Using Environmental Illusions to Immerse Players in a Game World

Hammed S. Al-Tamimi, Boris Fisher, and Elivorio Luna

***Abstract*—This paper covers the creation of an artifact, and tests various environmental illusions on their effectiveness of creating an immersive game world, while reducing the workload on developers. The tester created an artifact in the Unreal Development Kit that provides examples of environmental illusions. The tester gathered and compiled data to determine the effectiveness of the illusions.**

***Index Terms*—environment art, environmental illusions, interactive, video games**

I. Introduction

Environment artists in the game industry create the props and assets placed in the virtual worlds . They must help contribute to the composition of a game environment, determine the lighting of the world, and create an aesthetically unique but coherent visual space. [1] Environment artists must create highly polished and immersive assets in order to properly convince a player that he or she is in a cohesive world. One of the many methods of approach an artist can take involves using environmental illusions.

The purpose of this thesis is to identify and construct environmental illusions which create an immersive gameplay experience. The thesis presents a possible solution to creating an exciting and novel world that is enjoyable to the player without using a large amount of production resources. The accompanying artifact displays the usage of different environmental illusions.

First, the thesis research takes into consideration different types of illusions. Later in the process, the tester constructed an artifact demonstrating examples of environmental illusions.

II. Research Review

This study takes into consideration two types of real-world illusions, and provides examples of the usage of illusions in various forms of media. It then examines existing environmental illusions in video games.

A. Object Scale Illusions

Two illusions that hide the scale of an object are the Ponzo Illusion, and the Moon Illusion. The Ponzo Illusion takes place when two lines of equal length appear to be unequal when juxtaposed against their surroundings [2]. The Moon Illusion operates on the same basic principles. When the moon takes a position in the sky closer to the ground, it appears larger to an individual than when it occupies a space higher in the sky [3]. The tilt constancy theory states that these optical tricks occur when a person incorrectly perceives orientation through visual cues [2]. When illusions occur on larger scales, such as entire rooms, it amplifies the effect [2].

Forced perspective also serves as another example of an illusion that masks the true scale of an object. According to the Digital Photography School website, forced perspective involves creating the illusion that objects are bigger, smaller, closer, or more distant than they exist in reality. Photographers and filmmakers often use the technique as a way of making small props look gigantic. [4]

B. Anomalous Motion Illusions

Professor Akiyoshi Kitaoka from the Department of Psychology at Ritsumeikan Univerity in Kyoto, Japan, does research on anomalous motion illusions. On his website, he shows several examples of static images that appear to move when a person looks at them. In his “Rotating Snakes” work, several interlocking circles seem to rotate when the viewer moves his or her eyes. [5] In Kitaoka’s “Primrose’s Field”, he provides an example of a waving motion illusion. It consists of a checkered image with small flowers. Just like the other image, a person viewing the illusion sees motion when he or she moves his or her eyes. [5]

C. Existing Environmental Illusions in Video Games

The game Mass Effect hides level streaming by loading and unloading asset while the player stands in an elevator. This creates the illusion of a seamless world. [6]

Manuscript received May 9th, 2014.

Hammed S. Al-Tamimi is currently a graduate student at the Guildhall at Southern Methodist University, Plano, TX 75024 USA. (e-mail: haltamimi@smu.edu).

Boris Fisher is currently an Art Creation faculty member at the Guildhall at Southern Methodist University, Plano, TX 75024 USA. (e-mail: bjfisher@smu.edu).

Elivorio Luna is currently an adjunct professor at the Guildhall at Southern Methodist University, Plano, TX 75024 USA. (e-mail: ezluna@smu.edu).

