arena.* Atlus, 2012. XBOX 360.

Persona 4 Arena (P4A) focuses on creating highly competitive fighting. In order to bring players up to speed as fast as possible P4A uses explicit tutorials broken down into 46 steps that walk the player through all the moves available to them. While slow, it allows players to optimize their learning so that they can practice the skills before competing with other players.

P4A tells the story of four high school students watching a competitive fighting tournament. Each student is a persona of a famous fighter and channels their ability to fight once they enter into the tournament itself.

P4A provides an excellent example of explicit tutorials because it uses many common explicit tutorial techniques. The tutorial calls itself a tutorial, which immediately makes the player aware that they are about to learn. The tutorial is broken down into 46 steps that use interrupt messaging in the middle of the screen so that the game is not only paused several times within each gameplay step ("lock-stepping" the player) but the messages cover the screen as well (figure 1). In addition, extra HUD popups scroll down the side of the screen to serve as a reference guide. All the HUD popups deliver necessary information that the player uses as a reference guide as they learn each mechanic. When a player becomes confused, they evaluate the tips on the HUD and try again.

Figure 1: Persona 4 Arena HUD Popups

Persona 4 Arena uses explicit tutorials to try to speed players through the learning process to get to the later game. These explicit techniques diminish early game fun and make the first two hours of the game about learning and not experimenting.

Riot Games. *League of Legends.* Riot Games, 2009. PC.

League of Legends (LoL) stands out as the most successful "MOBA" (Multiplayer Online Battle Arena) to date. Inspired by the Warcraft III "Defense of the Ancients" mod, LoL pits teams of players against each other across symmetrical battle maps. Players gain experience and gold by killing both enemy players and waves of AI minions, which they then use to unlock better abilities/items within the match. The culmination of each match revolves around the winning team destroying the enemy team's base.

The game systems players must master for *League of Legends* makes LoL one of the most complicated games in the market right now. LoL utilizes explicit tutorials to teach those intricate mechanics and systems to new players as quickly as possible. The most common tutorial method in *League of Legends* delivers explanation of game systems by using non-interactive video walkthroughs to show the player how to play. This tutorial technique exists wholly separate from the gameplay itself and involves the player watching the video and then reenacting that gameplay to play well.

LoL provides an excellent example of explicit tutorial techniques because Riot believes that their game needs instruction up front so players can learn and then apply that knowledge.

EA Canada. *Fifa Soccer 13.* Electronic Arts, 2013. XBOX 360.

Fifa Soccer 13 puts the player in control over hundreds of professional soccer clubs from around the world to play soccer matches against other players or AI teams. *Fifa Soccer 13* extensively uses explicit critical path highlighting (Also known as "breadcrumbs") to teach the player when to shoot, when to pass, what type of pass to use in a given situation, and when to use offensive moves such as counter-attacking. Although critical path most often means how to get from point A to point B in a linear game, within a sports game the critical path refers to how to score points most effectively or how to prevent the opponent from scoring points while on defense. Within the standalone tutorial, *Fifa* relies on HUD arrows to show the counter-attack direction and optimum player positioning when using the counter-attack command (figure 2).

Figure 2: Fifa Soccer 13 Critical Path HUD Elements

Fifa relies on explicit instruction for teaching the player how to evaluate the flow of the game and how to maximize their opportunities on offense and defense. This tutorial style quickly delivers nuanced tutorial information to the player so that they recognize similar situations in-game and react appropriately. *Fifa* keeps their explicit tutorials separate from the actual game modes so that players can choose whether they want to figure it out for themselves or spend time learning some tips and improved gameplay behaviors so they improve their performance quickly.

B. Implicit Techniques

Christopher Paul. "Optimizing Play: How Theorycraft Changes Gameplay and Design." *GameStudies.org,* May 2011. October 15, 2012.

Theorycraft refers to the practice of analyzing game systems and maximizing their potential. The power of theorycrafting is that no explicit tutorial system is required to teach the player how to play. The entire principle of theorycraft relies on the absence of in-game explanation of how to play. Players encounter an environment where they are required to experiment in order to play, and that experimentation drives the player's learning, replacing the possibility of a step-by-step tutorial with player intelligence and greater immersion.

The article discusses Theorycraft in *World of Warcraft*, which is the player-driven practice of analyzing the game systems in *World of Warcraft* and applying that study to maximize the player's ability to play well. The concept behind Theorycraft is that players must have an opportunity to improvise and experiment without consequence to better learn the game systems. *Optimizing Play* continues to delve into the relationship between player interest and known game systems, discussing the positive outcomes of not knowing the exact details behind the systems themselves, thus driving the player to continue experimenting. At one point, the article quotes the Blizzard designer "Ghostcrawler."

> "There is a risk that players will stop experimenting and theorycrafting if they think we will eventually just dump all of the answers on them. We like for players to experiment with gear, talents and the like. Having black boxes adds depth and a sense of exploration to the game. When everything is known with certainty, you can do things like definitively know the best choice in every situation. Theorycrafting is dead." (2009, post 97)

This article and this quote strongly relates to game tutorials because theorycrafting time gives the player an opportunity to familiarize themselves with the game systems and begin to experiment with their capabilities. That environment should be incredibly low risk and should encourage experimentation (not overt dictation) for precisely the same reasons.

Aki Jarvinen. "First Five Minutes: How Tutorials Make or Break Your Social Game." *Gamasutra.com,* April, 2010. October 16, 2012.

The article examines how the social/mobile development space utilizes simple and accessible mechanics so that tutorials can bring the player up to speed on how to navigate and interact with the game world within seconds. Although social/mobile games usually use explicit tutorials, games like Farmville and Angry Birds are extremely successful because the tutorials are condensed and quick to get the player into the game afterwards. In this way, the social/mobile space represents the side of the industry that focuses on explicit tutorials for time consideration.

First Five Minutes discusses how social game developers seek to capture the attention of new players within the first few seconds while still teaching the game to the player. The focus for social developers is to not only to deliver an accessible user interface but also to give the impression of activity and purpose that game tutorials so often lack. Most tutorials depict very simple, stripped-down levels to ease the player into the game, but many social/mobile studios have found that notion can cost them players because the initial environment is very stark and inhospitable.

ThatGameCompany. *Journey.* Sony Computer Entertainment, 2012. PlayStation 3.

Journey serves as a critically acclaimed example of tutorials that rely on minimal instruction. Unlike most games that incorporate explicit instruction for new abilities and highlighting critical path, *Journey* focuses on player experimentation to master new player abilities and discover each level's critical path. *Journey* delivers gameplay that challenges the player but does not threaten them very often with death from enemies or environmental hazards (incorporating "Down-Time"). Very little of *Journey* focuses on survival, but instead emphasizes solving puzzles. That feeling of safety allows the player to exit out of "fight or flight" mode and instead approach the mechanic from a problem-solving standpoint. Simple visual instruction within the game world provides players with tutorial tips, including images of the controller on the HUD (figure 3 – controller image), flowing ribbon objects that serve as location indicators, and pop-up icons to indicate timing. These let the player know when they can interact with an object which creates player curiosity while the environment enables the experimentation itself. The environment remains calm and mostly still, which enhances the visibility of each tutorial notification while reassuring players that they have time to explore and experiment with the game systems.

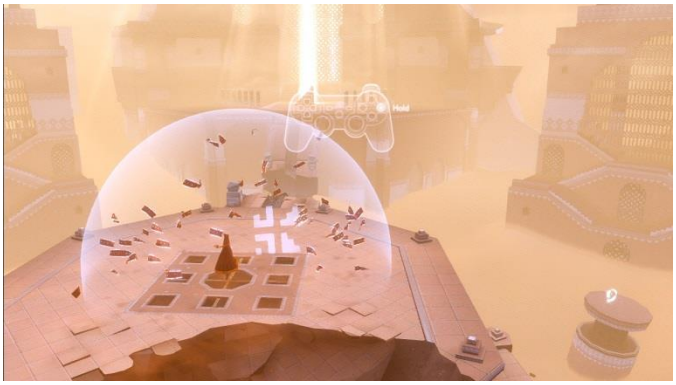

Figure 3: Journey gameplay image to show UI

Journey focuses on "experiential" gameplay that encourages the player to explore and find new ways of interacting with the world. Using complex cloth and sand physics as well as creative storytelling, the game delivers an impactful experience with only the very basic tutorial information (a ghost-image HUD icon for the correct button to press). *Journey* encourages positive player behavior by adding to the player's robe (the source of their power) when they reach new areas by using their character abilities in the correct manner (figure 3 – ribbons).

