university of groningen / faculty of science and engineering

Gamification of Virtual Ray Tracer

Bachelor Thesis

July 1, 2022

Author:
Peter Jan Blok

Primary supervisor:
Jiří Kosinka

Secondary supervisor:
Steffen Frey

Abstract

A ray tracing visualization application was made by C. van Wezel and W.A Verschoore de la Houssaije in their bachelor's theses in 2022. The application aims to help (Computer Graphics) students understand ray tracing better by exactly showing how a ray is being traced from the camera towards the light source(s).

In this thesis we set out to make the application more fun and entertaining while still keeping it educational by expanding the application with gaming features, which is also called gamification. We tested the users' engagement in the new version of Virtual Ray Tracer in a small user study. The user study showed that the modifications were effective in making the application more entertaining while even improving the educational aspect. With more research and improvements, Virtual Ray Tracer could become an even better educational application.

Contents

1 Introduction

Ray tracing is an essential part in the field of Computer Graphics. In the last few years, a lot of advances have been made towards real-time ray tracing. The concept of ray tracing, however, is difficult to grasp at first. That is why C. van Wezel and W.A Verschoore de la Houssaije made an application to visualize how ray tracing works [1][2].

The application is called Virtual Ray Tracer and an illustration of the application is shown in Figure 1. The application helps (Computer Graphics) students with understanding ray tracing. It consists of seven levels. Each level starts with a pop-up message with an explanation about the application itself or about ray tracing. In the level itself, the explained concepts can then be practiced and/or examined. After going through all the levels, the user should have a better understanding of how ray tracing works.

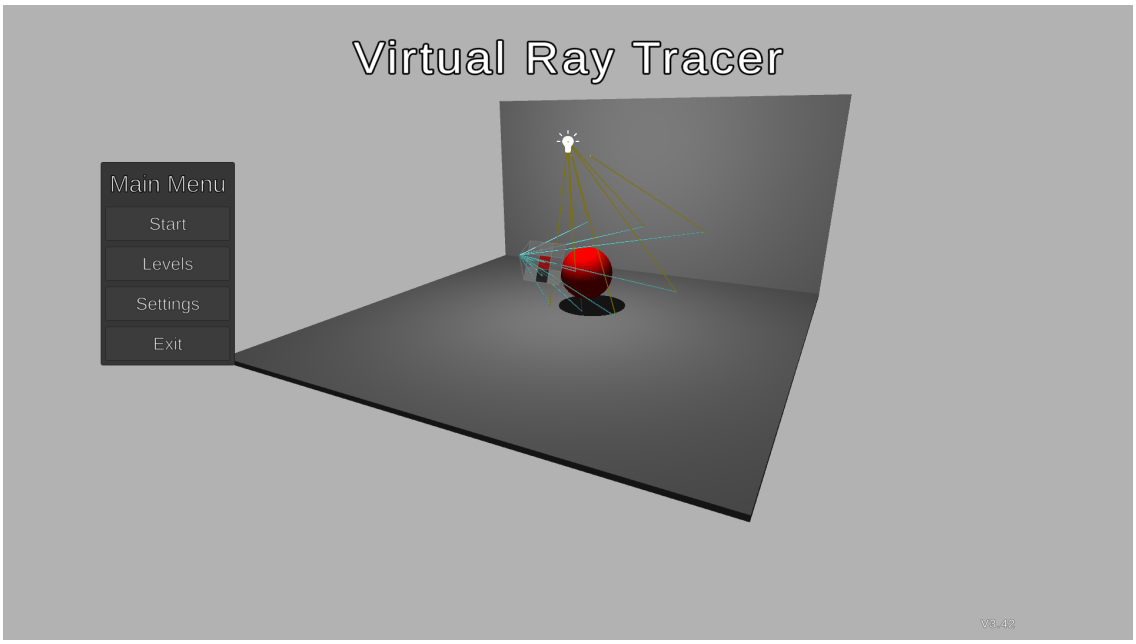


Figure 1: The original Virtual Ray Tracer application, made by C. van Wezel and W.A Verschoore de la Houssaije [1][2].

However, these pop-up messages with explanations are quite lengthy. A user with a lot of motivation coming into the application might be set back by these walls of text. Some users will skip the pop-up messages and miss important concepts. Others might exit the application before making it through all the levels. Additionally, the user is less motivated to go through the levels and learn new concepts, because they are not rewarded for it.

To address these issues, we made the application more entertaining and at the same time educational by extending the application with gaming features, which is also called gamification. In gamification, aspects and elements of games are integrated into everyday tasks or things [3]. The aim is to simulate the experience when playing games to motivate and engage users. To evaluate the effectiveness of the gamified Virtual Ray Tracer, we

reformulated the problem into three questions, implemented the relevant changes into Virtual Ray Tracer, and conducted a user study. The three questions are as follows:

1. What is the best way to implement gamification into Virtual Ray Tracer?
2. Does the addition of gamification improve motivation and engagement in Virtual Ray Tracer?
3. Which method or methods work best to improve motivation and engagement in Virtual Ray Tracer?