In Diablo III, the player starts out Act II in an area called the Hidden Camp, where the player sees a city in the cliffs below [7]. Blizzard Entertainment created an illusion of a large world with depth using a combination of matte painting and image planes. Developers know this technique as parallaxing.
The game, Super Mario 64, takes advantage of forced perspective in the hallway leading to the Tiny-Huge Island level. The player sees three paintings hanging on the wall, which appear to be the same size and distance from the center of the hallway. When a person approaches one of the paintings, it becomes clear that it is a lot larger than the others. The painting on the opposite side of the room is much smaller than the other two. [8]

Saints Row IV provides an example of using material shaders to create the illusion of depth. When players look into shop windows, it looks as though there are interiors inside of the buildings, and the perspective changes with the player's location. The illusion fails when the player stands at the corner of the building and faces parallel to the wall. [9]

III. METHODOLOGY

A. *Introduction*

In order to test the effectiveness of the three illusion types at enhancing a person's gameplay experience, the tester constructed a virtual environment. The tester then analyzed the illusions after construction, taking into account ease of execution, resources used, and success of the illusions.

B. *Product and Development Process*

The development of the artifact involved using multiple software tools. The tester created the assets using 3ds Max, Adobe Photoshop, Pixologic ZBrush, Autodesk Mudbox, and XNormal. The tester constructed the artifact for the study using the Unreal Development Kit (UDK). The tester scripted events in UDK using Kismet and Matinee.

The artifact itself exists in the form of a space station. The first illusion occurs when a player looks out of a window, and observes a large planet and its moon.

Figure 1: The moon illusion in game

In order to achieve this illusion, the tester used nearly flat disks in the form of image planes placed behind the windows of the ship. The illusion succeeded because the player was only able to see the planet and moon from a fixed perspective. This saves on resources, since the artist only needs to model and texture the geometry visible to the player. When an object obstructs the visual field enough, the ability for an individual to judge distance begins to fail.

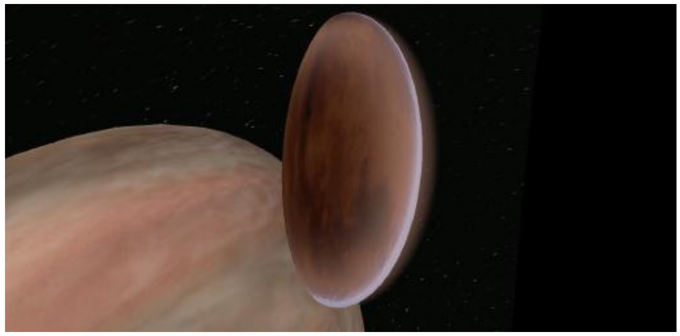

Figure 2: The illusion viewed from the side, demonstrating the disk nature of the moons

The second illusion involves creating fake holograms with geometry and shaders. The success of the illusion depends on efficiency in shader nodes, because an overly complex network causes frame rate drops.

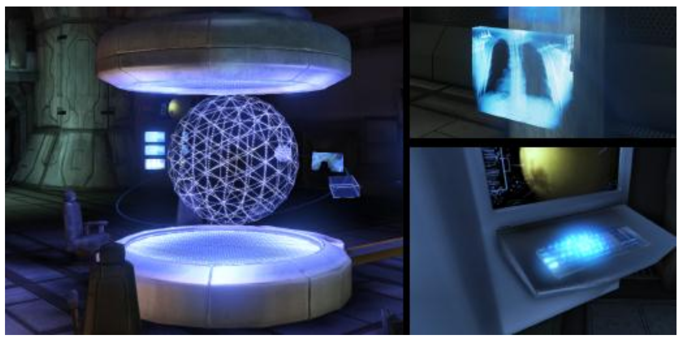

Figure 3: examples of holograms in the artifact

In order to successfully create the feeling of a hologram, the tester primarily depended on the emissive and opacity channels of the material. By blending multiple materials, the tester created visual interest, and slight motion in the holograms. A scrolling dithering effect made the holograms feel as though they were not solid objects. The tester highlighted more important areas of the hologram by applying a masked scrolling highlight to the emissive channel. On the projection of the planet, the grid lines popped out due to this effect.