According to conventional logic, this game tutorial would not adequately teach the game mechanics because of the lack of aural instruction and visual guidance. Large sections of the game utilize flat lighting and no overt instruction on new abilities other than a pop-up HUD image that appears after a pause if the player does not press the correct button ("adaptive messaging") to encourage the player to push that button and activate their new ability. However, the game appealed to a massive number of people and went on to critical acclaim and commercial success. The effortless, consequence-free gameplay environment throughout *Journey* enabled players to try their new mechanics on specifically placed nearby objects intended for use with the ability. *Journey* serves as an excellent example of delivering minimal instruction for new abilities. They enabled that success by creating an environment that does not threaten the player, thus encouraging the player to experiment with gameplay and teach themselves.

PlayDead. *Limbo.* PlayDead, 2010. PC.

Limbo tells the story of a young boy who journeys through a strange, dream-like, abstract world and jumps, pushes, pulls, and swings his way through a hostile environment. PlayDead built a game that uses frequent player death as a teacher. *Limbo* uses an unusual tutorial approach because traditional teaching techniques focus on player safety, not player jeopardy. However, gameplay that kills the player with high frequency reduces player aversion to death, and becomes a method of experimenting. In order to enhance the effect *Limbo* uses new death animations as a visual reward for the action to make death fun for the player. The game also utilizes one of the more robust checkpoint systems in modern gaming. By protecting player progress and not player safety, *Limbo* reflects a new trend toward using death as a tutorial mechanic.

Limbo represents an important take on fun tutorials because the player learns through death, which is highly unusual for a tutorial system. The player never truly loses any progress due

to "checkpointing" and the game challenges them to solve the same puzzle immediately. The effect is that the player does not feel the failure of death as a significant consequence and therefore keeps playing. *Limbo* allows players freedom of experimentation by turning death into an integral part of gameplay progression.

Valve. *Portal 2.* Valve, 2011. PC.

Portal 2 represents a strong movement in tutorial theory that focuses on seamless integration of the tutorial into the actual gameplay. When successful, the player does not realize that they are in a tutorial because the tutorial and the game are not distinct from one another in tone, play style, or pacing, and because the experience is just as much fun as the rest of the game.

Chell returns from the first *Portal* and through her eyes the player revisits the same complex puzzle platforming that defined the game. *Portal 2* takes that puzzle platforming component from the first game and introduces fluid-based platforming and a stronger, lengthier narrative.

Portal 2, much like the original *Portal*, teaches new mechanics throughout 80% of the game. With such an extended "tutorial", the Valve team had to approach the game from the standpoint of making learning fun. Their approach was to give a sense of accomplishment for mastering a new skill before allowing the player to progress on to the next challenge (and subsequent mechanic), thus introducing a "necessity of action" for every new mechanic. The game breaks down into a string of small successes that keeps the player interested in the game and still teaches the player new mechanics every few minutes. Once the player learns the new mechanic they encounter a situation where they must master it to solve the puzzle, and then they move on to the next challenge where the game introduces a new gameplay mechanic or combination of past mechanics. Each puzzle incorporates ample gameplay down-time, meaning the player has as much time as they need to solve each puzzle, which encourages more creative thinking.

The short levels of *Portal 2* ensure that the player always knows when they have done something right. The necessity of action empowers the player because they only move to the next stage of the puzzle when they have solved a movement puzzle. That "positive reinforcement" keeps the player motivated and interested by providing a continuous sense of progress. Portal 2's tutorials stay fun for players, even during subsequent playthroughs because of the positive reinforcement and its beneficial impact on players.

Portal 2 shows that learning itself can be fun. Instead of structuring a game around teaching a mechanic and then playing with that mechanic, Valve integrated player learning and player reward into the same game space which maintains player immersion and the gameplay's sense of fun.

Valve. *Half-Life 2.* Sierra Entertainment, 2004. PC.

Half-Life 2 (HL2) presents the player with a shattered future Earth that the protagonist, Gordon Freeman, must defend against the dominion of the alien Combine. The player uses a wide variety of tools and weapons to infiltrate enemy lines and keep the human resistance alive all while bending to the will of the mysterious G-man.

Figure 4: NPC mechanic foreshadowing

HL2 does an excellent job of using implicit "NPC foreshadowing" to teach the player how to use special weapons, especially the gravity gun. Midway through the game one of Gordon's companions, Alyx, takes Freeman outside and teaches the player how to catch and throw objects with the gravity gun (figure 2). The player ends up playing a game of catch with a third NPC, Dog, which teaches the player all the intricacies of the gravity gun mechanic. That NPC walkthrough differs from similar explicit techniques such as lock-stepping or non-interactive movies, because the entire tutorial takes place within the game world and works hard to avoid breaking immersion. That walkthrough teaches the player intricate gameplay behavior while remaining within the game narrative.

The implicit and explicit tutorial techniques used by these games as well as many others seek to teach the player new gameplay mechanics quickly and efficiently. The difference between the two intentions focuses on initial game experience. The explicit style relies on overt commands to teach the player the correct game behavior while sacrificing initial game immersion. Explicit tutorials frequently serve as "separate" experiences from the game proper. The explicit set includes both interrupt dialogue text pop-ups and "lock-stepping" the player through an extended gameplay sequence, using overt HUD and in-game visual indicators for critical path (not considered a part of game world logic), and non-interactive videos to show gameplay mechanics. Implicit tutorial techniques rely on game immersion and provide the player the tools they require to figure out how to play on their own. The implicit techniques used for the thesis artifact include introducing necessity of action and gameplay down-time, minimizing the penalty of death through frequent checkpoints, adaptive messaging to warn players of continued gameplay errors, visual cues for critical path, positive reinforcement for good gameplay actions, and NPC foreshadowing of gameplay mechanics. For the purpose of this thesis the implementation of implicit versus explicit techniques occur independent of each other. While many games use a combination of implicit and explicit techniques to teach the player how to play, this thesis tests them independently to discover the effect and player enjoyment of each overall approach. The inclusion of Down-Time and Necessity of Action are the only exceptions due to their prevalence in so many games and the difficulty involved in taking them out of a tutorial while still protecting the quality of that tutorial.

III. Methodology

This study addresses the industry-wide challenge of teaching players mechanics in an environment that promotes fun and immersive gameplay. Many games use explicit tutorials to teach new mechanics with the assumption that players benefit from quick, active instruction. Players can quickly get to the game with a more thorough body of knowledge versus longer tutorials that encourage the player to teach themselves through experimentation and active participation. This thesis challenges the notion that a tutorial must be a passive experience utilizing explicit communication and examines design techniques that incorporate fun gameplay into tutorials. In order to evaluate the success of different tutorial techniques this thesis focuses on teaching the player new gameplay mechanics with both explicit and implicit techniques. This thesis then evaluates which set of techniques better taught the new mechanics and which set of techniques players found more fun. The testing evaluates how quickly players learned the new mechanics through test proctor observation of the player's actions and a timed playthrough. A quick pre-test and post-test survey evaluates demographic information, player enjoyment and tutorial clarity.

This thesis focuses on answering the question, "Do players prefer a gameplay tutorial that uses implicit instruction over explicit instruction and do they learn at the same speed with implicit instruction?"

The overall test evaluates three levels that teach identical mechanics but differ in terms of the specific tutorials. Each tester plays only one version of the level, whether that is the explicit, implicit, or control version of the level. The thesis proctor randomly chooses which specific version a tester plays before the test begins. Each subject plays through one level with the goal of finishing the level by assassinating the local commander. The level structure ensures that the player must learn how to use every mechanic in order to finish the level.

Each level uses *Half Life 2: Episode 2* (Valve, 2007) as the engine base and the Source Software Development Kit as the level editor. A web survey and the test moderator's own recorded evaluation of the test represent the basis for evaluating each player's comprehension and enjoyment.