We start by exploring the existing ray tracing visualization applications and gamification concepts in Chapter 2. Then, in Chapter 3, we go into more detail about gamification in Virtual Ray Tracer. The used tools and implementation of gamification in Virtual Ray Tracer are summarized in Chapter 4. The user study and its findings are covered in Chapter 5. Finally, a conclusion is drawn in Chapter 6 and ideas for future work are proposed in Chapter 7.

2 Background

In this chapter we explore the existing ray tracing visualization applications and gamification concepts.

2.1 Ray tracing visualization

The Ray Tracing Visualization Toolkit [4] is a modular tool which allows investigators to debug the ray states for a particular scene. An illustration of the tool is shown in Figure 2. Ray states of a particular scene have to be recorded first, after which they can be individually investigated using the Ray Tracing Visualization Toolkit.

Rayground [5] is an online framework for rapid prototyping of ray tracing based algorithms. An illustration of the framework is shown in Figure 3. In Rayground, users are able to create scenes and implement ray tracing based algorithms without downloading any software (apart from a browser). The framework is, however, more geared towards the actual implementation of ray tracing.

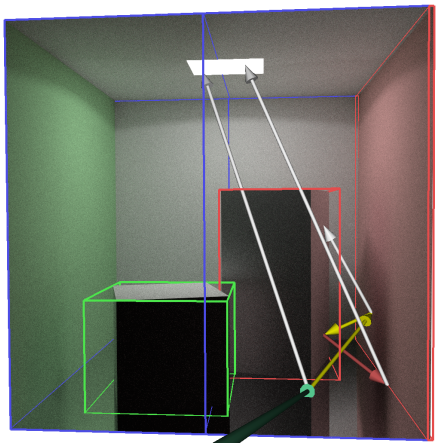


Figure 2: Layered visualization in the Ray Tracing Visualization Toolkit [4].

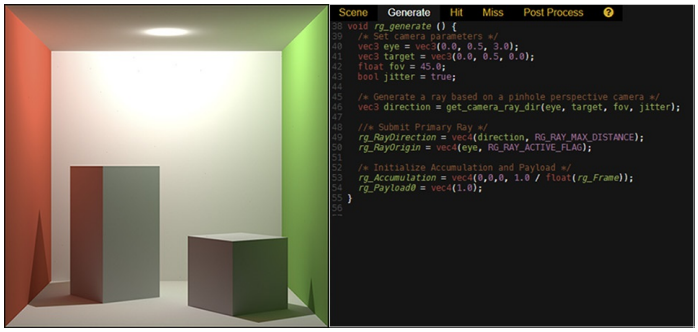


Figure 3: Interface of Rayground showing the preview window (on the left) and the shader editor (on the right) [5].

Virtual Ray Tracer [1][2] is an application that offers immediate ray tracing visualization. The application aims to educate its users about ray tracing by going through several levels while explaining and visualizing the ray tracing process. An illustration of the application is shown in Figure 1.

Although the Ray Tracing Visualization Toolkit and Rayground could potentially be used to show the inner workings of ray tracing, they do not offer immediate ray tracing visualization, let alone gamification. Virtual Ray Tracer does offer immediate ray tracing visualization and already included some gaming features such as levels and a main menu.

2.2 Gamification

The main idea behind gamification is for users to engage in specific actions/behaviors in return for rewards. When our brain expects these rewards, it will release dopamine, which will give us feelings of pleasure. Users will be motivated to engage with our application in order to receive these rewards.

We can think of motivation in two ways: extrinsic (outside) and intrinsic (inside). Intrinsic motivation is the motivation to do something for its own sake, purely for the pleasure of the task itself. Extrinsic motivation is the motivation to do something to achieve an external goal or to fulfill an externally placed requirement [6].

Intrinsic-based gamification can be achieved by helping users in finding their own reasons for engaging in the task. Deci and Ryan [7] refer to this concept as Self-Determination Theory. This intrinsic motivation can be attained by, for example, adding a score. Some people are highly motivated to get a good score. Others get intrinsic motivation from exploring, making their own (interesting) decisions, and playing with others [8]. Intrinsic motivation is hard to obtain as different people get intrinsic motivation from different design concepts.

In contrast, extrinsic motivation can be obtained a lot easier. Extrinsic-based gamification can be achieved relatively easily by adding Badges, Levels/Leaderboards, Achievements, and Points. Nicholson [9] refers to this concept as BLAP gamification. Badges can be earned by visiting new locations or completing certain tasks, achievements for reaching certain goals, levels and leaderboards to stimulate progression, and add a competitive element, points to unlock locations or to redeem for (real-life) items.

The problem with extrinsic rewards is that the player will need to be supplied continuously with these rewards in order to remain satisfied. However, with Virtual Ray Tracer, we are learning a skill with real-life benefits. We are already intrinsically motivated to learn about ray tracing, in which case extrinsic-based gamification can be very effective [9].

3 Gamification in Virtual Ray Tracer

In this chapter we explain the methods we used to apply gamification to Virtual Ray Tracer (VRT). The methods are ordered from most to least critical, meaning that we consider adding a step-by-step tutorial the most important modification to VRT, and animations the least important.

