Shadows in 3D platformer games

A comparative analysis of shadowmaps and blob shadows

Faculty of Arts

Department of Game Design

Authors: Hangning Zhang, Clément Pirelli

Bachelor’s Thesis in Game Design, 15 hp

Program: Bachelor's Programme in Game Design

Supervisor: Masaki Hayashi

Examiner: Henrik Warpefelt

May, 2020

Abstract

The challenges found in three-dimensional platformer games heavily rely on the player's ability to perceive depth and it is therefore crucial for them to feature shadows as they are an important depth cue for the human visual system. In this paper, we compare two different kinds of shadowing solutions, blob shadows and shadow-maps. Subjects split into two groups each played a differently shadowed version of a 3D platformer game prototype as well as a common, unshadowed version. The first group's prototype featured blob shadows while the second group's used shadowmaps. The players were asked to input their own proficiency level from one to five into the program. The data collected through in-prototype logging to a file hinted at shadowmaps being more desirable as the shadowmap group's delta from shadowed to unshadowed playthrough was lower than the blob shadow group's. However, this data featured a number of problems, such as low sample size, subjective evaluation of player proficiency in 3D platformer games and different distribution of proficiency in and across groups.

Key words: Computer games, 3D platformer games, shadows, depth perception, game design

Sammanfattning

Utmaningarna funna i tredimensionella platformsspel beror starkt på spelarens förmåga att uppfatta djup och det är därför avgörande för spelaren att ha skuggor eftersom de är en viktig antydan till djup för det mänskliga visuella systemet. I detta examensarbete jämför vi två olika typer av skuggningslösningar, "blob shadows" och "shadowmaps". Försökspersonerna delades upp i två grupper, var och en spelade en annorlunda skuggad version av en prototyp för ett 3D-plattformsspel samt en ordinär, version utan skugga. Den första gruppens skuggor gjordes med "blob shadows" medan för den andra gruppen användes "shadowmaps". Spelarna ombads att mata in sin egen kompetensnivå från ett till fem i programmet. Uppgifterna som samlats in via loggning i prototypen skrevs till en fil som antydde att "shadowmaps" var mer önskvärda eftersom "shadowmap"-gruppens delta från skuggad till icke-skuggad igenomspelning var lägre än skugga-gruppens. Dessa data innehöll emellertid ett antal problem, såsom låg provstorlek, subjektiv utvärdering av spelarens kunskaper i 3D-plattformsspel och olika spridning av kunskaper i och över grupper. Insikter för framtida forskning inkluderar skapandet av en objektiv ram för utvärdering för kunskaperna för spelarna såväl som förminskningen av prototypens omfattning.

Nyckelord: Datorspel, 3D-plattformsspel, skuggor, djupuppfattning, speldesign

Contents

1 Introduction

The first three dimensional (3D) platformer game, Alpha Waves (Atari SA, 1990), featured a crude triangular shadow under the player avatar. Nowadays, when photorealistic shadows are relatively inexpensive to render, one may wonder why some contemporary video games choose to use non-photorealistic shadows instead.

3D platformer game developers should be able to make educated design decisions about the type of shadows their game should feature since shadows are an important marker for the depth perception of their players.

Our study aims to shed light on the effect of shadows on gameplay in 3D platformer games, as there has been little academic research which focuses on this area. By investigating the behavior of players when playing the same game with different types of shadows, we can conclude a recommended shadow solution to aid in gameplay and answer the question “do blob shadows affect 3D platformer gameplay differently than shadowmaps?” Our initial hypothesis is that there will be no gameplay difference between blob shadows and shadowmaps due to ground markers being more important than silhouette in depth perception.

In the optic of answering this question, a test featuring a game prototype with different shadow solutions was developed and tested on two groups, one featuring shadowmaps and the other, blob shadows. The game prototype logged the players’ performance, which was then compiled, analyzed and discussed. Insights for future work in the area were then provided.

2 Background

First, we will explain what a shadow is by introducing the physical principle of shadow and its terminology. Then we will discuss how the use of shadow will aid the user's depth-perception in a digital 3D world. Two commonly used shadow solutions in 3D platformer games will then be introduced, followed by some examples of how game studios choose their type of shadows in game.

2.1 Shadow terminology

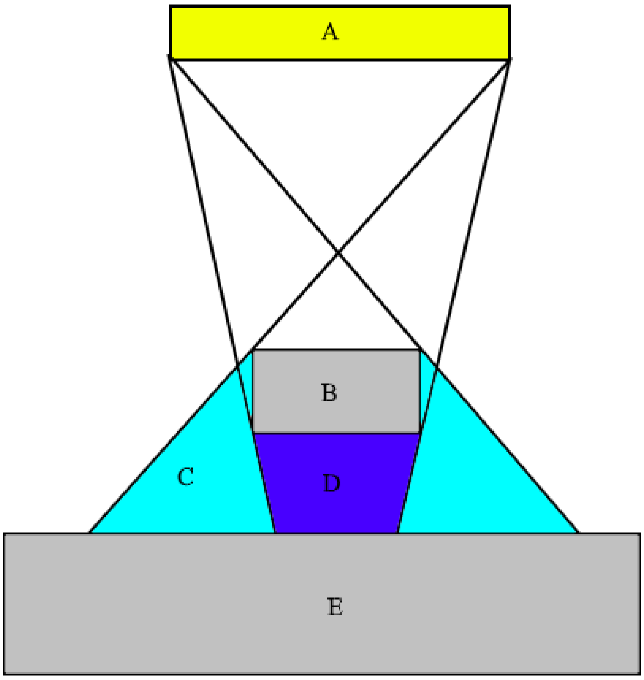

Figure 1, diagram of a rectangular light casting a shadow on a surface

Shadows are dark areas which are cast onto a surface, referred to as the receiver, because light coming from a light source is occluded by another surface, referred to as the occluder. The part of the shadow in which the receiver completely obscures the light is called the “umbra”, while the part which is only partially occluded is called the “penumbra”, as seen in Figure 1, where surface “B” occludes the light coming from a light source “A” onto a receiver “E”. The area “D” is umbra, while the area “C” is penumbra.