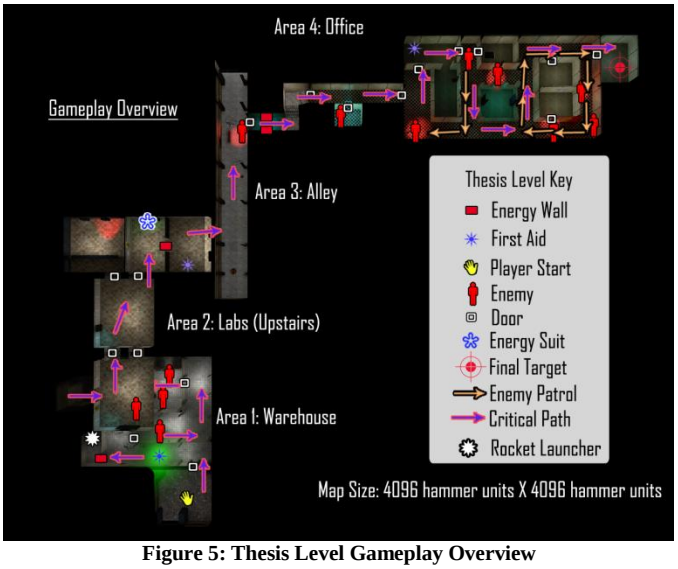

Figure 5: Thesis Level Gameplay Overview

All three level versions include the same set of mechanics, presented in the same order to the player. Changes to the base *Half-Life 2* controls ensure that even experienced players must learn new gameplay mechanics. Although non-standard shooter controls may have increased player frustration, the risk posed by experienced players testing the game from memory threatened to skew the results even more. Therefore, the level relied on remapped keys for reload, shoot, crouch, and use, while eliminating sprint. Once the player completes the intro cutscene and enters the Warehouse, they encounter a first aid station that restores their health (the player starts the level at low health). This mechanic focuses on player vitality, which represents a basic concept in gaming. The player then passes by an Energy Wall that damages the player if they try to walk through. The Energy Wall teaches the player that certain environmental objects are harmful to the player. If the player picks up the Energy Suit (located much farther on in the level), they can walk through any Energy Wall without taking damage. To test player creativity the first Energy Wall contains a Rocket Launcher pickup that the player can use if they backtrack after finding the Energy Suit.

Next, the player engages a group of enemies and must kill them using their rifle. The mechanic of acquiring targets and then firing the weapon at those targets should also be a fairly easy mechanic for most gamers to pick up because so many popular games use it to create gameplay. In addition, any damage the player takes during this section of the level reinforces the first mechanic (health) by removing small amounts of health per hit.

After the player engages enemies, they obtain the Energy Suit that allows them to walk through the Energy Walls to reach previously unreachable sections of the level. Walking through Energy Walls is not intuitive to the player and therefore poses a challenge in communicating how the mechanic works. It is important to evaluate intuitive and less-than-intuitive mechanics within the same test to properly determine the overall effectiveness of explicit and implicit techniques.

Once the player shoots out the windows of the Lab area and they drop down into the Alley, they encounter the first set of automated turrets. The turrets challenge the player to crouch

and sneak by the red detection cone. Weapon fire cannot damage the turrets; therefore, players can only bypass the danger.

Last, the player must sneak through the Office past automated turrets and super soldiers that are much harder to kill. This final area requires all of the mechanics that the player previously learned. Players use their new skills to reach the local commander's room (the assassination target). Sneaking through the Office represents the most challenging mechanic to learn. The complexity involves identifying what provides cover, understanding when enemies see you, and that players must sneak by turrets in order to progress through the section and kill the final target.

This thesis tests teaching gameplay mechanics, therefore the design focus remains how the level teaches the mechanics. There are three varieties of the testing level: One in which all given instructions are explicit, one in which they are implicit, and one control level that has no tutorial system at all to ensure that this thesis tests the merits of tutorials without any unknown coloring the results.

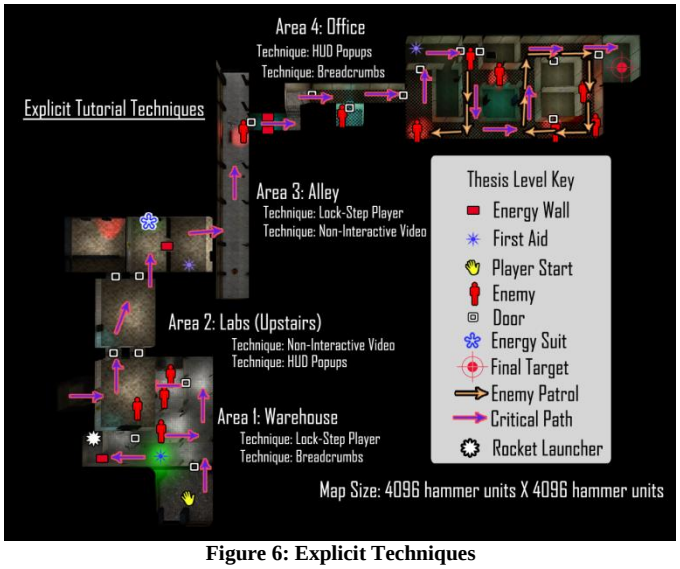

Figure 6: Explicit Techniques

The explicit tutorial techniques used in the testing of this thesis come from games that focus on delivering information quickly so that the player starts the game with an enhanced understanding of how to play. The following list of explicit techniques derived from the Research Review.

- **Down-Time:** The player encounters a gameplay environment that does not hurry or threaten them, which encourages experimentation and exploration.
- **Necessity of Action:** Each level section requires the player to successfully learn and apply a new mechanic.
- **HUD Popups:** Helpful information appears on the HUD that acts as a reference guide for the player. The pop-up does not pause gameplay but remains until the player either uses the information or the pop-up disappears after a timer expires.
- **Lock-stepping:** The player encounters interrupt messaging that introduces the new mechanic and then allows them to perform the first step of the mechanic. The tutorial then presents a congratulatory message on successful completion coupled with instructional text for the next step of the mechanic.
- **Breadcrumbs:** Highlighting the critical path leads the player in the correct direction while allowing them to move about under their own power. The player possesses the freedom to explore, but are clearly guided in a particular direction. Using this technique to show pickups or critical parts of level layout ensures that the player knows how to play the game as intended.
- **Non-Interactive Video:** Walks the player through an action without giving them control during the sequence. The player observes the game perform the actions at the right time within the context of the situation. While the game becomes a passive experience, the goal remains that the player knows how to play once the video completes.

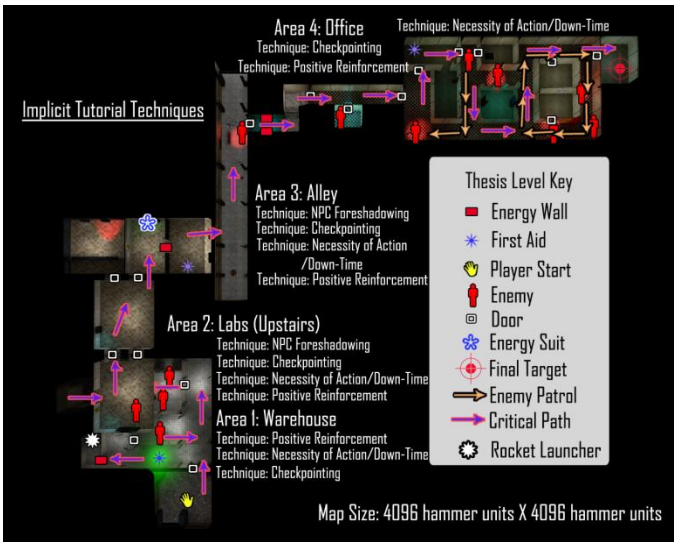

Figure 7: Implicit Techniques

The core implicit tutorial techniques of this project come from games that actively experimented with enabling the player to teach themselves the mechanics. The following list of techniques derived from the Research Review.

- **Down-Time:** The player encounters a gameplay environment that does not hurry or threaten them, which encourages experimentation and exploration.
- **Necessity of Action:** Each level section requires the player to successfully learn and apply a new mechanic.
- **Checkpointing:** Player progress remains preserved throughout the entire level, which reduces the risk of death. This encourages the player to experiment and explore. Death becomes a learning tool instead of a prohibitive concern.
- **NPC Foreshadowing:** The NPC tutorial fits into gameplay to preserve the player's immersion. An NPC shows the player how to perform a complex action and the player then attempts the action themselves. NPC foreshadowing occurs within gameplay, and does not take control away from the player or break immersion.

- **Adaptive Messaging:** Visual help that only displays if the player continues to struggle with a mechanic. Players who successfully complete the mechanic never encounter the message while players who are struggling see the help tip and avoid further frustration.
- **Positive Reinforcement:** Aural or visual congratulatory messages let the player know the precise moment they do something correctly. This reinforces the action and encourages the player to perform the action again in a similar situation.