3.1 Step-by-step tutorial

As already mentioned before, VRT includes a lengthy pop-up message with explanations about the application itself or ray tracing before the start of each level. Figure 4 shows one of these pop-up messages.

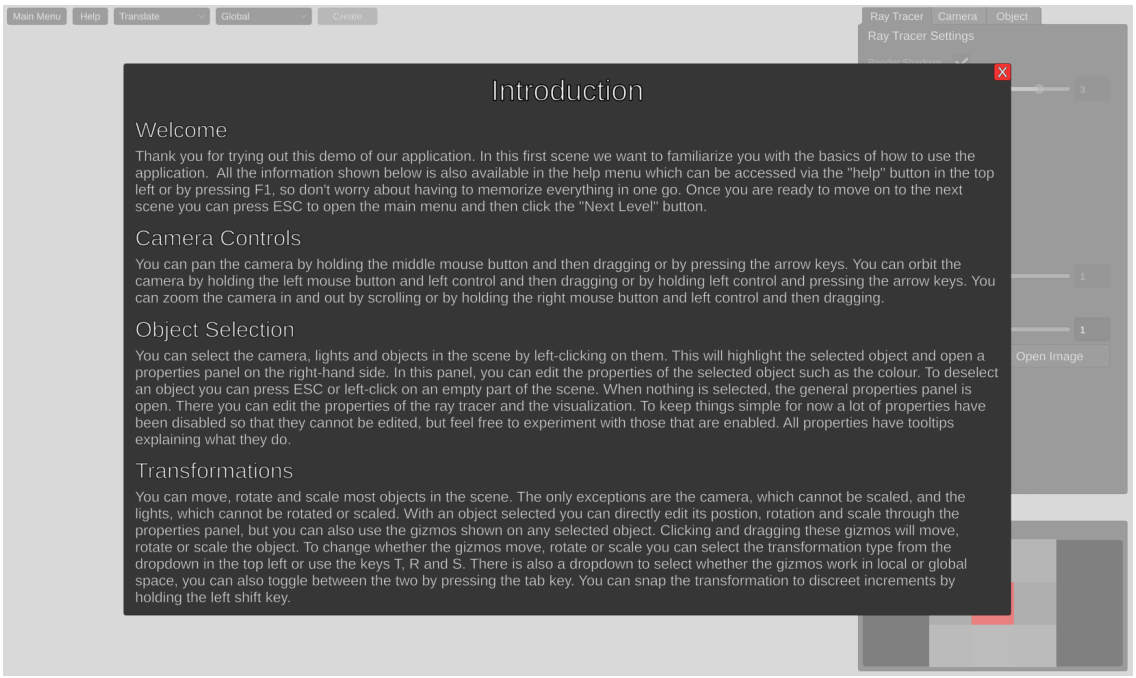


Figure 4: Pop-up message in the first level of the original Virtual Ray Tracer.

These pop-up messages might be skipped because of their length, causing users to miss important concepts. Others might read them but still miss important concepts because they failed to remember (some of) them.

Replacing these pop-up messages with a step-by-step tutorial (Figure 5 shows a sketch) solves these problems with some additional benefits. The lengthy text is split into small, individual steps, which makes the users feel like they never have to read large amounts of text. Furthermore, the complementary concepts can be separated from the critical concepts. The critical concepts are made mandatory and the complementary concepts are made optional, cutting down on the amount of required reading even further. Additionally, the steps that explain certain concepts can be transformed into small tasks by adding extra actions that the user needs to complete before being able to continue to the next step/task. This way concepts can be practiced immediately, making users more likely to remember them. If users

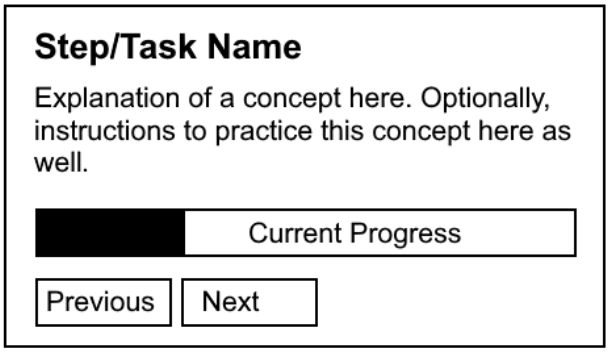


Figure 5: Sketch of a step-by-step tutorial.

do get lost somewhere, they can always go back to (a) previous task(s) and refresh their memory. Lastly, a progress bar is shown at the bottom. Besides giving the user an idea of their current progress, it also gives the user motivation to fill in the entire progress bar.

3.2 Points and unlockable items

To reward users extrinsically, the concept of points is introduced. Points can be earned by completing the optional and mandatory tasks in the step-by-step tutorial. Tasks that are more difficult and take longer are rewarded with more points, giving users more satisfaction when finally completing a more difficult/longer task. Optional tasks are also rewarded with more points to give the users more motivation to complete those tasks as well.

Of course, having lots of points would be useless if they could never be used somewhere. Fortunately, VRT offers a sandbox level where users are able to add objects themselves to create their own scenes. So, to add usefulness to the points, these objects are locked until enough points have been gathered, after which they are unlocked.