Shadows can be described as being “hard”, having no penumbra whatsoever (as can happen when simulating an infinitely small point casting light, such as a point light), or soft, having a penumbra.

2.2 Shadows and depth

With the notable exception of Virtual Reality, computer-rendered images project a three dimensional environment into the two dimensions of the image plane. Kim et al. (1987) write: “to present three-dimensional information on a two-dimensional surface, a depth cue for the third dimension must be provided.”

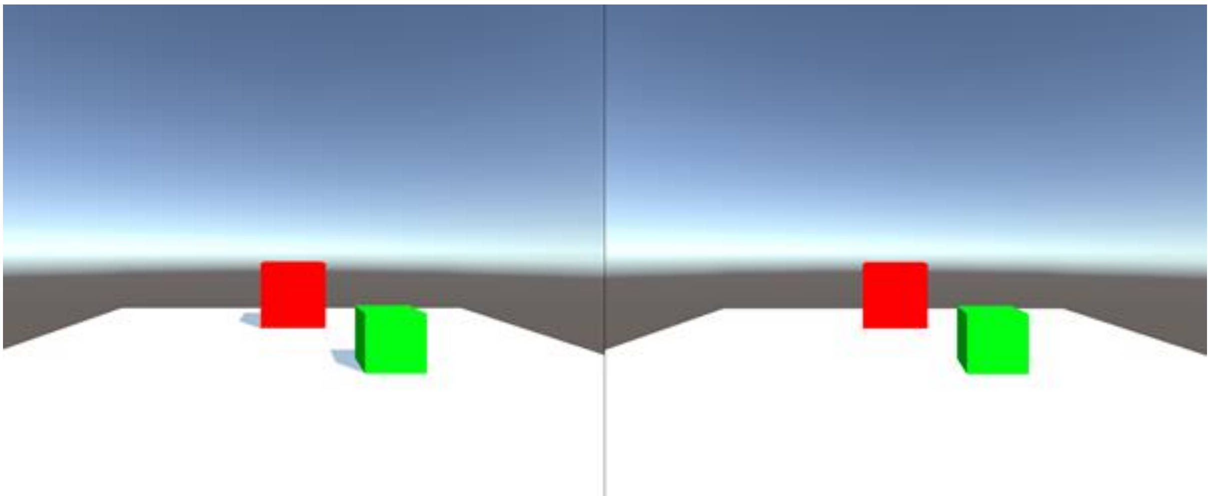

Figure 2, comparison between the same scene (shadowed on the right, un-shadowed on the left). Rendered in the Unity engine (Unity Technologies, 2020)

Shadows are one such depth cue. Surdick et al. (1994) write that without a marker acting as an anchor to the ground, a “floating object’s location would be ambiguous relative to the ground”. They test other depth cues such as linear perspective alongside ground markers. Such markers “[simulate] cast shadows from the objects”, though they used simple lines going from the object to the ground instead of photorealistic shadows.

Such a grounding effect is shown in Figure 2, where the same scene is rendered with and without shadows on the left and right respectively. In the scene without shadows, the depth of the red cube is ambiguous, while in the scene with shadows, shadows “ground” it as being behind the red cube.

The importance of such markers in depth perception explains why all three-dimensional platformer games we have come across feature shadows on the player avatar : shadows help anchor the player avatar to the ground even when they are in the air, which is crucial to the completion of challenges usually found in 3D platformer games such as jumping from platform to platform.

2.3 Different shadow solutions in games

Computer graphics are bound by the speed at which one may render one image, or a “frame”. The rate at which interactivity starts to develop is around six frames per second (also shortened as “fps”), though most video games aim for thirty or sixty fps, or even higher (Akenine-Möller et al, 2018). A rate of sixty fps implies no more than 16.7 milliseconds needs to be spent on any single frame. Real-time computer graphics techniques used to simulate shadows are therefore more interested in approximations of the correct results, as calculating a physically correct simulation of light in a scene simply takes too much time.

2.3.1 Blob shadows

Blob shadows are an approximation of the soft shadow cast by a light source right above the occluder. A simple black circle (or “blob”) is applied on the receiver, regardless of the true silhouette of the object, as seen in Figure 3. This creates the aforementioned sense of “grounding” for objects, while also being computationally inexpensive.