The Control version of the level exists to ensure that both the Explicit and Implicit versions of the level actually teach the mechanics. To prove this, the Control version forces testers to play without any tutorial information or the standard *Half-Life 2* HUD. That way, even experienced players encounter an unfamiliar gameplay environment that requires them to learn new game mechanics.

The expected testing result is that the player not only prefers implicit instruction but also learns the game mechanics in roughly the same amount of time. The thesis testing focuses on creating three levels that each present the same mechanics to the player, although each uses different techniques to do so. The test proctor evaluates level of comprehension by observing the tester's progress on a second screen and timing how quickly players move through each level section. The position of the second monitor allows the test proctor to view the test subject's face throughout the test. A quick pre-test and post-test survey evaluates player enjoyment and tutorial clarity.

Example questions asked during pre-test survey:

1. Do you consider yourself a gamer?
2. Circle your Bartle Gaming Psychology type or N/A if you aren't familiar with the term.
    a. Explorer
    b. Achiever
    c. Killer
    d. Socializer
3. How many hours per week do you play video games (Pre-Guildhall for Guildhall students)
4. Have you ever played Half-Life 2 or its episodic content?
    a. No
    b. Yes, <10 hours
    c. Yes, 10-30 hours
    d. Yes, 30-50 hours
    e. Yes, 50+ hours
5. What is your favorite gaming genre? (Select One)

Example questions asked during post-test survey:

1. How much did you enjoy the level overall (Scale 1-7)?
2. Where in a campaign would you place this level?
    a. Beginning
    b. Middle
    c. End
3. What type of game tutorial do you prefer (multiple selection)?
    a. Standalone Explicit (i.e. FIFA, Persona 4 Arena)
    b. Integrated In-Game Explicit (i.e. Fallout, Halo)
    c. Experiential Tutorial (i.e. Journey)
    d. Long Form Implicit (i.e. Portal, Limbo)
    e. No Tutorial
4. Final Thoughts (Open Answer)

The player must answer the following questions for each level area. There are four level areas (Warehouse, Labs, Alley, and the Office).

1. How much did you enjoy this section (Scale 1-7)?
2. What mechanic(s) did you learn in this section?
3. How did you learn those mechanics?
4. Were you frustrated at any point during this section? Why?

The third question of the post-test survey, "What type of game tutorial do you prefer?" uses terms not previously identified in this thesis. "Standalone Explicit," "Integrated In-Game Explicit," "Experiential," and "Long Form Implicit" all describe nuanced approaches to Explicit and Implicit tutorials. Due to the nature of the test, the evaluation of tutorial techniques focused on the success and failure of the application of Explicit tutorial techniques as a whole and Implicit tutorial techniques as a whole. However, to try and gauge the specific preference of the testing group the question uses more specific terminology for testers to clearly indicate their preferences. Standalone Explicit tutorials relate to Explicit tutorials that exist as separate entities from the main game, like the tutorial videos for *League of Legends* or the tutorial walkthroughs for *Persona 4 Arena*. Integrated In-Game Explicit tutorials represent the most commonly used tutorial system. They focus on up-front Explicit tutorials to teach the mechanics quickly and then immediately transition to the rest of the game. Experiential tutorials expose all the mechanics to the player at once and leave it up to the player to discover each mechanic's value. Finally, Long Form Implicit tutorials, such as those found in *Portal*, provide the player an opportunity to thoroughly teach themselves a new gameplay mechanic before providing additional mechanics later in the game.

The testing subjects are a broad mix of different ages, nationalities, and gamer types chosen by convenience (volunteers). The thesis tested the topic by obtaining 31 total testers split across the three test versions (Explicit, Implicit, and Control). Testing occurred within a dark room with minimal outside noise and interference. Testers played their test level on a single screen laptop using a mouse and keyboard all provided by the thesis author. All testers participated in a pre-test survey to provide demographic information and described their gaming habits, as well as a post-test survey where they described their gameplay experience.

Professor Myque Ouellette set the criteria and requirements of the product, and the Masters Student worked with Professor Ouellette to maintain completion for every milestone. The schedule for this thesis began with testing of the thesis research artifact on the 11th of February and continued for

three weeks until the 4th of March. During that time, the thesis author organized the testing data in addition to acting as test proctor. Once testing concludes, the thesis author analyzed the data and writes the findings by March 31st. The analysis of the data did not require any special tools other than a series of Excel spreadsheets.

IV. Results

The testing pool consisted of 31 randomly assorted testers. The test proctor ensured even tester distribution across the three levels. 11 testers played the Explicit version of the level, 10 testers tested the Implicit version, and the final 10 testers played as part of the Control group. The Explicit group contained 11 testers, because the test proctor thought it in the best interests of the research to obtain as many testers as possible in the given amount of time. All testers were between 20 and 30 years of age, with an average age of 26. Testers marked the number of hours they played games every week. Four testers played less than ten hours/week, ten testers played 10-20 hours/week, eleven testers played 20-30 hours/week, three testers played 30-40 hours/week, and three testers played an average of over 40 hours/week. The test proctor randomly assigned testers to a version of the level, independent of their familiarity with video games. The following pie chart visualizes the breakdown in hours of gameplay per week.

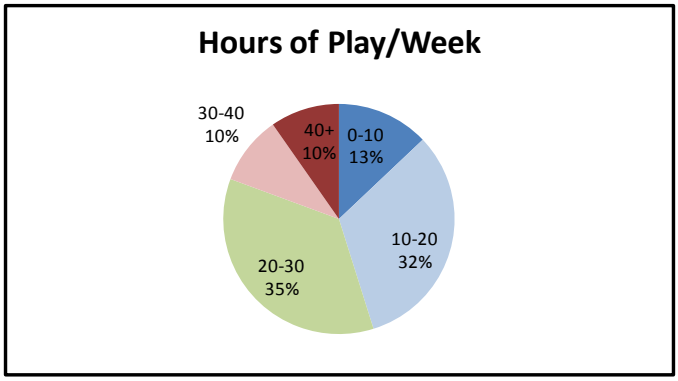

Figure 8: Hours of Play per Week

How many hours testers play games per week identifies whether or not players are familiar with common gameplay mechanics, which directly relates to how necessary testers found the tutorial instruction during their test experience. Having a broad mix of tester experience strengthens the validity of the results since it captures the effect of different tutorial types on both inexperienced and experienced gamers.

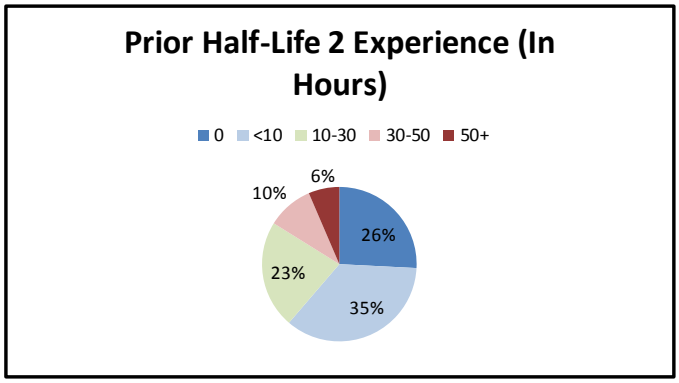

Figure 9: Half-Life 2 Experience

Another pre-test question seeks to establish player familiarity with first person shooters and the *Half-Life* franchise explicitly. The question represents an important distinction from overall game experience, because the research artifact operates as a modded version of the First-Person Shooter (FPS) *Half-Life 2: Episode 2* (Valve, 2007). If most testers were overly familiar with FPS gameplay or with *Half-Life 2* (HL2) gameplay or controls then it could minimize the effectiveness of tutorial instruction. With similar gameplay, experienced players would not need to pay close attention to some of the tutorial information. 8 testers had never played HL2, 11 testers had played less than 10 hours, 7 played 10-30, 3 played 30-50, and 2 testers played HL2 and subsequent episodes for over 50 hours. As the chart above this paragraph shows, the majority of testers (19 out of 31) had not played long enough to finish the *Half-Life 2* campaign, which takes about 20 hours for most players. That data suggests that a significant majority of testers were not familiar enough with *Half-Life 2* or the sequels to ignore the tutorial instruction and play the level from memory or from being overly familiar with the gameplay.

While the test proctor did not record other demographic information, such as gender or race, all participants identified their Bartle gaming type. The Bartle gaming types are based on the theory that all player gameplay preferences can be summarized using an x/y coordinate graph that measures player desire to socialize with other players versus exploring the game world, and preference for unilateral action versus cooperative gameplay. The Bartle composition of the total testing group was 5 Achievers, 7 Killers, 1 Socializer, and 17 Explorers, and 1 person that did not identify.