3.3 Badges

Badges are introduced to further reward users extrinsically. A badge indicates some kind of accomplishment. In this case, badges can be earned by playing with VRT for a certain amount of time, gathering a certain number of points, and creating a certain number of objects in the sandbox level. A design sketch is shown in Figure 6. Earning these badges

Figure 6: Sketch of a badge.

gives satisfaction to the users. In addition, some users are motivated to earn all badges, keeping them longer engaged with VRT.

3.4 Sounds

Sounds are important to create a better and more fun experience in VRT (or any other game/program); they create a more immersive experience. In this case, sounds can be played when interacting with the interface elements, completing tasks, and earning badges. The sounds add extra satisfaction to these elements. However, it is important that these sounds are not annoying as that would create the opposite effect.

3.5 Animations

Animations are also a great way to improve the experience; they make transitions feel fluid and add satisfaction to certain events. In this case, animations can be added when transitioning between levels, completing tasks, and earning points. As with sounds, it is important that these animations are quick to prevent any annoyance.

4 Implementation

In this chapter we explain how we implemented gamification into Virtual Ray Tracer (VRT). In order to explain this, we will first need to explain the relevant aspects of VRT itself.

4.1 Virtual Ray Tracer

Virtual Ray Tracer was created in the `Unity` game engine in combination with the `C#` programming language. Naturally, any modifications made to VRT will also be in `Unity` and `C#`. Figure 7 shows the VRT project opened in `Unity`.

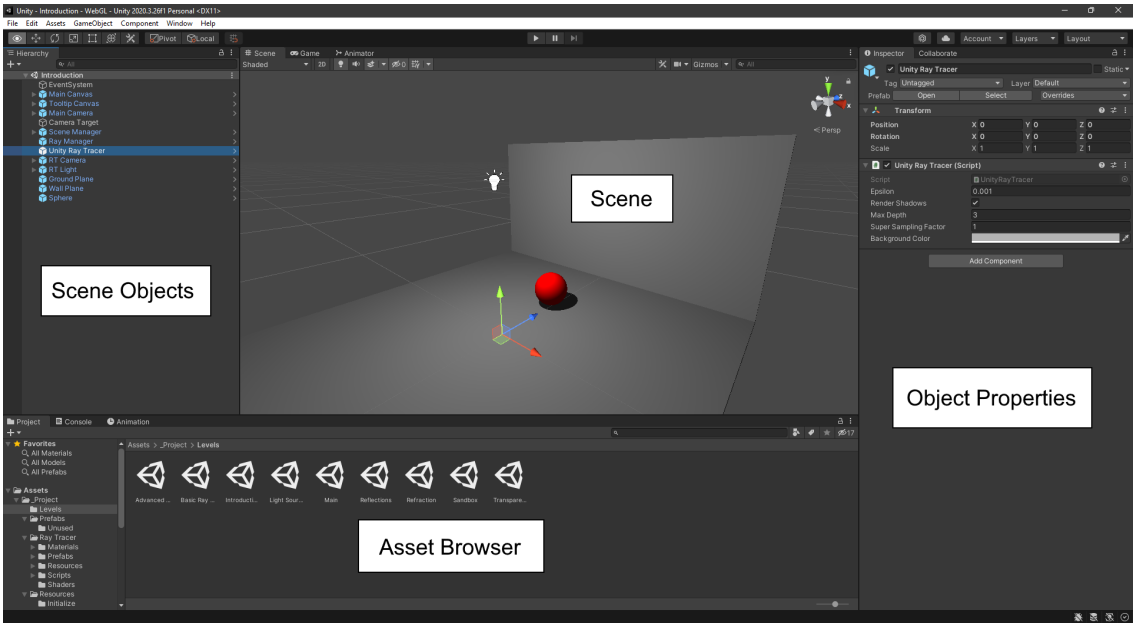


Figure 7: Virtual Ray Tracer project opened in `Unity` including annotations.

The User Interface (UI) is a relevant aspect of VRT as our modifications are (almost) entirely centered around the UI. UI elements, such as buttons, can be created by putting them in a Canvas object. VRT already has a Main Canvas object in each scene which holds all the UI elements except for tooltips. The Main Canvas is created as a Prefab (template), so changes to the Main Canvas Prefab are automatically applied to all scenes. Needless to say, we will be adding our modifications to the Main Canvas.

Another relevant aspect of VRT is the Game Manager object. The Game Manager object is instantiated when the first scene is loaded and kept alive until the application quits. This is different from the other objects in the scene(s) as they are destroyed as soon as the scene is unloaded. The Game Manager is therefore perfect for storing information that needs to be kept in memory for the lifespan of the application. VRT uses the Game Manager to store settings such as whether the FPS Counter is enabled and textures for different Cursor types. As expected, we will be using the Game Manager to store information from our modifications as well.

4.2 Modifications

Below is a description of how each modification is implemented into VRT.