The next illusion involves creating a fake hallway extension. Instead of modeling out the unused gameplay areas, the tester created material shaders similar to the ones used on shop windows in Saints Row IV as described in the Research Review section. The tester took a screenshot of a hallway in the game, and modified it in Photoshop to create the image used in the illusion in order to save time.

Figure 4: the hallway illusion viewed at a distance

Creation of the fake hallway extension was a process which the tester find quick, and easy to create. The tester took a screenshot of an existing hallway in the game, and applied image filters in Photoshop in order to create atmospheric effects. In UDK, the tester assigned the image to both the diffuse and emissive channels of the material. He then applied a bump offset modifier to the UV channel. The bump offset modifier distorts the image based on the viewer's perspective angle, creating the illusion of depth. The tester applied the material to a mesh which did not accept scene lighting or shadows.

The fourth illusion occurs in the next area of the space station, the laboratory. In the lab, various specimen tubes look as though they contain an aqueous solution. The tester created this illusion using a distortion channel in a material, and a flat bubble texture placed on a particle system. A particle system is essentially a controlled system of points in 3D space which have dynamic rules that define their behavior. Game engines attach images to the particles. They are advantageous over actual geometry since they require less system memory, and are more dynamic. [10]

Figure 5: the specimen tubes within the lab

In the fifth illusion, two large spinning spirals surround two bridges. On the first bridge, the spiral spins in one direction, and on the second bridge, the spiral spins in the opposite direction. Even though the tester creates both bridges at the same length, the spirals, if successful, make one of the bridges appear to be longer than the other.

The tester created this illusion by importing spiral geometry into the engine, and then converting it into an InterpActor. This changed the geometry into a dynamic object. The tester then applied a predefined physics state of rotational to the spiral. When played in the editor, the spiral spins along its center axis and creates a feeling of either forward or backward movement depending on which perspective the player views it from.

Figure 6: the spiral illusion surrounding a bridge

The next illusion demonstrates a method designers and artists can use to hide loading times using fake elevators. The elevator uses scrolling textures to simulate movement between multiple floors. Artists and designers in the industry often hide level streaming from the player using this technique.

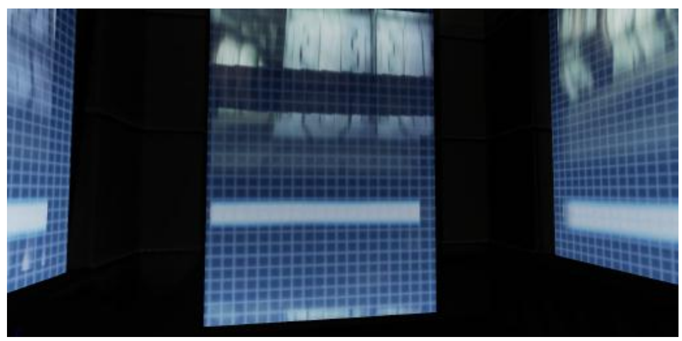

Figure 7: the elevator illusion from the player's perspective

When viewed externally, the construction of the illusion becomes more apparent. Both the interior and the fake glass of the elevator remain static. Scrolling images directly outside of the windows quickly pan vertically.

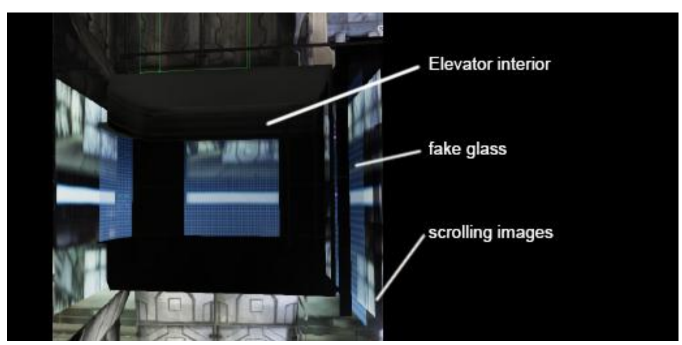

Figure 8: the elevator illusion from an external perspective

Illusion number seven involves faked heat distortion near a burst pipe. Using a blotted texture panning across a translucent material, the illusion should make the area near the pipe seem hot to a player. Although efficient in resource usage, this effect requires a high quality execution of the art in order to be believable.