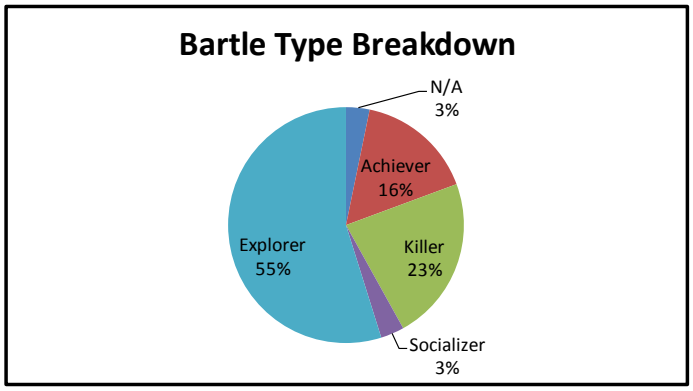

Figure 10: Tester Bartle Types

The intention behind recording this information sought to uncover any trends between Bartle types and play experience. Due to the skewed nature of the results (far too few of every type besides Explorers to reliably measure effect) this data only demonstrates the random collection of test participants.

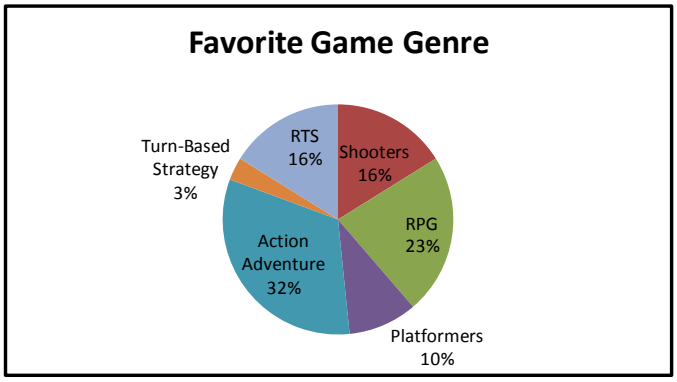

Figure 11: Tester Genre Preference

The final pre-test question addresses tester's favorite gaming genre. Testers responded to this question to provide additional information on their experience playing games with similar mechanics. This question touches two different themes. The first important theme focuses on the preference makeup of the collective testing group. The results show a fairly diverse group, with the largest game preference only receiving 1/3 of the total vote.

The second important theme that this question addresses actually relates to the prior HL2 experience question. As the figure shows, 16% of testers prefer Shooters over other genres. To reiterate, the research artifact level borrows heavily from established shooter conventions, such as enemy behaviors, visual cues, and the user interface while eschewing conventional mechanics from other genres. Therefore, only 16% of testers preferred games that use those conventional gameplay mechanics.

The "Favorite Game Genre" graph shows that the breakdown of favorite genre remains fairly evenly spread across most of the options. Even the dominant preference (Action Adventure) only represents 1/3 of the total. The breakdown suggests that this thesis demonstrates how tutorials affect a broad range of gamers, not just those of a particular gaming persuasion.

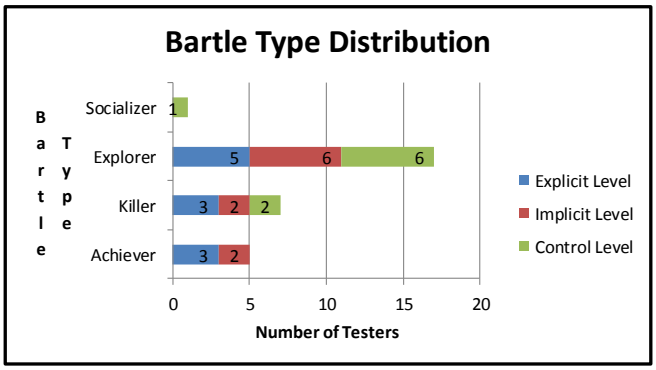

Figure 12: Bartle Distribution across Level Types

The variety of Bartle types distributed randomly across tutorial versions ensured that different play styles would not distort the results. In other words, the test evenly distributed players that typically take their time and those that attempt to finish as quickly as possible across the different tutorial versions of the level. As the figure shows, every Bartle type ended up evenly distributed across the different versions of the tutorial level. The Achiever group represents the only exception. By chance, Achiever distribution landed evenly between the Explicit level (3 testers) and the Implicit level (2 testers). No Achievers ended up testing the Control level.

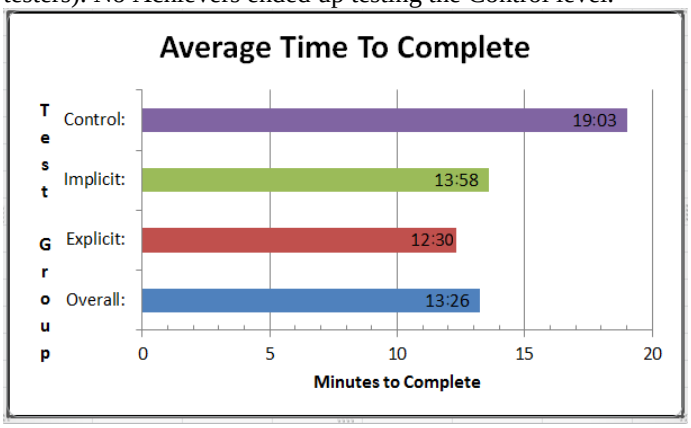

Figure 13: Average Completion Time

One of the key experiment observations recorded by the test proctor focused on whether players from the Implicit or Control group could complete the level as quickly as the Explicit group. To prove the hypothesis, the test proctor timed each level playthrough. The average time to complete the level for all participants took 13 minutes and 26 seconds. The Explicit group completed the test the quickest (12 minutes and 30 seconds) despite the extra instruction and lock-stepping. The Implicit group completed the level close to the overall average time for all participants (13 minutes and 58 seconds) and the single Control tester than finished the level took much longer (averaging 19 minutes and 3 seconds). In order to statistically analyze the difference between the Explicit and Implict average times, the thesis utilized a two-sample T-test assuming unequal variance. Using the equation (t(df) = t Stat, p = P(T=t) two-tail) provides the result, t(14) = -.8679, p = .4. Thus, the gap between the Explicit and Implicit average times is not significantly different. The percentage of testers that finished each level varies by group. 91% of the Explicit testing group finished, 70% of the Implicit group completed the test, and only 10% of Control testers completed the research artifact. All Explicit players reached the Office level area and all Implicit testers except one reached the Office. The tester that did not reach the Office stopped midway through the Alley. The Control group regularly dropped off throughout both the Alley and Office sections, but all Control testers passed both the Warehouse and the Labs. The average completion time only averages those testers that successfully completed the level. The Conclusion below elaborates on the importance of these results.

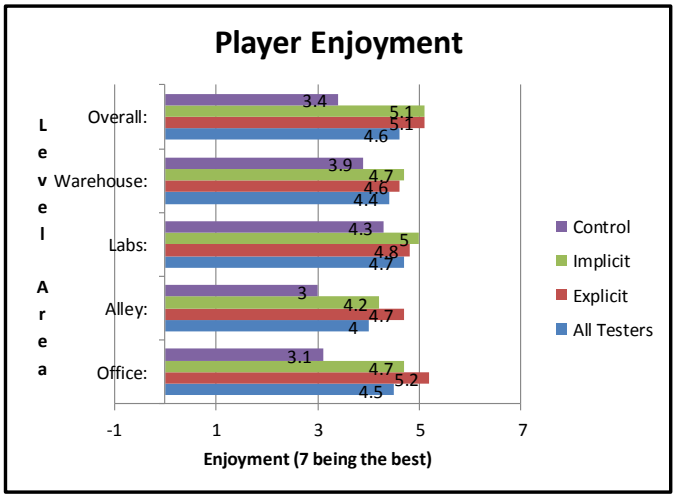

Figure 14: Player Enjoyment

Another critical feature of the research rested on how much players enjoyed each level version. The original hypothesis stated that players would prefer the Implicit level version over both the Explicit and the Control versions. The results suggest that players do not have a tutorial preference during relatively easy sections (such as the Warehouse and Labs) while they slightly preferred the Explicit version of the tutorial during the more abstract sections (the Alley and the Office). The Warehouse section received an average Implicit score of 4.7 (out of 7 possible) and an average Explicit score of 4.6. The Labs earned an Implicit score of 5 to the Explicit 4.6. The Alley and Office (more difficult gameplay) flipped the popularity of the tutorial techniques. The Alley garnered a 4.2 average from Implicit testers as compared to the average 4.7 from Explicit testers for the same section. The Office earned an Implicit 4.7 and an Explicit average of 5.2. Due to the slight difference in preference, the early sections earned effectively the same score. The two abstract sections, the Alley and the Office, created a wider gap between the Explicit and Implicit groups. Yet, neither the standard nor abstract gameplay proved significantly different. Using a two-sample T-test assuming unequal variance with the formula, (t(df) = t Stat, p = P(T=t) two-tail), the overall outcome for the Warehouse and Lab section resulted in t(33) = -0.406, p = 0.687. The Alley and Office yielded a result of t(39) = 1.138, p = .2619. Neither of those values proves statistically significant preference. When averaging every level area together, the research demonstrates players did not prefer either tutorial version over the other. However, the results of the experiment show that players preferred both the Explicit and Implicit versions over the Control in every situation. The overall player average of both the Explicit and Implicit versions came in at 5.1 points out of a possible 7. However, the Control group averaged 3.4 out of 7. The Conclusion section below offers further evaluation of the data.