1. **Step-by-step tutorial**

   The step-by-step tutorial is created as a Prefab with a Tutorial Manager script attached to it, and added to the Main Canvas. An illustration of the step-by-step tutorial is shown in Figure 8. It displays the task name and task description as well as the

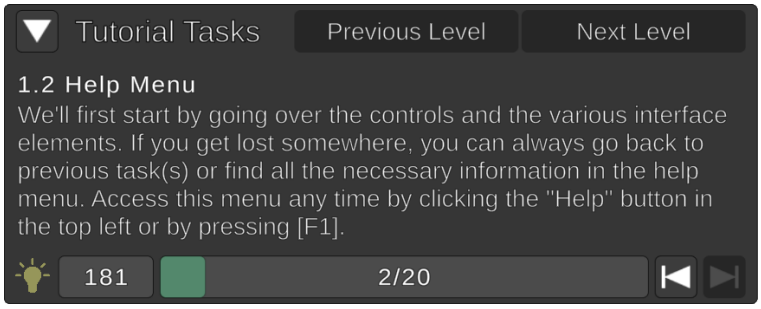


Figure 8: The step-by-step tutorial in the new version of Virtual Ray Tracer, displaying the "Help Menu" task of the first level.

   current level and task number. The progress bar shows the current progress in both visual and textual form. The step-by-step tutorial also contains buttons to go to the previous/next level and to go to the previous/next task. As we can see, most of the buttons are in a disabled state. This is the first level, so going back is not possible. The mandatory tasks for this level have not been completed yet, so we cannot go to the next level either. We are also not allowed to go to the next task yet as we first need to complete the task mentioned in the task description.

   Tasks for each level are stored in a list along with helper functions to, for example, increment the current level's task index. These tasks are then stored inside the Game Manager, so we can store the progress for each level when switching between levels. This way, the Tutorial Manager is also able to retrieve the current task information and update the tutorial's UI accordingly.

   Whenever a user has to complete a task by executing an action, we need some way of associating this action with the task. We do this by adding the same identifier to the task and the action. When a user executes a specific action (i.e. clicks the "Help" button), we send this identifier to the Tutorial Manager. If the identifier matches the current task's identifier, the task is completed and the UI is updated.

   We also added a free/cheat mode option that allows the user to skip all tasks and levels. The option was mainly added for development purposes but might also be interesting for all other users of VRT.

2. **Points and unlockable items**

   Points are earned by completing the tasks in the step-by-step tutorial. They are displayed in the tutorial's UI on the bottom left. We can see that in Figure 8 points are

displayed with a light bulb icon and that 181 of them have been accumulated so far. Each task has a number of points associated with it. On a task's completion, those points are added to the total number of points and stored inside the Game Manager.

Inside the sandbox level users are able to create a number of objects themselves. However, they are locked until enough points have been obtained. To lock/unlock these objects, we simply check the number of points that have been earned and lock/unlock the objects accordingly. As can be seen in Figure 9, the object creation menu has been modified to disable locked objects and display the number of points that need to be earned in order to unlock them.

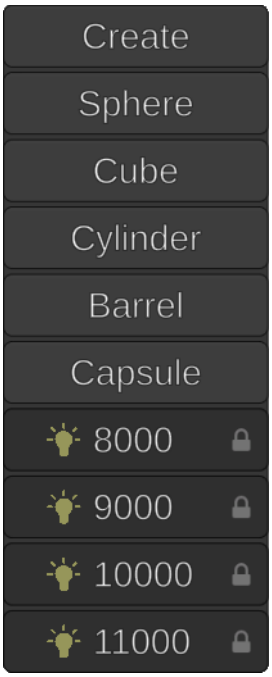
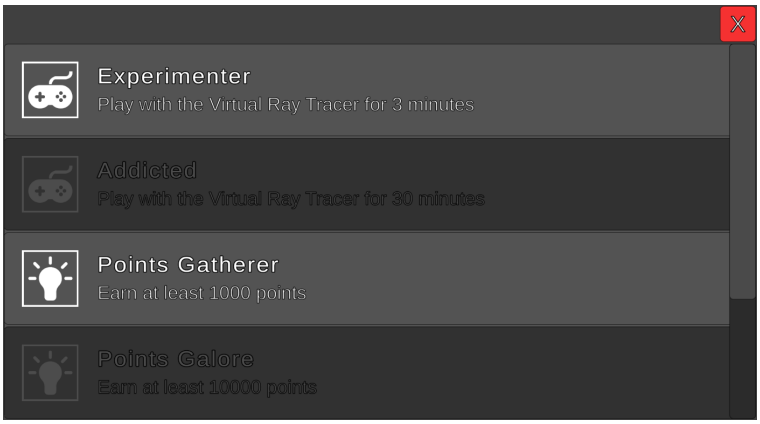


Figure 9: The modified object creation menu with some objects being locked, showing the number of points required to unlock them.

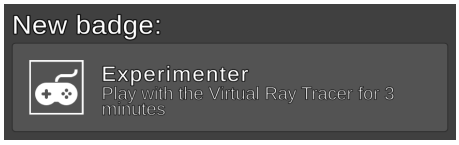
3. **Badges**

   Badges are added to the Main Canvas in two ways. There is a badge collection menu that displays all the badges (including badges that are not yet earned), and there is a badge notification menu that shows a notification whenever a new badge has been earned. Figure 10 illustrates both menus. They have one thing in common, which is that they both display one or more badges. So, we are able to create a single badge as a Prefab and use it for both purposes.