The eighth illusion uses floating props and camera effects to create the feeling of a vacuum in space with low gravity. The illusion exists as a shorthand way for communication of areas which a player can jump higher.

Figure 9: Floating object hint at low gravity

Next, the tester created an airlock room as another alternative to loading and unloading areas into game memory with minimal disruption of gameplay. The picture below shows the layout of the room from a player's perspective.

Figure 10: the airlock room pressurizing

The tester uses particles for a pressurization effect. The objective of the illusion is twofold. The first is to create a sense of an increase in air pressure, and the second is to hide loading times and level streaming with a temporarily inescapable room, as demonstrated by the diagram below.

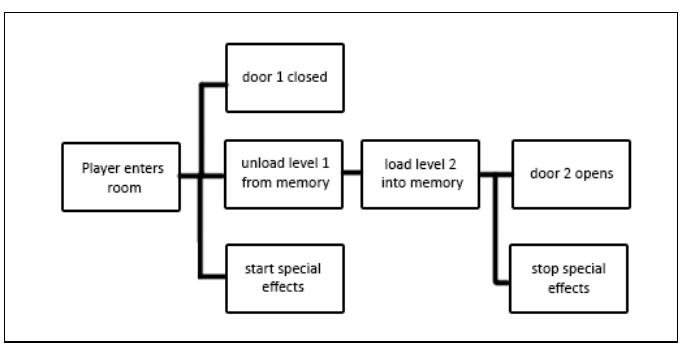

Table 1: the events of the airlock room

For the final illusion, the tester constructed another example of hiding loading times using scrolling materials, this time in a horizontal instead of vertical scroll direction. The player enters a moving tram in a short tunnel.

The artifact's assembly occurred in different stages.

Stage one: Proof of concept- The first stage involved creating a demonstration of the technologies and techniques used in the artifact. It also takes into consideration of the time and resources each technique takes.

Stage two: Whitebox- This step involved the completion of basic placeholder models in the artifact.

Stage three: Vertical Slice- All assets have base textures, and placed by the experimenter in the editor by the end of this stage.

Stage Four: Alpha- The artifact has basic functionality, and nearly completed assets.

Stage Five: Beta- All assets polished, and any outstanding problems with functionality are resolved.

Stage Six: Completion- All assets and functionality completed, and the artifact is ready for review and analysis by the end of this phase.

C. Data collection and Procedures

Data collected by the tester included the tri count, vertex count, and file size of the geometry used in illusions. The tester checked performance by running the artifact on an Alienware M18x laptop with an Intel Core i7-3610QM CPU @2.30GHz 2.30GHz, 24.0 GB of RAM, an Nvidia GeForce GTX 680M video card, and with Windows 7 64-bit as the operating system.

The tester collected data on the amount of time he took creating each illusion, and the differences in file size between methods.

D. Summary

The construction of the artifact provides data on the resources required and success of various environmental illusions. The tester constructed the artifact using digital tools such as 3ds Max, and UDK. Afterwards, the tester collected performance data on an Alienware M18x laptop.

IV. RESULTS

Based on the process of creating the artifact, the tester discovered various methods for creating environmental illusions.

The moon illusion was one of the most effective illusions, especially when viewed from a fixed perspective. The tester compared using a full sphere, partial sphere, and flat image plane based on performance and effectiveness of the illusion.

The full sphere had the highest tri count, vertex count, and file size. The partial sphere did not have geometry on the sides which the player never sees, costing the engine less resources. The image plane technique was the least resource intensive method. It had two disadvantages over the other two techniques, however. The texture for the partial sphere requires an alpha channel for masking off transparency, and does not hold up from as many angles as the partial sphere. One method for viewing a flat image plane from multiple angles is to constrain its orientation to the player camera.