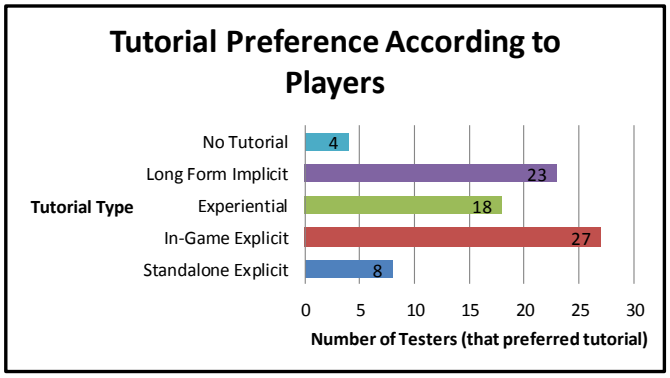

Figure 15: Tutorial Preference According to Players

To account for player-stated preference, the test included a question that directly asked players what types of tutorials they enjoy. The possible responses were phrased to be as specific as possible (more specific than the focus of this thesis). In addition, each tutorial format included a comparison to a popular game to aid the player in understanding the choice (such as "Long Form Implicit- i.e. Portal/Limbo"). The tester could select as many of the five options as they liked. To relate the results to the terminology used in this thesis, No Tutorial describes the Control group, Long Form Implicit and Experiential describe the Implicit group, and both Explicit responses relate to the Explicit group. Relatively few testers preferred pure non-gameplay tutorials (only 8 out of 31 testers) or the absence of a tutorial altogether (4 out of 31). Most testers preferred a combination of Implicit and Explicit tutorial formats. As the chart shows, all three remaining formats received votes from a majority of testers (at least 18 out of 31). The analysis of tutorial preference continues in the Conclusion.

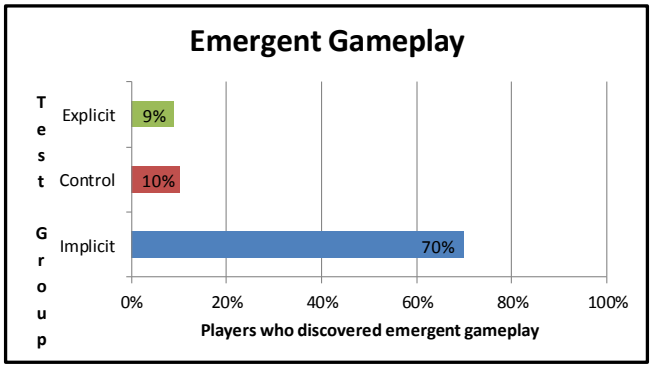

Figure 16: Emergent Gameplay

One of the more profound results came from analyzing emergent gameplay. Emergent gameplay describes players discovering new solutions to the level challenges not originally intended by the designer. The test proctor tracked this data by recording any unknown solutions to a level challenge as well as recording every time a player backtracked to find the hidden Rocket Launcher pickup. While there was no initial expectation to find emergent gameplay data, the hidden pickup served as a catchall in the unexpected case of interesting results. The tutorials do not teach the player about the Rocket Launcher or hint at its location. Therefore, when players venture off the critical path and find the pickup they are creating a unique solution to their current challenge,

simulating emergent gameplay. In both the Explicit and Control groups roughly 10% of players completed at least one level area in an unexpected manner. Surprisingly, 70% of the Implicit group discovered an unknown solution to complete the level, such as using boxes for cover or running past turrets before they could activate and target the player. The Conclusion section below further discusses these findings.

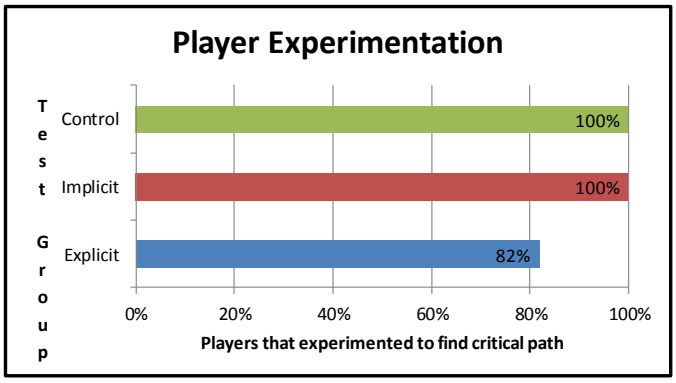

Figure 17: Player Experimentation

The final interesting result that emerged from the research revolves around player experimentation. During the post-test questionnaire, testers listed how they learned the solutions to the level challenges, such as "Instruction" or "Trial and Error." The purpose of this question focused on capturing the tester's thought process when completing each level area to better understand how they reacted to the tutorial instruction or lack thereof. When asked to list how they completed the level, almost all players responded that in addition to other methods (such as following instruction) they experimented to find the solution. As the figure shows, almost every player said that at one point or another they experimented to progress through the area.

Following the question on how players learned level mechanics the post-test questionnaire asked players if they were frustrated at any point throughout the level. 28 out of the 31 total testers responded that they were indeed frustrated at some point during the test. Frustration indicates that players were unsure of at least one part of the level, not that they were frustrated by the level as a whole. The area breakdown of where the frustration occurred tells us more of what frustrated players. When analyzing all testing groups, 32% of players were frustrated during the Warehouse, 32% were frustrated during the Labs, 63% were frustrated during the Alley, and 88% of players were frustrated during the Office.

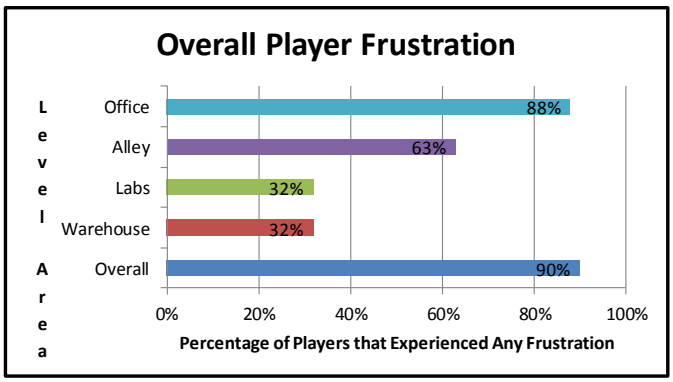

Figure 18: Overall Player Frustration

The frustration curve corresponds to the increasing difficulty of the level. As the level gets harder from the Warehouse through to the Office, the level of frustration goes up. Lack of clear tutorial information or level errors can also generate frustration. However, when compiling the testing results, it became clear that tutorial mechanics were not primarily responsible for either Explicit tester frustration or Implicit tester frustration. Lack of communication stemming from poor tutorial instruction, difficulty, confusing sounds, or level errors such as running out of ammunition were all equally stated reasons for frustration. The only time when the Implicit group recorded significantly higher player frustration than the Explicit group occurred during the Labs section. 50% of Implicit players said they were frustrated at one point during the labs as compared to only 9% of the Explicit group. The use of gunfire to direct the Implicit players to a scene using NPC Foreshadowing most likely contributed to the increased frustration. When compiling the data from all testers, 90% of the test pool expressed frustration at some point during the test. The group that did not experience frustration at any point during the test comprises two Explicit testers and one Implicit tester. The Conclusion below elaborates on the importance of player experimentation and its connection to player frustration.