   Badges are stored in a list in the Game Manager with each badge containing a type, name, description, icon, and a number that determines when a badge is earned. There are three types of badges: playing with VRT for a minutes, earning b points, and creating c objects.



(a) Badge collection



(b) Badge notification

Figure 10: Badges in the new version of Virtual Ray Tracer, being displayed in two different ways.

In the badge collection menu, we simply create all available badges listed in the Game Manager's badge list. A badge that has not been earned yet is indicated by a disabled-looking state and barely readable text. In the badge notification menu, we continuously loop over each badge and check whether they have been earned since our last check. If so, the user receives a notification displaying the newly earned badge.

4. **Sounds**

   The difficult part of implementing sounds is finding or creating the sounds itself. Fortunately, we were able to find a button click and badge notification sound in the public domain. With some modifications, they were implemented into VRT.

   The sounds are added as objects in the Main Canvas. Each event that requires a sound to be played makes a reference to either one of the sound objects and invokes the play method. These references allow us to easily modify one of the sounds for all events at once instead of having to modify the sound for each event.

5. **Animations**

   Animations are easily added to VRT using `Unity`'s built-in animation system. We implemented two animations: a level transition animation and a badge notification popup animation. The level transition animation is a simple fade-out to black and fade-in from black. Before a new scene is loaded, we start the fade-out animation. After a scene is loaded, we start the fade-in animation. The badge notification animation simply comes up from the bottom of the screen when a new badge has been earned, stays visible for about five seconds, and then moves out of the screen.

5 User Study

In this chapter we explain the setup behind the user study and its results. The full list of questions from the user study can be found in Appendix A.

5.1 Questions

The goal of the user study was to determine whether a gamified version of Virtual Ray Tracer resulted in a more entertaining but still educational application. We expected to recruit about 10–20 participants for the user study. We made a web build for both the original and new version of VRT. This way, the participants could easily test out both versions without having to download any software. They were instructed to play around with both versions for about 10 minutes, while keeping track of their time, and then answering some questions. The cheat/free mode option was completely removed from the web build to limit the number of paths a user could take, resulting in more meaningful results.

5.1.1 Background questions

In order to retrieve some background information of the participants, they were asked what they are/were studying at their highest education level, to rate their skill with computers on a scale from 1 to 10 and, optionally, their age.

5.1.2 Technical questions

Participants were asked whether they had any technical difficulties, how much time they spent in both versions, and the differences they noticed. These questions helped rule out any participants that had technical difficulties with either one of the versions or did not take the user study seriously.

Furthermore, it might show that participants spend more time in one version than the other, which could indicate one version being more entertaining.

5.1.3 Educational questions

Participants were asked to indicate their understanding of ray tracing before and after playing with both versions of VRT. If the participants learned something about ray tracing, they were asked which version helped them understand ray tracing better. This helps determine whether the new version is worse, the same, or better at educating users in ray tracing.

5.1.4 Entertainment questions

Participants were asked which version they found easier, more fun and preferred overall to help determine whether the new version is more entertaining. In addition, we asked questions regarding the freedom of the new version compared to the original version to determine whether offering a free mode would be beneficial to the application.

5.1.5 Feedback

Lastly, participants were asked what they do or do not like about the new version compared to the original. They were also offered the option to leave any other feedback or general marks. This feedback could later be used to help improve the application further.

5.2 Results

The user study had a total of 39 participants. The participants were split into two groups. One group of 6 participants, group A, was gathered from friends and family, and the other 33, group B, was gathered via SurveySwap[1]. Unfortunately, of those 33 participants from group B, 9 had to be removed as they did not follow the provided instructions properly. This left us with 6 participants from group A and 24 participants from group B, so 30 participants in total. Table 1 shows a summary of the results of the user study's multiple choice questions.

Almost all users noticed the step-by-step tutorial in the new version. Only a few users noticed the addition of points and badges, and no one noticed either the added sound or animations. On average users spend 9 minutes and 10 seconds in the original, and 11 minutes and 31 seconds in the new version. Group A mostly had participants that were computer scientists and were familiar with ray tracing beforehand. Group B had participants of very differing educations, most of them not being familiar with ray tracing. The results were mostly similar between the two groups, so when we are talking about the results, we are talking about the combined results of group A and group B, unless specifically mentioned otherwise.

On the technical side, the percentage of users that experienced issues decreased from 37% in the original version to 20% in the new version. The decrease was mostly due to people better understanding of what to do in the new version. The users that still experienced issues in the new version were mostly having trouble going to the next level, as they were used to being able to immediately move on to the next level in the original version.

Looking at the results for the educational questions, we notice that 73% of the users preferred the new version in understanding ray tracing better, while only 13% preferred the original version in understanding ray tracing better. The rest of the users had no preference or did not understand ray tracing any better.

This same trend, of people preferring the new version over the original version, continued with the entertainment questions as well: 77% found the new version to be easier to use, 63% found the new version to be more fun to use, and 77% preferred the new version overall. Users that did prefer the original version in these questions were, as mentioned before, having trouble going to the next level or preferred the freedom of the original version. Interestingly enough, group A and group B were divided on whether there should be a free mode option in the new version, 33% versus 83% respectively.