The table below lists the creation of a full sphere, partial sphere, and image plane for comparison.

Technique	Tri count	Vertex count	file size (.fbx)	Texture requires alpha channel
Full sphere	3024	1514	91.6kb	No
Partial sphere	1860	963	71.2kb	No
Image plane	2	4	15.4kb	Yes

Table 2: comparison of different methods for creating a moon

In the thesis artifact, the tester decided to use the partial sphere method, since it allowed for more accurate vertex lighting than the image plane, and held up better at multiple angles. The tester viewed the illusion's integrity at different camera field of view values, and determined the effectiveness of the illusion as demonstrated by the table below.

Field of view (in degrees)	Success of the illusion
45 º	Successful
60 º	Successful
75 º	Successful
90 º (default)	Successful

Table 3: the quality of the moon illusion based on camera Field of View

The moon illusion had a very consistent success rate. In the artifact, the tester found the moon illusion worked best when he placed the objects between 3100 and 15,300 Unreal units of distance from the player. He also found that the illusion works best when he juxtaposed more objects to the moons for scale reference, and when the moons consume at least a quarter of the screen space.

Figure 11: the moon illusion at a 45º field of view

Figure 12: the moon illusion at a 90º field of view

The tester found the hologram illusion one of the harder illusions to produce quickly at a believable level. The hologram appeared to be a solid object until the tester applied dithering and distortion effects to the material. He revised the illusion a total of six times.

The hallway illusion had mixed results. Although the illusion successfully created the feeling of a long hallway from far away, the illusion falls apart when approached closely from an extreme angle as illustrated below. In the artifact, the tester placed a slightly open door in front of the plane as a way to block the player from getting close enough to the illusion for it to fail. One issue the tester faced with the hallway illusion was keeping its integrity at different camera field of views. As demonstrated in the table below, the illusion failed at lower FOV levels. This occurred because the perspective lines of the image no longer matched the perspective of the real hallway.

Field of view (in degrees)	Success of the Illusion
45 º	Complete failure
60 º	Poor quality
75 º	Acceptable quality
90 º (default)	Optimal quality

Table 4: The quality of the hallway illusion at various fields of view

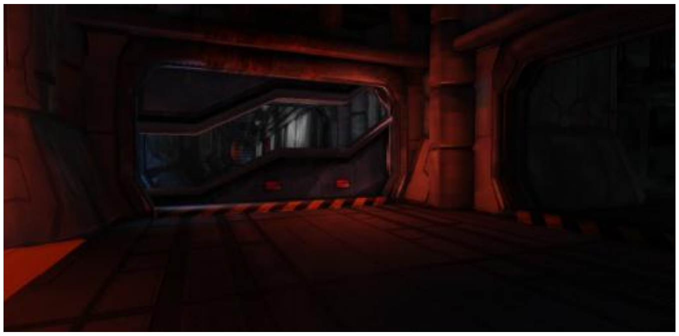

Figure 13: the hallway illusion at a 90º field of view

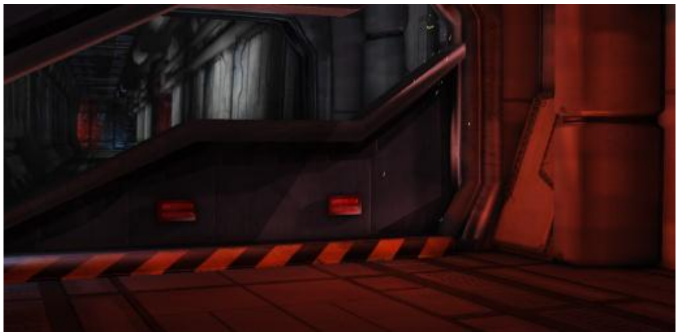

Figure 14: the hallway illusion at a 45º field of view

The tester found that in his scene, the hallway illusion functioned best at distances from the player between 300-2900 Unreal units. The geometry used in the effect had a width of 1024 Unreal units, and a height of 512 Unreal units. The image below demonstrates the illusion failing at a distance of 150 Unreal units with a 90º field of view.