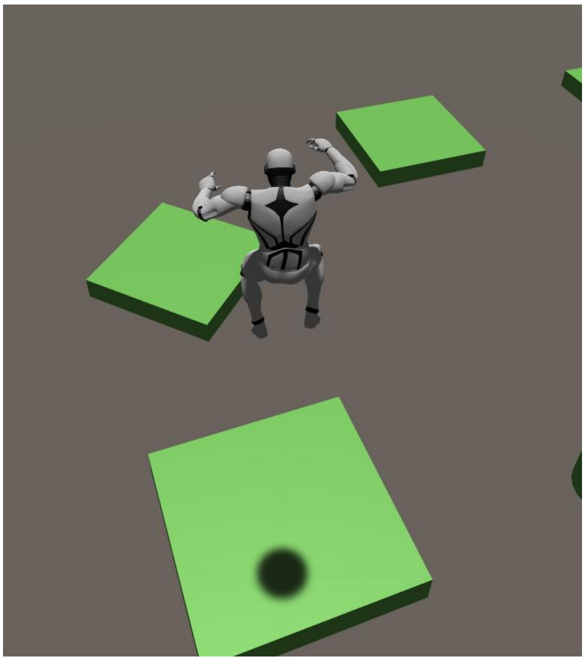

Figure 3, blob shadow in the Unity engine (Unity Technologies, 2020)

Some implementations place a flat textured quadrilateral (or “quad”) onto the receiver and angle the quad so that the plane it forms is tangent to the receiver’s surface normal. This works well for most cases except when the receiver has uneven normals, such as a bumpy road or the edge of a platform.

Other implementations project the texture onto the receiver as part of a pixel (also called fragment) shader.

2.3.2 Shadowmaps

Shadowmaps (Williams, 1978) are a more realistic approximation of the shadows cast by an object, as shown in Figure 4.

To render hard shadows, the 3D scene's depth is rendered from the perspective of the light and stored in a buffer. The distance from a point to be shaded to the light is then calculated and compared to the depth value in the previously generated buffer. This allows for the detection of potential occlusion from the point to the light, which implies an object is occluded.

Figure 4, shadowmaps in the Unity engine (Unity Technologies, 2020)

While it yields more accurate results and renders an accurate silhouette of the character as seen from the light, this technique is highly computationally expensive as compared to blob shadows, and was therefore absent in early 3D platformers.

There are several other problems shadowmaps encounter which blob shadows do not. If the resolution of the depth buffer is too small, or if the light's position is far away, the shadow will be jagged because of aliasing, a phenomenon in which the rate at which a continuous signal is sampled is too low, as seen in Figure 5, where the rate of sampling, in grey, of the black signal produces the aliased red signal.

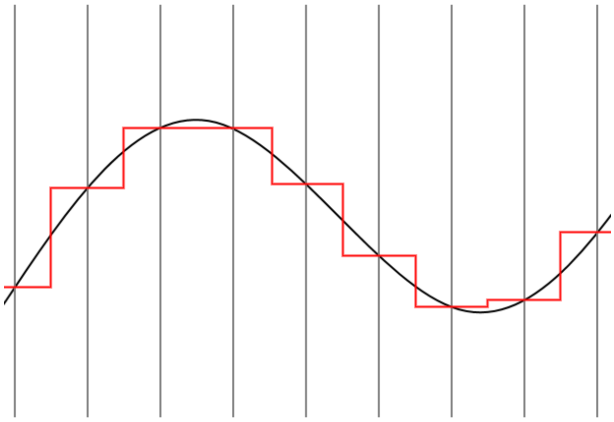

Figure 5, An example of aliasing

Shadows produced with shadowmaps may be softened using percentage-closer-filtering (Reeves et al., 1987), needing even more computation. Their precision may also be improved by using a technique called “cascaded shadowmaps” (Zhang et al., 2006), in which multiple depth buffers are rendered, with different positions inside the camera frustum. This also increases the amount of computation required.

2.4 Use of shadows in video games

These following examples will help us get a better understanding of shadow selection by game developers. In this section, we list several 3D platformer games and which shadow solutions they feature.

Super Mario 64 (Nintendo Ltd., 1996) featured extensive use of blob shadows as markers for both the main character and for environments, as seen in figure 6 with both the main character and the trees.

Figure 6, Super Mario 64 gameplay screenshot

Super Mario Galaxy (Nintendo Ltd., 2010) featured use of both shadowmaps for the main character and blob shadows for other moving characters, as seen in Figure 7.

Figure 7, Super Mario Galaxy gameplay screenshot

Effie (Inverge Studios, 2019) also made simultaneous use of both shadowmaps and blob shadows. While the player avatar is in the air, a blob shadow shows where they are grounded. After the avatar lands on the ground however, the blob shadow disappears and a realistic shadow is shown, as seen in Figure 8.

Figure 8, Effie gameplay screenshot

3 Method and materials

So as to comparatively analyse blob shadows and shadowmaps, we gathered data by having playtesters test a game prototype (source code can be found in Appendix A) with different shadow solutions.

3.1 Explanation of the test

Testers who agreed to participate in the study were split into two groups. The two groups received different versions of the prototype featuring one of the two shadowed versions, as well as the unshadowed version of the game.

The unshadowed version of the prototype is meant to act as a baseline to which we can compare the other two kinds of shadowed version (blob shadows and shadowmaps). The problem with simply making the testers play first with one and then the other shadowed versions is that the players learn from their experience testing the prototype, meaning they will perform better during their second playthrough than their first. To avoid the noise from the learning process, testers first played one of the two shadowed versions of the prototype and then the unshadowed version. This meant that our sample size was effectively split into two groups and therefore halved. However, we believe this approach yields more accurate results due to it removing part of the noise coming from the player learning the level.