The feedback showed that people really liked the step-by-step tutorial, especially as beginners. It also explained why group A was more against a free mode option compared to group B. The users in group A found that they discovered more features/concepts of VRT in the

[1] https://surveyswap.io/

Questions	Allowed responses	Group A	Group B
Did you experience any issue(s) while playing around in the original version?	Yes.	2	9
	No.	4	15
Did you experience any issue(s) while playing around in the new version?	Yes.	2	4
	No.	4	20
Did the application(s) help you understand ray tracing better?	Yes.	2	16
	Slightly, I already understood ray tracing beforehand.	2	2
	No, I already understood ray tracing beforehand.	1	0
	No.	1	6
Which version helped you understand ray tracing better?	Original version.	0	4
	New version.	5	17
	Neither.	1	3
Which version was easier/simpler to use?	Original version.	0	5
	New version.	5	18
	Neither.	1	1
Which version was more fun to use?	Original version.	0	9
	New version.	5	14
	Neither.	1	1
Did you prefer the freedom of the original version or did you prefer the more restricted/ guided experience of the new version?	I preferred the freedom of the original version.	0	5
	I preferred the restricted/guided experience of the new version.	5	16
	I have no preference.	1	2
	Other.	0	1
Would you have liked a free/cheat mode option that allowed you to skip tasks/levels in the new version?	Yes.	2	20
	No.	4	4
If you preferred the freedom of the original version, would you this free/cheat mode option in the new version change your preference?	I would still prefer the freedom of the original version.	0	1
	With a free/cheat mode option, I would prefer the freedom and restricted/guided experience of the new version.	0	13
Which version did you prefer overall?	Original version.	0	5
	New version.	5	18
	Neither.	1	1

Table 1: A summary showing the amount of responses for the multiple choice questions from the participants of group A and B.

new version and that with this option, these features could be missed. Moreover, one user remarked that the free mode option would not be necessary if the progress was saved, as there would be no more restrictions after completing the tutorial for the first time.

Other users suggested that the tutorial could be further improved by adding visual indicators helping the users complete the tasks.

6 Conclusion

The goal of this bachelor thesis was to improve the original Virtual Ray Tracer by making it more entertaining while still keeping it educational using gamification. We conceptualized and then implemented several modifications to achieve this. Those modifications include the addition of a step-by-step tutorial, points and unlockable Items, badges, sounds, and animations.

The results of the user study showed that these modifications were mostly successful in achieving our goal. The feedback received from the participants is pointing towards the step-by-step tutorial being the most important modification in achieving this goal. The other modifications were hardly noticed but might still have had a positive influence on the experience in the new version of Virtual Ray Tracer. More specifically, on the educational aspect of the application we surpassed our goals. 73% of the participants preferred the new version in helping them understand ray tracing better, while the goal was to keep the educational aspect at least the same. Entertainment wise, 77% found the new version easier to use and 63% found the new version more fun to use, which was lower than expected. However, most of the users that did prefer the original version in these aspects were frustrated by the (lack of) freedom in the new version compared to the original version. Participants complained they could not move forward to the next level, as in the original version. We believe that new users, with no prior experience with the original version, would not have experienced those issues, resulting in a more positive result.

The user study also shows that 73% of the participants would have liked a free mode option to allow skipping tasks/levels in the new version. However, this does not necessarily mean that this option should be added. The participants in the user study (almost) all first played with the original version before playing with the new version, which might have skewed the results. If they first played with the new version and then played with the original version, or had never experienced the original version at all, the results might have been completely different. Moreover, if the progress is saved after exiting the application, the application would offer full freedom after completing the step-by-step tutorial for the first time. Unfortunately, as of right now, the progress is not saved. So, we believe it would be best to leave this option (kind of) hidden away in the settings menu for now.

7 Future Work

This chapter discusses some suggested future improvements to the gamified version of Virtual Ray Tracer:

- **Progress saving** Currently all progress is lost after exiting the application. However, saving the progress would have the benefit of users being able to split their learning sessions, and would replace the need for a free mode option. We would suggest to add options to save and load to/from a file. This way the web build would also allow saving/loading of the progress.
- **Visual assistance** The addition of a visual component to the tasks in the step-by-step tutorial would make it even easier for new users of the application. This could be implemented by adding red circles to show where to click and virtual keys to show which buttons to press. Implementing this could be quite challenging however, as some tasks require the user to be in a specific menu before being able to click a specific interface element.
- **Controls** Panning and orbiting the camera is slightly difficult for beginners in VRT due to the unusual controls. Additionally, there are two ways for panning and orbiting the camera. This could be improved by choosing only one set of controls and making those controls more user-friendly.
- **Help menu** The information provided in the help menu is a bit outdated compared to the information provided in the step-by-step tutorial. Making the help menu more consistent with the step-by-step tutorial would be a nice improvement.
- **Animations and sounds** We were only able to implement a few animations and sounds. Adding more animations and sounds could improve the application even further. Some example events where animations and sounds could be added are when: receiving points, completing tasks, completing levels, etc.