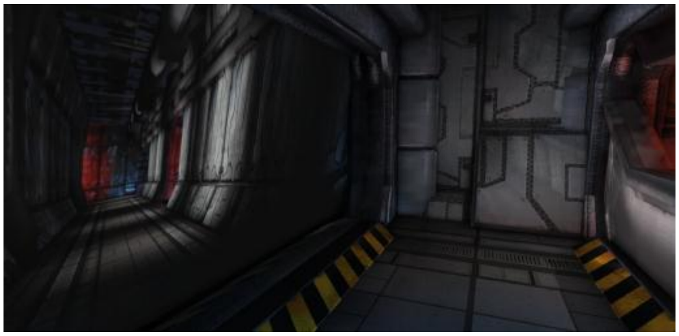

Figure 15: the hallway illusion fails when observed from extreme angles

The tester recommends using a hallway illusion instead of modeling out an entire hallway if the development team needs a time saving method, and if the player only ever views the illusion from one field of view. It took the tester twenty minutes to build a real hallway with similar aesthetic properties. On medium lighting settings with three lights, it took the tester four and a half minutes to build the lighting within the small hallway section. The hallway illusion does not require lighting, since it already exists in the image. Another advantage of using a fake hallway is an artist can paint effects such as depth of view blur, and bloom onto the image.

The fourth illusion with the water tubes was moderately successful, yet difficult to produce realistically. The tester modified the bubbles in the tubes multiple times before he was satisfied with the result. He discovered two ways of creating the bubbles in the tubes, both with their own advantages. The first method involves placing a texture with multiple bubbles on a flat image plane, and using a panner node in the material editor, so the bubbles moved vertically. The tester's second technique involved placing the bubbles on individual sprites, and rendering them using a particle system.

The tester observed no noticeable effect on the illusion's success based on camera field of view. The tester also found no observable effect on the illusion's success based on player distance.

Both methods are valid, but the tester preferred the look of the particle system method. It gave him more control on the dynamic movement of each individual bubble. At his current skill level, the tester spent more time creating the particle system version of the bubbles than the flat image plane version. The tester took 18 minutes setting up the particle system, and only 3 minutes setting up the image plane. He suggests that environment artists use the particle system method unless they are under heavy time constraints.

The spinning spiral illusion created a unique and disorienting area within the level. The tester created the spiral in 3DS Max using a spiral-shaped spline, and duplicated geometry along the curve. Afterwards, he welded vertices on the object, creating it into a single piece of geometry. This allowed the tester an easier time unwrapping the object for texturing.

The tester noticed that with both spirals, the greater the camera's field of view, the more intense the effect appeared on the screen. The tester based the success of the illusion based on the perceived motion of the player when standing still. The chart below demonstrates the success of the effect.

Field of view (in degrees)	Success of the illusion (spiral one)	Success of the illusion (spiral two)
45 º	Barely effective	Barely effective
60 º	Somewhat effective	Somewhat effective
75 º	effective	effective
90 º (default)	Very effective	Very effective

Table 5: the success of the spiral illusions at different fields of view

The tester recommends environment artists use the illusion sparingly, since the effect might cause simulator sickness due to the amount of motion it creates on the screen.

The elevator illusion was one of the easiest and least resource intensive illusions for the tester to create. The illusion only failed if viewed externally, but the tester prevented this by placing geometry to block undesired views of the illusion. The tester found no correlation between the field of view of the camera, and the success of the illusion.

It wasn't until the addition of stars in the background of the low gravity illusion that the tester felt confident in its success.