We will be analyzing the delta between the shadowed and unshadowed versions' data. This will assure that the difference between the two versions is in comparison to one common result, which should be the same for the same tester regardless of which shadowed version the tester played beforehand.The participants were sent their assigned version of the prototype remotely and were asked to play the game in their free time.

At the start of the test, the participants were asked to rate their own proficiency at three dimensional platformer games on a scale of one to five. They then played their assigned shadowed version of the prototype first, followed by the unshadowed version. A log file of relevant data, including the previously answered proficiency level of the participant, was written. Its exact contents are discussed in section 3.2. The participants were asked to send their log file to the research team after completion of the test. Incomplete log files were discarded.

3.2 Explanation of the prototype

3.2.1 Overview

The mechanics of the prototype reference typical 3D platformer games. Players control an animated avatar and jump from platform to platform until they reach the end of the level, as seen in Figure 9. Once testers land on a new platform, their position will be stored. In the case they fall off, the player avatar will respawn on the last position stored. There is no time limit, so testers will complete the game eventually. Testers need to play through the prototype twice, the first time with blob shadows or shadow maps on and the second time without any shadows on the player character.

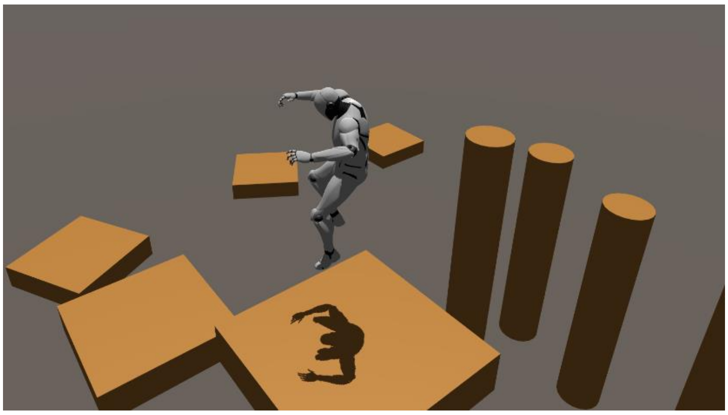

Figure 9, A screenshot from the prototype

3.2.2 Engine and Plugins used

The game prototype was developed in the Unity engine, which is a modern game development solution used in contemporary games such as “Ori and the Will of the Wisps” (Moon Studios 2020). The Unity asset store features a series of useful plugins, which shortened the time needed to build the prototype. In this project, we used the “Third Person Controller” and “Fast Shadow Projector” plugins. The former package includes a multifunctional 3D character controller whose attributes, such as jump height, camera controls, air speed and gravity, can be adjusted, as seen in Figure 8. “Fast Shadow Projector” provided us with a simple solution to render the blob shadows featured in the prototype.

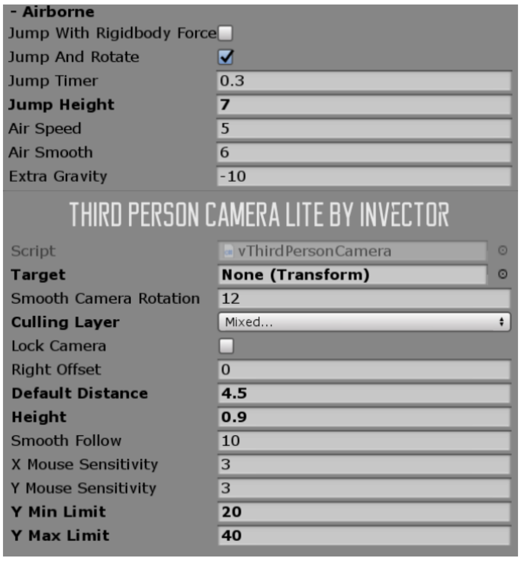

Figure 10, Unity editor view of the Third Person Controller plugin

3.2.3 Level design

The first version of the level featured multiple mechanics used in real 3D platformer games, like moving platforms and rotating platforms. A small group tested these mechanics as part of a pre-test. Their feedback was that the mechanics focused too much on the timing rather than the accuracy of jumps which would've brought unnecessary noise into the results of the playtests.

We therefore made the current version of the prototype focus more on the accuracy of the jumping by removing the mechanics of moving and rotating platforms. The level only consisted of platforms with different dimensions and in different positions, arranged in three color-coded parts which featured gradually increasing difficulty, as seen in figure 11.