Additionally, there are also four other students working on VRT: Jesper van der Zwaag is working on adding distributed ray tracing; Roan Rosema is working on mobile and web versions of VRT; Bora Yilmaz is working on adding support for acceleration data structures; Anton Bredenbals is working on adding support for ray marching. Our projects should be merged and new tutorial tasks will need to be added.

8 Acknowledgements

I would like to thank my supervisors Jiří Kosinka and Steffen Frey for helping me throughout the project by answering questions and providing great feedback. Additionally, I would like to thank Chris van Wezel for providing resources and answering questions. Lastly, I would like to thank my family and friends, as well as the users from SurveySwap, for taking part in the user study.

References

[1] C. van Wezel, "A virtual ray tracer," Bachelor's Thesis, University of Groningen, 2022. [Online]. Available: https://fse.studenttheses.ub.rug.nl/26455/.

[2] W. Verschoore de la Houssaije, "A virtual ray tracer," Bachelor's Thesis, University of Groningen, 2022. [Online]. Available: https://fse.studenttheses.ub.rug.nl/24859/.

[3] O. Goethe, "Gamification mindset," ser. Human-computer interaction, Cham, Switzerland: Springer, 2019, ISBN: 9783030110772. DOI: 10.1007/978-3-030-11078-9.

[4] C. Gribble, J. Fisher, D. Eby, E. Quigley, and G. Ludwig, "Ray tracing visualization toolkit," in *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, ser. I3D '12, Costa Mesa, California: Association for Computing Machinery, 2012, pp. 71–78, ISBN: 9781450311946. DOI: 10.1145/2159616.2159628.

[5] N. Vitsas, A. Gkaravelis, A.-A. Vasilakis, K. Vardis, and G. Papaioannou, "Rayground: An Online Educational Tool for Ray Tracing," in *Eurographics 2020 - Education Papers*, M. Romero and B. Sousa Santos, Eds., The Eurographics Association, 2020, ISBN: 978-3-03868-102-1. DOI: 10.2312/eged.20201027.

[6] B. Hennessey, S. Moran, B. Altringer, and T. M. Amabile, "Extrinsic and intrinsic motivation," in *Wiley Encyclopedia of Management*. John Wiley & Sons, Ltd, 2015, pp. 1–4, ISBN: 9781118785317. DOI: 10.1002/9781118785317.weom110098.

[7] E. L. Deci and R. M. Ryan, "Self-Determination theory," in *Handbook of Theories of Social Psychology: Volume 1*. 1 Oliver's Yard, 55 City Road, London EC1Y 1SP United Kingdom: SAGE Publications Ltd, 2014, pp. 416–437, ISBN: 9780857029607. DOI: https://doi.org/10.4135/9781446249215.n21.

[8] T. Reiners and L. C. Wood, "Gamification in education and business," in Cham, Switzerland: Springer, 2015, ISBN: 9783319102078. DOI: 10.1007/978-3-319-10208-5.

[9] S. Nicholson, "Strategies for meaningful gamification: Concepts behind transformative play and participatory museums," *Meaningful Play*, vol. 1999, pp. 1–16, 2012.

A User Study Questions

Q1: What are/were you studying at your highest level of education? (Current studies apply)

Q2: On a scale from 1 to 10, how skillful would you consider yourself to be with computers?
R: 1–10, with 1 being not skillful at all and 10 being very skillful.

Q3: How familiar were you with ray tracing before this survey?
R: 1–5, with 1 being not familiar at all and 10 being very familiar.

Q4: How much time (in minutes) did you spend in the original version?

Q5: How much time (in minutes) did you spend in the new version?

Q6: What differences did you notice between the two versions?

Q7: Did you experience any issue(s) while playing around in the original version?
R: Yes, No.

Q8: If you answered Yes to the previous question, what issue(s) did you experience? (optional)

Q9: Did you experience any issue(s) while playing around in the new version?
R: Yes, No.

Q10: If you answered Yes to the previous question, what issue(s) did you experience? (optional)

Q11: Did the application(s) help you understand ray tracing better?
R: Yes, Slightly I already understood ray tracing beforehand, No I already understood ray tracing beforehand, No.

Q12: Which version helped you understand ray tracing better?
R: Original Version, New Version, Neither.

Q13: Did you prefer the freedom of the original version or did you prefer the more restricted/guided experience of the new version?
R: I preferred the freedom of the original version, I preferred the restricted/guided experience of the new version, I have no preference.

Q14: Would you have liked to have a free/cheat mode option that allowed you to skip tasks/levels in the new version?

R: Yes, No.

Q15: If you preferred the freedom of the original version, would this free/cheat mode option in the new version change your preference? (optional)

R: I would still prefer the freedom of the original version, With a free/cheat mode option I would prefer the freedom and restricted/guided experience of the new version, Other (fill in).

Q16: Which version was easier/simpler to use?

R: Original Version, New Version, Neither.

Q17: Which version was more fun to use?

R: Original Version, New Version, Neither.

Q18: Which version did you prefer overall?

R: Original Version, New Version, Neither.

Q19: What do/don't you like about the new version compared to the original version?

Q20: If you have any other feedback or general remarks, you may leave them here (optional)

Q21: What is your age? (optional)