The addition of floating objects solidified the illusion. Since the low gravity illusion depends so much on visual cues, the tester suggests environment artists to either add rotation on debris floating in the space, or set up an interactive physics system for full simulation of solid objects in a low gravity area.

The pressurization effect took a lot of balancing of system resources and graphical fidelity, because the tester used particles to convey air flooding in the chamber. At first, the tester used too many particles, which caused a sudden frame rate drop in the test. The tester later used a lower number of particles with a higher resolution texture. The tester found the optimal balance of visual fidelity and performance on the Alienware M18x laptop with a distribution rate constant of 5.000 at 60 frames per second, while using a 256x256 .,png texture with alpha transparency. The tester chose the texture resolution, since the particles spawn within three meters of the player.

Some illusions, such as the hallway illusion, took some extra research by the tester, but still reduced the amount of system resources allocated towards the section of the artifact. The tester found some of the other illusions, such as the moon illusion, to be easy to create, and saved resources in the game engine. The tester believes that the hypothesis was mostly correct.

If the tester had more time to work on the individual illusions, he could improve them in several ways. The tester might animate the fake holograms, and give them a more dynamic feeling using material effects. The tester can fix the issues with the hallway illusion by swapping the textures so the perspective lines match when the field of view of the camera changes. In the elevator and the tram, the tester would have the speed of scrolling textures used in the illusions follow a sine wave. If implemented correctly, the observer would perceive the elevator and tram gradually speed up, and then slow down after reaching the peak of the sine wave.

Although these illusions work very well on computers and current generation consoles, some of them use features that mobile devices do not support. Current mobile devices and most tablets do not have enough system resources to support many post-processing effects, and dense particle systems.

Since the hallway illusion depends on a bump offset effect in the material, the illusion does not work on mobile devices at this time with the default settings. The tester recommends artists use a combination of billboards and simple geometry as an alternative method of achieving a similar effect.

The tester also recommends creating all particle systems for a mobile game in a program such as After Effects, and then rendering the particle systems on to sprite sheets. Particle systems require resources for both calculating the physical behavior of particles, and rendering them. A sprite sheet does not require any behavior calculations, so they use less memory resources than particle systems.

The tester also recommends that game developers replace any post process effects such as distortion, lens flares, and bloom with alpha cards for mobile and tablet platforms. He also recommends developers replace distortion on meshes with vertex painted normals.

The tester used an iterative and loosely structured workflow for the majority of the construction of the artifact. The tester overscoped his project in the beginning of planning. He originally created his level artifact with a lot of empty space, and planned an individual room for each illusion. He created modular pieces in the beginning of the project, and quickly found the level aesthetically repetitious. He then removed the majority of empty space in his level, and rearranged the layout of the illusions. At that point he focused most of his time on refining the illusions.

During the production of assets, the tester organized his working textures in layer folders within Photoshop. Each texture set had its own Photoshop file with a unique name. He created the majority of his assets grouped together in 3DS Max files named for their contents. He then exported his assets to folders, and each asset followed a naming convention which identified the file directory location and the function within the engine. This workflow maintained an organized file directory, and made importing assets into the engine easy.

The tester suggests developers organize their projects even further by taking some additional steps that he did not take during development. The tester never created an asset list at the beginning of his project, and often made assets as he needed them within the engine. He highly recommends future developers create an asset list, even for smaller or personal projects.

The tester also did not name the contents of his working files after the final naming convention. He wasted time looking up the naming convention he created for assets within the engine when he rewrote files that he revised. He suggests developers label Photoshop layers and meshes within 3DS Max working files with the exact name of the final asset.

The tester created too many meshes inside of his working 3DS Max files. He could have increased his work efficiency if he created his meshes in more files organized by purpose. The tester found that he spent a lot more time preparing assets for texture baking because of the cluttered nature of the files.