V. Conclusions

An examination of player enjoyment and learning yielded surprising and important results. On one hand, the expected results of the thesis revolved around the idea that players prefer Implicit tutorials over Explicit tutorials. Tester reaction indicates that both tutorials have their place and can be effective teaching tools. The difference between the two tutorial forms goes deeper than player preference and completion time. Specifically, unexpected player behavior suggests that Implicit tutorials have a profound impact on player creativity and experimentation. In addition, Explicit tutorials seem to teach the player the required mechanics with a higher success rate than Implicit tutorials. These results aid game developers in deciding how they structure the tutorials in their games, as the right application of a tutorial type could create a profoundly different game experience.

The thesis hypothesis turned out to be largely inaccurate. The expectation focused on players completing the Implicit tutorial version as fast as the Explicit group while having more fun playing the Implicit. Testers rated both the Explicit and Implicit tutorial versions of the level an average of 5.1 out of 7 possible points. Testers did seem to slightly prefer the less controlling Implicit tutorial while learning basic mechanics while they preferred learning from the Explicit tutorial while figuring out more abstract concepts, such as stealth. In fact, results suggest that players may have more fun learning abstract gameplay from Explicit techniques, since it delivers critical information that would otherwise take the player some time to discover on their own. However, the difference in average score did not seem drastic in any section and suggests more of a slight change in tutorial effectiveness per level area over clear player preference for one type of tutorial.

As for the average time to complete the level, players completed the Explicit version about a minute and a half

faster. Explicit testers finished the test at 12 minutes and 30 seconds and the Implicit group completed the level at 13 minutes and 58 seconds. The time discrepancy failed to prove significant. The results imply that Explicit communication delivers enough information quickly that the pause in gameplay still remains shorter than the time it takes Implicit testers to figure out the section. Unfortunately, the lack of a full data set distorted the average time for the Control and Implicit groups. If it became clear to the Test proctor that players were stuck in a frustrating gameplay loop and could not figure out the mechanic then he stopped the test. Therefore, because 30% of the Implicit group and 90% of the Control group did not finish the level, the time estimates are only approximate. For reference, 9% of the Explicit group did not finish the level. It seems as though there exists a higher player retention rate when using Explicit tutorial instruction.

The difference between 91% of the Explicit testers finishing the level and 70% of the Implicit testers finishing the level comes down to two factors. First, the Implicit tutorial implementation was less successful than the Explicit tutorial implementation. The higher failure rate for Implicit tutorials most likely comes down to weaker implementation of Implicit tutorials for specific level areas, such as the Alley and the Office. The Implicit version of the Alley did not indicate that the player must crouch and sneak by the first Turret to enough testers. Implicit techniques did not adequately teach that the player must remain crouched in all hallways throughout the Office, either. Greater environmental storytelling and better use of NPC Foreshadowing may have lowered the failure rate of the Implicit tutorials. The inability to get adaptive messaging working for the Implicit group also proved damaging to the completion rate of the tester group. Second, the principle behind Explicit tutorial techniques focuses on providing the player with a guided walkthrough. While players participate in the game mechanics, the experience stays mostly passive. Testers receive instruction and follow it exactly. Implicit tutorial instruction relies on player perception and creativity. When players encounter an Implicit tutorial they must teach themselves the correct mechanics. That self-teaching puts the player in an active state the entire time. The different mentalities for Explicit and Implicit tutorials likely affected the average times as well, since the two tutorial styles naturally seek different outcomes.

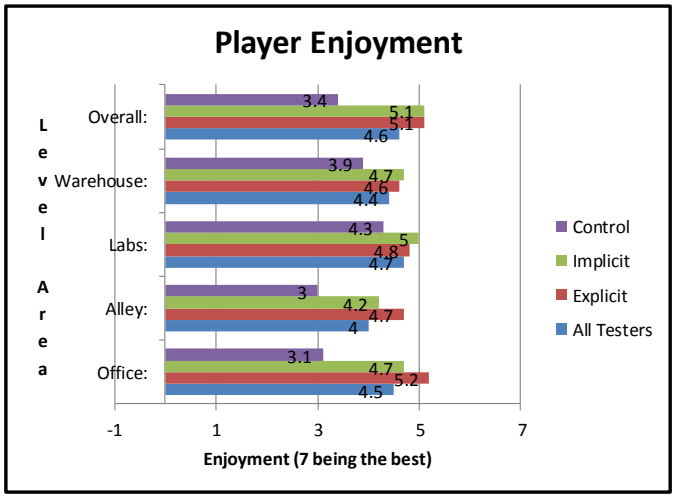

Figure 19: Player Enjoyment

The second component of the hypothesis also proved inaccurate. Players did not prefer one type of tutorial over the other. Two different methods focused on answering this question. First, the testers described their level of enjoyment for each level area and the level overall. Comparing this information between each tutorial group (Explicit, Implicit, and Control) showed that people generally preferred versions of the level with tutorial instruction over no tutorial at all (5.1 versus 3.4 out of 7). However, the hypothesis stated incorrectly that people would prefer Implicit instruction over Explicit instruction. Overall, players enjoyed both versions of the tutorial the exact same (5.1/7). The data seems to suggest that players prefer Explicit instruction when learning more abstract gameplay concepts such as stealth and do not have a tutorial preference when learning simple concepts such as movement and healing. However, no significant difference emerged when analyzing the data using T-tests. A larger testing pool could identify statistically significant preferences. Players rated the Explicit version of the abstract sections .5 higher than the Implicit version (4.7 Explicit compared to 4.2 Implicit for the Alley and 5.2 Explicit compared to 4.7 Implicit for the Office). However, the difference remains slight, especially for the small testing pool of 31 players.

It should be noted at this point that the design behind the research artifact focused on comparing all common Explicit tutorial mechanics with common Implicit tutorial mechanics. To ensure that the player learned, the research artifact designer constantly used multiple tutorial mechanics of the same persuasion within the same section. Mechanics of the same persuasion refers to two or more tutorial techniques that belong to the same tutorial type, such as how both Breadcrumbs and Lock-Stepping classify as Explicit tutorial techniques. That made it impossible to identify the merits of individual mechanics, although it did seem to result in more consistent teaching throughout each level version. Initial feedback from pre-test playthroughs indicated that using a single tutorial technique per level area did not adequately teach the mechanics. If players did not respond to that tutorial technique, then they effectively became Control testers. Thus, each level area uses multiple tutorial techniques to ensure that the player learns.

The more interesting results of this thesis emerged from analyzing player behaviors during the tests. The original intention behind monitoring player behavior focused on capturing atypical instances of tutorial effect on player behavior. It became clear over the course of the research that different tutorials encouraged different or surprising play styles.

When analyzing player experimentation it became clear that 82% of the Explicit group still experimented while playing to learn the level mechanics. That experimentation refers to players that learned the necessary mechanics without tutorial help. The Explicit group received the required information to learn every level mechanic, so experimentation should not have been necessary. Interestingly, because tutorial clarity did not emerge as the dominant source of frustration, the fact that 82% of the Explicit group still experimented to discover gameplay suggests that many players ignore, forget, or do not understand the instruction. Explicit tutorial information always takes up the middle of the screen, which puts in directly in the player's field of view. Players would have to experiment to discover the information if they ignored the Explicit tutorial or did not internalize the information. If they had attempted to follow the Explicit information and failed, it would be likely that tester frustration would not only be a much higher percentage, but lack of communication would be cited as the dominant source of frustration. For reference, the sources of frustration include level difficulty, level errors, sounds (in particular the machine gun noise from the Implicit version), unusual controls, an incorrect expectation of *Half-Life* gameplay, no checkpoints (Explicit and Control versions), and a lack of communication (ineffective tutorial).