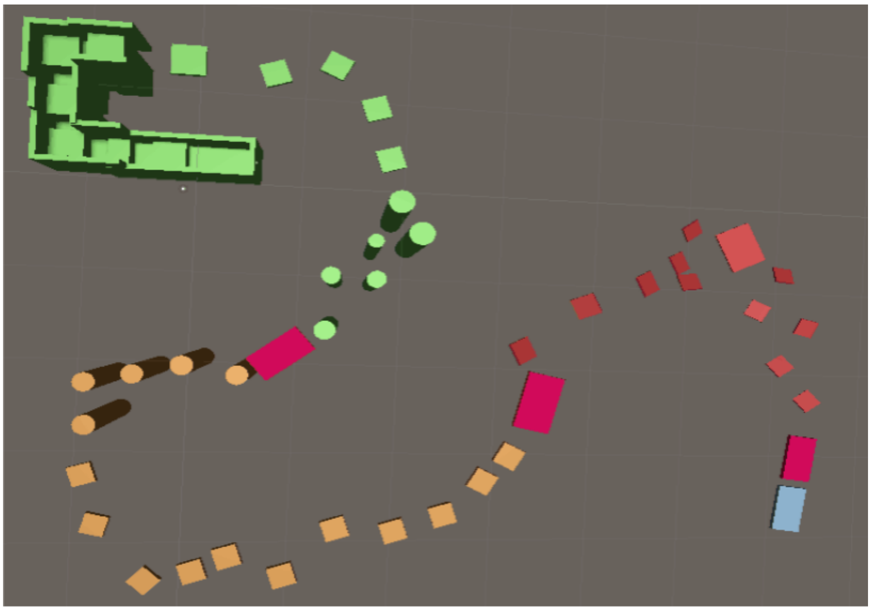

Figure 11, the layout of the prototype

The green part is the first and easiest area of the prototype. It starts with a room testers will not fall off of due to it being surrounded by high walls. This is so players get used to the mechanics and controls of the game. Both the size of the platforms and the gap between each platform is tuned so that this part of the level is the least challenging.

The orange part which follows requires testers to jump and land more accurately due to the platforms being smaller.

The red part is the final and hardest part of the prototype. In this section, the platforms are slightly tilted. Each platform is also smaller and further apart than in the other parts of the level which makes it harder to evaluate the depth of the player avatar.

The pink platforms are triggers for data collecting. As the testers jumped on them, the relevant data for the previous part is logged to a file. The data collection for the next part of the level then triggered when the tester left the pink platform.

The blue platform is the final platform of the prototype which acts as a simple trigger for finishing the level. When testers jumped on it, the data for the entire level was collected. One tester needed to clear the level twice, meaning the first time a tester jumped on the blue platform, the player avatar was moved to the start of the green area and the shadows were removed. The second time the tester jumped on the blue platform, the prototype was closed which indicated the end of testing.

3.3 Data needed to answer the research question

In the prototype, four kinds of data is gathered in both shadowed version and the unshadowed version.

1. The **proficiency** of the tester in 3D platformer games.
2. The times the tester **fell** in the prototype.
3. The times the tester **jumped** in the prototype.
4. The **total time spent** for completing the prototype.

One main variable we cannot obtain from the players simply playing the prototype is how good their 3D platformer game skills are. We need the player to answer what their proficiency is so that we can compare the data of testers accurately by taking their proficiency level into account.

The amount of times the player falls in the prototype is directly proportional to their performance in the game. The more a player falls, the more poorly they are able to estimate depth.

The amount of times the player jumps gives us a rough estimate of their level of proficiency as well. A player who jumps from one platform to the next will jump less than one who jumps from one platform to itself, which is a sign a player is not comfortable playing the game. Furthermore, the easy section of the level (section 3.3.3) features areas where a player may not jump off the platforms, simply colliding with the floor if they do not manage to jump to the next platform. Overall, jumping more correlates with more difficulty playing the game.

The time spent in the level also indicates the level of ease with which the player traverses the prototype. A player who falls and hesitates more will take more time than a player who perfectly traverses the level.

4 Results

Testing was performed online, by sending one of the two versions of the prototype to the testers, who then played their assigned version and sent the log file back to us. In total, 21 testers played the prototype. Two testers' log files were incomplete and therefore are not included in the results. Testers were mostly Swedish students from the ages of 19 to 28.

The full results can be found in Appendix B.

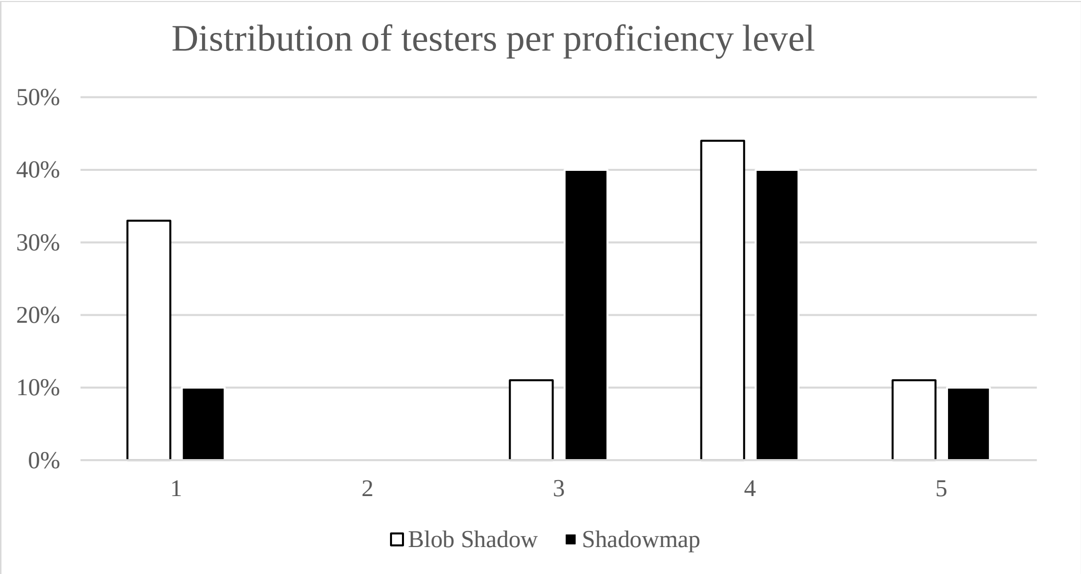

Figure 12, Distribution of testers by proficiency level

The deltas are calculated as such :

$$\Delta = Shadow - Control$$

The averages are then calculated by summing all of the deltas together and dividing by the amount of deltas