The tester baked shadow maps and normal maps within 3DS Max using the 3dsmax.scanline without advanced global illumination render setting. He captured shadows from a single skylight within the scene. His skylight had 20 rays per sample for the majority of his project. Although the settings provided crisp shadow maps, it took 18-24 minutes for 3DS Max to bake the textures. The tester suggests that future developers reduce baking times by using only 1-3 rays per sample on a skylight, and then hide any grainy artifacts in the shadow map with a Gaussian blur filter in Photoshop.

V. Conclusions

The tester demonstrated methods of creating memorable experiences for a player by using environmental illusions. In order to test the hypothesis, the tester constructed an artifact containing different types of environmental illusions. The development of the assets used various software tools, and the tester assembled it in the Unreal Development Kit. Afterwards, the tester analyzed the success of various illusions.

One limitation that the tester found while constructing the artifact was a lack of proper level design experience. He reconstructed the entire level in UDK multiple times. If the tester did more planning of level layout in the beginning, he could have used the time towards recording more data. The tester used an iterative and somewhat organized workflow.

The tester created this thesis artifact with the goal of demonstrating environmental illusions, so he constructed the artifact around the illusions. The opposite holds true in actual game development. During video game production, the illusions serve as ways for developers to decrease performance issues and production time. If the tester creates the artifact a second time, he would create a smaller level with more illusion variants. He recommends that future experimenters research into detail how the illusions affect performance on mobile devices and tablets.

VI. References

Z:\C19\MIT Repository Folders\Al-Tamimi

[1] Platinum Games, "A DAY IN THE LIFE OF AN ENVIRONMENT ARTIST," Platinum Games, 2nd April 2014. [Online]. Available: http://platinumgames.com/2014/04/02/a-day-in-the-life-of-an-environment-artist/.

[2] W. Prinzmetal, A. P. Shimamura and M. Mikolinski, "The Ponzo Illusion and the Perception of Orientation," 2001.

[3] M. Bach, "Moon Illusion from Visual Phenomena & Optical Illusions," 13 Jun 1999. [Online]. Available: <http://www.michaelbach.de/ot/sze_moon/index.html>

[4] N. Kay, "15 Forced Perspective Technique Examples", 3 July 2009. [Online]. Available: <http://digital-photography-school.com/forced-perspective>

[5] A. Kitaoka, "Akiyoshi's Illusion Pages," 3 June 2002. [Online]. Available: <http://www.ritsumei.ac.jp/~akitaoka/index-e.html.>

[6] S. Totilo, "An Oddly Long, Banter-Filled History of Mass Effect Elevators", (Kotaku) [Online]. Available: <http://kotaku.com/an-oddly-long-banter-filled-history-of-mass-effect-ele-472630862 >

[7] Diablo III. [PC Game DVD] Irvine, California, Blizzard Entertainment, 2012.

[8] Super Mario 64. [Nintendo 64 Game Cartridge]. Japan: Nintendo, 1996.

[9] Saints Row IV. [Steam PC Game Client]. Champaign, Illinois, Deep Silver Volition, 2013.

[10] J. Lauder, "The Ocean Spray in Your Face," July 1998. (Game Developer) [Online]. Available: <https://www.lri.fr/~mbl/ENS/IG2/devoir2/files/docs/particles.pdf.>

Hammed Al-Tamimi is a master's Student at the Guildhall at Southern Methodist University studying art creation in Plano, Texas, USA. He received a bachelor's of fine arts in game art and design at the Art Institute of Seattle in 2011 in Seattle, Washington, USA.

Boris Fisher is an art creation faculty member at the Guildhall at Southern Methodist University. He graduated from the Illinois Institute of Art with a bachelor's of fine arts in 2004 in Chicago, Illinois, USA. He was the Senior Artist at Day1Studios, LLC in Chicago, Illinois, USA.

Elivorio Luna is a faculty member at the Guildhall at Southern Methodist University. He graduated from the Art Institute of Dallas with an associates of applied arts in computer animation in 1999 in Dallas, Texas, USA. He was the Lead Artist at Terminal Reality in Lewisville, Texas, USA.