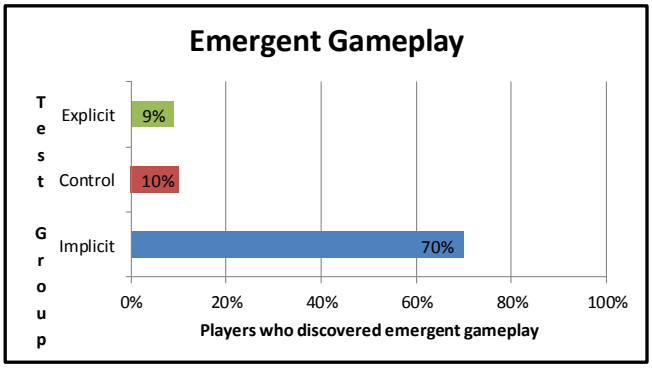

Figure 20: Emergent Gameplay

The emergent gameplay data proved to be even more profound. 60% more players experimented to find unknown solutions to the level challenges within the Implicit tutorial than the Explicit or Control tutorials. Such stark contrast suggests an actual change in player behavior when provided just enough information to get the player going without giving away the entire solution. The test proctor observed that Implicit testers displayed a willingness to test alternate theories, even after the tester discovered the first successful solution. The Implicit testers could very well have taken longer to complete the level simply because they were taking the time to explore alternate solutions. By contrast, Explicit testers seemed predominately interested in completing the most recently explained mechanic as quickly as possible, while Control testers seemed anxious about the possibility of dying and losing their progress. Once Control testers died and restarted, they seemed interested in getting back to where they were as quickly as possible instead of taking the time to explore alternate options. A cause of the behavioral difference probably stems from the different intentions of each tutorial type. After all, Implicit tutorials encourage players to teach themselves. Explicit tutorials teach the player how to play as fast as possible. Those inherently different goals could create a natural difference in how quickly players complete the tutorial, depending on their tutorial type.

Results suggest that developers could use Implicit tutorials as a tool to enhance player creativity when desired. Emergent gameplay became a surprise critical factor in the data. Thoroughly testing and refining the level would have reduced unintended shortcuts. An analysis of the level construction and subsequent gameplay suggests that the unrefined nature of the level enabled players to express their creativity by accidentally leaving multiple avenues open for testers to complete each challenge.

Throughout the process of creating the research artifact and compiling the data, it became clear that there were several limitations with the thesis. The low sample size (31 total people) emerged as the largest problem, particularly since a number of testers did not complete the level. Further study requires significantly higher numbers of testers to prove any of the results. In addition, the selection of Source SDK created a challenge when creating the level area for the Office. In order to properly convey that the player needed to crouch it would have been much better if the engine supported higher numbers of dynamic lights so that the player could see the sentries scanning the walls above a certain height. This lack of clarity seemed to frustrate many players. Certain tutorial techniques became problematic to implement and thus provided less helpful tutorial information within the level, such as the Implicit Necessity of Action in the Office. Players frequently expressed confusion over how the turrets reacted to their motion. Better environmental storytelling would have clarified the need for the player to remain crouched at all times.

A common result from the open feedback question at the end suggested that the more difficult sections of the Implicit tutorial (during the Alley and Office) were not comprehensible to the player. In retrospect, using a game engine that fully supported dynamic lights would have dramatically improved the Implicit element of Positive Reinforcement. Even more so, it remained exceptionally difficult for the thesis designer to implement Adaptive Messaging throughout the project and as such the designer cut that Implicit technique from the level due to it being error-prone. Judging from the written feedback and drop-off in player enjoyment during the more difficult stages of the tutorial, Adaptive Messaging would have been a perfect mechanic for both the Alley and the Office. Future research on this topic should strive to include that functionality.

Also, the decision to use multiple techniques per level area made it impossible to analyze the effectiveness of individual tutorial techniques. Instead, the thesis could only compare Explicit tutorial theory with Implicit tutorial theory. Finally, the most surprising results came from the psychological side of the player experience. Emergent gameplay stood out as a

major consequence of the different tutorial types, and had a psychologist been involved in the research from the beginning the thesis would properly analyze the results. As such, the research results only indicate a connection between player behavior and tutorial type.

As a recommendation for future research, this thesis suggests further examination of the relationship between player creativity and tutorial instruction with an emphasis on what type or frequency of instruction modifies player behavior. The significant difference between Implicit emergent gameplay and that of the Explicit and Control groups implies that even though the level challenges remain the same, players approach the challenges differently depending on the amount and type of information they receive. As previously stated, additional research should also incorporate Adaptive Messaging as an Implicit tutorial mechanic. Adaptive Messaging could serve as the bridge between the experience Implicit-minded players want with the eventual explicit information they may need to pass through a section. Player enjoyment results may have been very different if the research artifact had included Adaptive Messaging. This thesis took the position of comparing Explicit tutorial techniques with Implicit tutorial techniques as a whole. Further research should examine the impact and success of each technique separately. Games would benefit from an understanding of the right time and situation to use each individual technique. Finally, future research should strive to create a testing environment not based on a previously released game. A short schedule and limited resources necessitated building the research artifact as a mod of an existing game engine, but that created significant risk of experienced players ignoring new tutorial instruction.

In conclusion, the limited information currently available on the subject of player enjoyment and learning with Explicit versus Implicit tutorials suggests that the game industry should further explore the impact of both Explicit and Implicit tutorials. The results from this thesis suggest that players find both tutorial types acceptable and equally helpful when used in the right context. Implicit tutorials reduce player frustration while learning basic concepts while player prefer Explicit tutorial techniques to help teach difficult gameplay concepts. More importantly, the thesis uncovered a possible relationship between Implicit tutorials encouraging player creativity. This thesis aided in directing that research towards player preference and completion time, but resulted in asking more questions about emergent player behavior and the effectiveness of individual tutorial techniques. Results from both direct questioning and player feedback suggest that players do not seem to have a clear preference for tutorial types so long as the tutorials do not remove the player from gameplay for too long. In addition, players complete Explicit tutorials faster than other tutorial forms but do not create nearly as much emergent gameplay. Of all the results, the connection between Implicit tutorials and emergent gameplay stands out as both the most unexpected and the most significant. Games that focus on player creativity and cleverness should explore the potential of Implicit tutorials to generate dependable player creativity. If developers can replicate the connection between implicit instruction and emergent gameplay then they could better direct the player experience, and ideally create more successful games that rely on player creativity.

VI. References

Repository: //ghsrv/C17/MITFolders/C17/CampbellCrapsey

Aki Jarvinen. “First Five Minutes: How Tutorials Make or Break Your Social Game.” *Gamasutra.com,* April, 2010. October 16, 2012.

Arc System Works . *Persona 4 Arena.* Atlus, 2012. XBOX 360.

Carlsen, Jeppe. “Limbo: Balancing Fun and Frustration in Puzzle Design.” Playdead. Game Developers Conference Europe. Cologne, Germany. August 17, 2010. Summit Address.

Christopher Paul. “Optimizing Play: How Theorycraft Changes Gameplay and Design.” *GameStudies.org,* May 2011. October 15, 2012.

EA Canada. *Fifa Soccer 13.* Electronic Arts, 2013 . XBOX 360.

Fan, George. “10 Tutorial Tips from Plants vs. Zombies.” PopCap. Game Developers Conference. San Francisco, CA. February 28th, 2012. Summit Address.

Gaynor, Steve. “Techniques for In-Level Storytelling.” The Fullbright Company. Game Developers Conference. San Francisco, CA. March 22nd, 2013. Tutorial Address.

Jones-Rodway, Neil. Sun, Aldric. “Teaching Players: Tutorial and Opening Mission Design for COMPANY OF HEROES.” Relic Entertainment. Game Developers Conference. San Francisco, CA. February 28th, 2008. Summit Address.

PlayDead. *Limbo.* PlayDead, 2010. PC.

Riot Games. *League of Legends.* Riot Games, 2009. PC.

ThatGameCompany. *Journey.* Sony Computer Entertainment, 2012. PlayStation 3.

Valve. *Portal 2.* Valve, 2011. PC.

Valve. *Half-Life 2.* Sierra Entertainment, 2004. PC.

Valve. *Half-Life 2: Episode 2.* Valve, 2007. PC.

VII. Appendix

Crapsey_BartleTypePreference.xlsx
Crapsey_FavoriteGenre.xlsx
Crapsey_FPS-HL2PriorExperience.xlsx
Crapsey_PlayerEnjoyment.xlsx
Crapsey_PlayerFrustration.xlsx
Crapsey_PlayerStatedTutPreference.xlsx
Crapsey_RocketLauncher-EmergentGameplay.xlsx
Crapsey_SurveyResponses.xlsx
Crapsey_Thesis_Abstract.docx
Crapsey_ThesisTestingTracker.xlsx
Crapsey_TimeToComplete.xlsx

Campbell Crapsey. Plano, Texas. April 25, 1988. Earned a Bachelor of Arts in French at the University of Wisconsin: Madison. He is currently finishing his Master of Interactive Technology in Level Design at the Guildhall at Southern Methodist University in Plano, Texas.