Table 1. Averaged proficiency and deltas per shadow solution

	Proficiency	**Jump Δ**	**Time Δ**	**Fell Δ**
Blob Shadows	3.00	3.11	32.19	4.67
Shadowmap	3.40	2.70	20.76	1.50

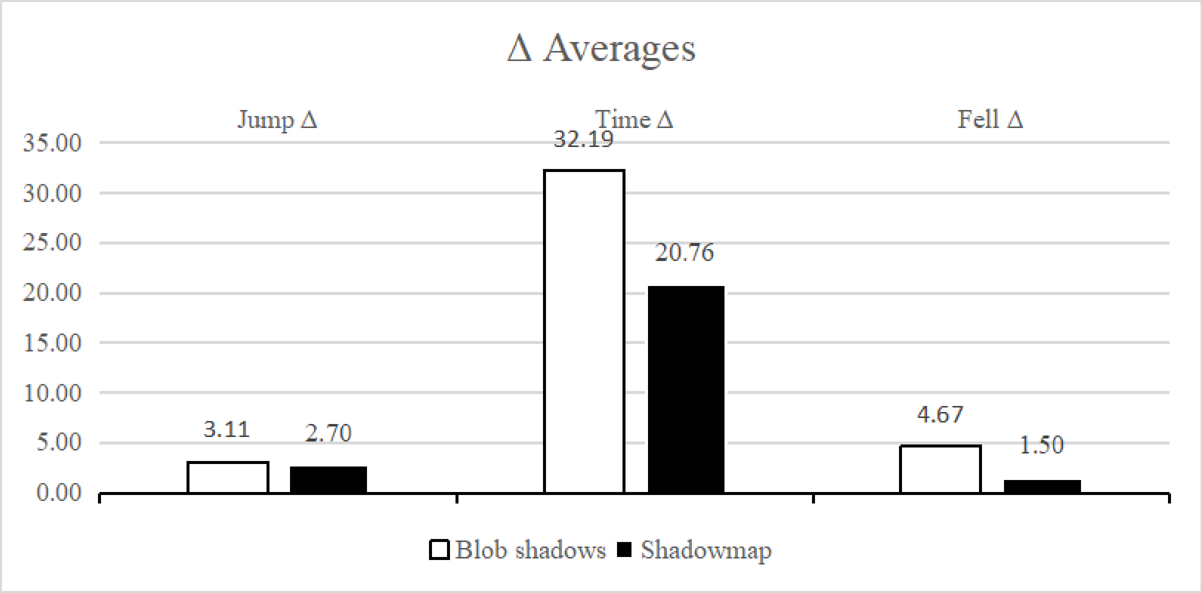

Figure 13, Average delta data for blob shadows and shadowmaps

5 Analysis

5.1 Average delta comparison

Positive deltas show testers's performance improved in both solutions. Naively, this would suggest that having no shadows in the game would make it easier for the players to complete the level. However, the player learns how to play the game during the first playthrough, meaning they are naturally more proficient during their second playthrough, which features no shadows.

If players learn at a higher rate depending on which shadow solution their first playthrough featured, this in part means that this shadow solution is more desirable. This means that even though players might learn at different rates, the deltas of the results can still be compared, with a delta tending more towards a negative value being associated with a better performance by the player.

We can observe that the shadowmap averages tend more towards lower values (Time Δ 20.76 vs 32.19, FellΔ 1.50 vs 4.67), indicating a better performance by the testers. This may be counterbalanced by the different distributions of proficiency in the two groups, as well as a slightly higher average level of proficiency reported by the shadowmap testers, who would therefore jump less, fall less and spend less time completing the prototype. Notwithstanding the average proficiencies of each group being only slightly different, a third (3 out of 9) of the blob shadow testers, compared to only a tenth (1 out of 10) of the shadowmap testers, rated their own proficiency as being 1 out of 5. This may be correlated with the blob shadow group taking more time in their shadowed version than the shadowmap group, with an average absolute time of 171.1s and 127.6s, respectively.

However, this expected correlation between proficiency and tester performance was not found in our data. For example, two players from the blob shadow group, who both reported a very low level of proficiency (1 out of 5) had drastically different performances in the game (-4.9 vs 69.7 time delta, -1 vs 13 fell delta). One player from the blob shadow group who reported a proficiency level of 1 out of 5 had a better absolute performance in their shadowed playthrough than another player from the same group who reported a proficiency level of 4 out of 5 (respectively, 123.8s absolute time vs 150.8s, 11 times fell vs 26).

5.2 Standard deviation analysis

To analyze the data further, it is important to bring up the concepts of variance and standard deviation.

Variance measures the spread of a data set. The smaller the variance, the more the data set tends towards its mean, or average. A set whose variance is small is therefore probably more aligned with reality than a set whose variance is high, simply due to the nature of normal distributions. An average of highly variant values therefore does not represent the population adequately.

Variance for a set of samples is calculated as :

$$\frac{\sum_{i=0}^{n}(x_i - \bar{x})^2}{n-1}$$

Where n is the amount of samples, x_i is the ith sample starting with sample x_0 and $\bar{x}$ is the average of the samples.

The standard deviation s is calculated as the square root of variance. It measures a sample's average difference from the mean of the data.

The current average data suggests a trend towards shadowmaps being more desirable than blob shadows. However, the variance of both data sets is considerable. As seen in Table 3, the data sets seem to be highly influenced by individual factors instead of representing the majority of the population.

Table 2, Variance and standard deviation for each delta set

	Variance (Jumped Δ)	Variance (Time Δ)	Variance (Fell Δ)	Std. Dev. (Jumped Δ)	Std. Dev. (Time Δ)	Std. Dev. (Fell Δ)
Blob Shadows	29.11	874.35	42.75	5.40	29.57	6.54
Shadowmap	16.90	256.80	60.94	4.11	16.02	7.81

The variance for time delta, for example, is quite high. Standard deviation for the blob shadow group is 29.67s, compared to shadowmaps with 16.02s. Comparing this to the time delta itself (32.19 and 20.76 respectively) we can see that the standard deviation is comparatively very large (92% and 77% respectively). This further suggests our data is not significant.

5.3 Statistical significance testing

To analyze the significance of the data, we performed a t-test (Student, 1908). A student's t-test would be suitable for two data sets with similar sample sizes and variance. However, in our case, there is a difference in variance between the two samples, meaning the commonly used Student's t-test is not suitable for our data (Ruxton, 2006). We therefore used the unequal variances t-test as it is more reliable when two samples have unequal variances or unequal sample size.

First, the standard deviation s must be calculated, which is simply the square root of the variance of the sample.

The unequal variances t-test defines the t-value as :

$$t = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\frac{s_1^2}{n_1} - \frac{s_2^2}{n_2}}}$$

Where $\bar{x}_1$ and $\bar{x}_2$ are the sample means, s_1 and s_2 are the sample standard deviations and n_1 and n_2 are sample sizes.

The estimated approximate degree of freedom in the unequal variances t-test is defined as :

$$v \approx \frac{\left(\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}\right)^2}{\frac{s_1^4}{n_1^2 v_1} + \frac{s_2^4}{N_2^2 v_2}}$$

Where v_1 and v_2 are $n_1 - 1$ and $n_2 - 1$, respectively.

Once the t value and degree of freedom have been computed, we used Microsoft Excel to calculate the p value.

Table 3, Unequal variance T test results

	Sample 1 (Blob shadow)			**Sample 2 (Shadowmap)**			**Unequal variance t-test**		
	n_1	$\bar{x}_1$	s_1	n_2	$\bar{x}_2$	s_2	t	v	P
Jump Δ	9	3.11	5.40	10	2.70	4.11	0.185	14.923	0.856
Time Δ		32.19	29.57		20.76	16.02	1.031	12.040	0.323
Fell Δ		4.67	6.54		1.50	7.81	0.963	16.928	0.349

Using an alpha value of .05, which is a commonly used level of significance in scientific research, we can see that none of our data sets can be proven to be statistically significant as their p-values are all above the alpha value. This implies we cannot reject the null hypothesis of our test as our sample is too noisy to carry any meaningful information.

6 Discussion

We cannot statistically prove that shadowmaps are more desirable compared to blob shadows. Instead we can only hint at the possibility of this being the case.

6.1 Weaknesses of the test

The test features several shortcomings which engender the high variances, percentages of standard deviation and p-values of the data sets. Many factors, such as the small size of the sample set, the impact of the learning process, the different group size of each proficiency level and the low correlation between proficiency level and performance all contribute to the lack of statistical significance of the results.

The current data sets only contain 10 and 9 testers each, meaning individual variations will have a significant impact on the study's results.

This is further accentuated by the level design of the prototype we created, which is much like other 3D platformer games. Although this may yield conclusions which better apply to real-life scenarios compared to simpler tests, this method features problems such as player learning significantly affecting the results of the study. Problems such as the high percentage of standard deviation in time spent in the level may be linked to different players feeling different levels of urgency in regards to finishing the prototype regardless of skill. Furthermore, the metric of times jumped in the prototype failed the T test with a p-value of 0.856. This implies the metric is fundamentally flawed.

The decision to split players into two groups each playing only one version and the control version, compared to simply asking players to play both shadowed versions in random order, may have been a mistake in this context. As previously stated, the results of the second playthrough were still heavily influenced by player learning. Since different people learn at different rates, it is therefore likely that splitting the players into two groups did not reduce noise from player learning.

The prototype we created featured level design much like other 3D platformer games. Although this may yield conclusions which better apply to real-life scenarios compared to simpler tests, this method features problems such as player learning significantly affecting the results of the study.

The current prototype also lacks an objective standard to evaluate the proficiency level for the testers, as we let the tester manually type in their 3D platformer game proficiency. An explanation for the abnormal difference in performance from testers in the same proficiency level could be that their evaluation is based on a different personal standard. An objective method to test the skill of each player would therefore highly increase the reliability of the proficiency data.

Furthermore, the proficiency data does not feature a similar distribution of people in different proficiency groups and in the two shadow groups. Although we have established that the current data regarding the proficiency level of the testers is not reliable, variation between the two datasets' proficiency levels should not occur.

In the next section, we will list ways the experiment can be improved which may result in a lower variance for the data.

6.2 Possible improvements

One possible improvement would be a higher sample size, which would result in less noisy data.

With the current method, there may not be a need to measure time spent in the level because of too many other influencing factors.

The prototype could also be much more jump-oriented, instead of consisting of a level similar to real 3D platformer games, as players can learn from playing the current prototype. The prototype could be reworked into one-jump levels, during each of which the players try a randomly chosen shadow solution. This might yield more relevant data as it would focus on the core gameplay of 3D platformer games and would remove the noise from the learning process. Not splitting our testers into two groups would also double the sample size of the test, which is likely to yield better results. The metric of jumping could either be reworked to only include jumps which do not start and end on the same platform, or simply removed. This would leave the prototype with only one metric, the amount of times the player fell, which might be insufficient to fully understand player behavior.

The proficiency level of testers should also have been better balanced between the two groups as well as across the groups. Ideally, Figure 12 would show a normal distribution for both groups. This can be achieved through both increasing sample size and keeping track of the chosen testers' proficiency levels, only choosing new testers which have a proficiency level which their dataset lacks.

An objective standard should be established to measure players' 3D platformer game proficiency. Self-appraisal of the testers' proficiency in 3D platformer games may yield results influenced by the level of confidence of the testers. With such a small sample size, this may affect the results of the test.

7 Conclusion

In this paper, we first showcased how shadows are rendered in 3D computer graphics by explaining the two shadow solutions usually used in 3D games, blob shadows and shadowmaps. References on how different shadow solutions are used in several 3D platformer games were shown.

A game prototype was built, with which we logged four different kinds of data (proficiency, times fallen, times jumped and total time spent) to indicate the performance of two groups, each testing a version featuring one shadow solution as well as a non-shadowed version of the prototype to act as control data.

The results of the test indicate shadowmaps are more desirable than blob shadows. However, because of the data failing an unequal variances t-test, it does not reach sufficient reliability and we therefore cannot statistically prove any gameplay difference between blob shadows and shadowmaps. Indeed, we pointed out multiple potential issues with the data in the discussion section, such as low sample size and a lack of an objective standard for the 3D platformer proficiency of participants, as well as possible solutions to improve the experiment.

Future studies in this area should tackle the problem of player learning affecting the results of consecutive testing, as well as establish an objective method of appraisal for the level of 3D platformer proficiency of testers.

References

Literature

Akenine-Möller, T., Haines, E., Hoffman, N., Pesce, A., Iwanicki, M., Hillaire, S., 2018. *Real-time Rendering, Fourth Edition*. A K Peters/crc Press.

Fan, Z., Hanqiu, S. , Leilei, X. , Lee Kit, L., (2006). “Parallel-split shadow maps for large-scale virtual environments,” Proceedings of the 2006 ACM international conference. Hong Kong, China

Kim, W.S., Ellis, S.R., Tyler, M.E., Hannaford, B.. & Stark, L. (1987). “Quantitative evaluation of perspective and stereoscopic displays in three-axis manual tracking tasks”, IEEE Transactions on Systems. Man. and Cv . bernetics. SMC-17, pp. 61-71.

R. Surdick, E. Davis, R. King, G. Corso, A. Shapiro, L. Hodges, K. Eliot. (1994). “Relevant cues for the visual perception of depth: is where you see it where it is?", Proc. 38th Mtg. Human Factors and Ergonomics Society, pp. 1305-1309

Student (1908). "The probable error of a mean", Biometrika, 6, 1-25.

Unity Technologies. n.d. “Unity for All.” Accessed 09 May 2020. https://unity.com/.

Williams, L. (1978). "Casting Curved Shadows on Curved Surfaces," Computer graphics (SIGGRAPH '78 Proceedings), vol. 12, no. 3, pp. 270-274

Ruxton G. (2006) “The unequal variance t-test is an underused alternative to Student's t-test and the Mann–Whitney U test”, Behavioral Ecology, Volume 17, Issue 4, pp. 688–690

Games

“Alpha Waves”, Atari SA, 1990.

“Super Mario 64”, Nintendo Co., Ltd, 1996.

“Effie”, Inverge Studios, 2019

“Super Mario Galaxy 2”, Nintendo Co., Ltd, 2010.

“Ori and the Will of the Wisps”, Moon Studios, 2020.

Appendices

Appendix A: Downloading for game prototype (2020.05.12):

https://github.com/Clement-Pirelli/PlatformerGameThesis

Appendix B: Raw data of proficiency and deltas per participant

Proficiency	Shadow Solution	Jumped (Shadow)	Time (Shadow)	Fell (Shadow)	Jumped (Control)	Time (Control)	Fell (Control)
1	Blob	51	123.8	11	48	128.7	12
1	Blob	54	222.6	22	45	152.9	9
1	Blob	51	118.6	22	49	95.9	23
4	Blob	59	150.8	26	47	103.3	17
3	Blob	58	190.3	30	60	170.8	20
5	Blob	47	120.7	13	46	97.6	5
4	Blob	54	193.4	22	60	184.2	29
4	Blob	48	134.7	10	44	117.7	7
4	Blob	55	284.6	20	50	198.7	12
1	Shadowmap	66	201.3	29	55	167.9	41
3	Shadowmap	49	144.0	12	45	122.6	8
4	Shadowmap	47	107.3	4	45	88.8	0
3	Shadowmap	46	88.5	17	49	104.61	23
5	Shadowmap	46	124.7	2	44	87.8	6
4	Shadowmap	51	107.3	18	44	79.9	5
4	Shadowmap	45	87.6	2	43	62.2	2
4	Shadowmap	44	107.9	12	45	91.7	10
3	Shadowmap	45	127.1	11	41	90.1	10
3	Shadowmap	50	140.5	22	51	133	9