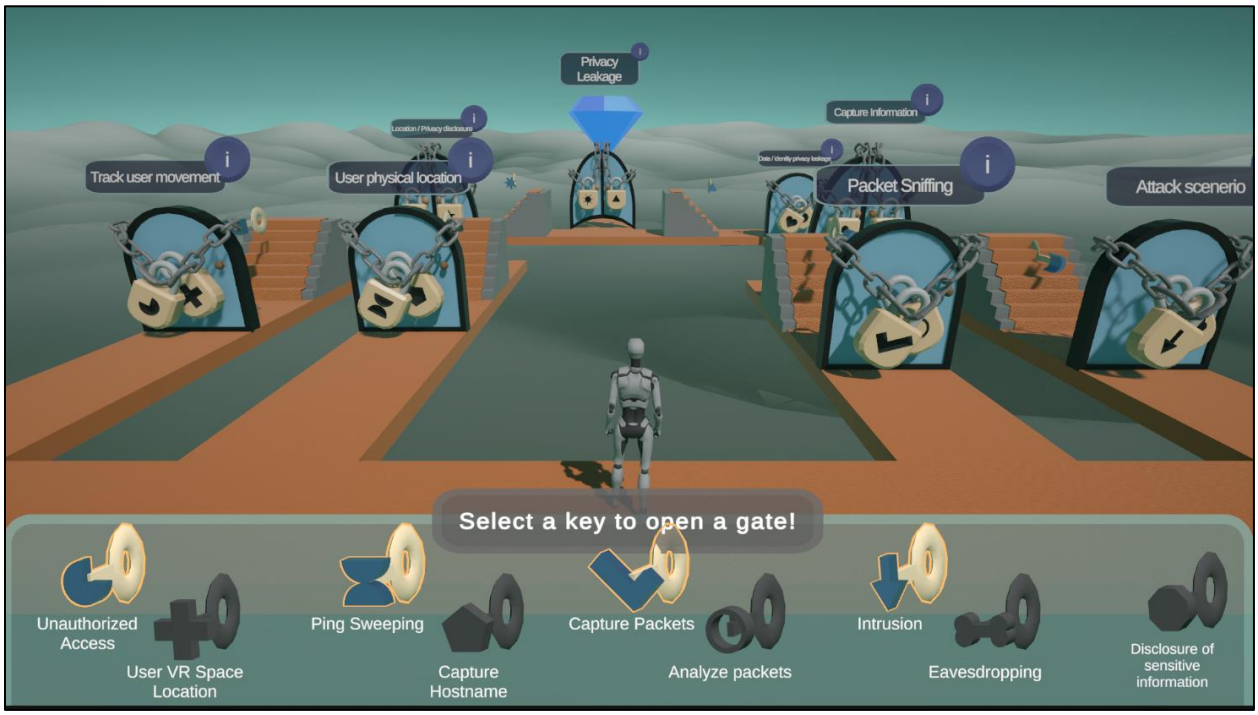
Making Qualitative Attack Trees Accessible: A Gamified Interactive Visualization for Non-Expert Stakeholders



Written by: Casper ten Holder

Supervisor: Prof. Dr. M.I.A. Stoelinga

Critical observer: M.A. Lopuhaä – Zwakenberg

Date: February - July 2024

Abstract

This thesis explores the impact of interactive and gamification elements on effectively communicating qualitative attack trees to non-expert stakeholders. The challenges identified that possible hinder the effective explanation of attack trees include: a lack of contextual information, the use of technical jargon, insufficient emphasis on risk severity (such as minimal attack sets), and an overwhelming structure of attack trees that can lead to cognitive overload. An engagement framework was developed to guide design choices that enhance the effective communication.

To address these challenges, a gamified interactive visualization was created. The analogy of locks on doors that must be opened with keys is used to help users map this familiar relationship onto the various logic gate material in the attack tree. This design incorporated various gamification elements, such as a leaderboard for displaying minimal attack sets, a challenge to find the minimal attack set by selecting keys for doors, UI elements providing instant feedback, a visually appealing environment, and a movable character to boost engagement and understanding among non-expert stakeholders.

The results showed that the experimental group had a better understanding of attack sets compared to the control group, and participants found the prototype engaging. Some design flaws were identified, suggesting areas for improvement to further enhance engagement.

Acknowledgements

I would like to extend my deepest gratitude to my supervisor, Prof. Dr. M.I.A. Stoelinga, for her support, guidance, and encouragement throughout my project. Her feedback and energy were valuable in the completion of this graduation work. I would also like to thank dr. M.A. Lopuhaä – Zwakenberg for his valuable feedback on the design of the prototype.

I also want to express my thanks to my parents, Olaf ten Holder and Ragna ten Holder. Their patience in listening to me talk about my project has been a source of inspiration. Without their support, this achievement would not be as valuable as it is now.

Thank you all for believing in me and for your support.

1 Contents

2 Introduction

In 2023, the Royal Dutch Football Assocation (KNVB) was attacked by the Russian hacker group Lockbit and stole sensitive data such as passports and medical, with an estimated worth of approximately 1 million euros of data [1]. To prevent fatal consequences of such events, risk assessment models are developed by engineers. A popular risk models used in security risk analysis are attack trees. Attack trees show how a target might be attacked, and it can be displayed as multi-levelled diagram, a tree, consisting of root, child nodes and leaves. An attack is considered successful when a complete path can be made from one or multiple leaves, through the children nodes' conditions, up to the root.

The challenge of these expert-based models is, however, that they are not always understood correctly by non-expert stakeholders. The objective of this research is to design an interactive visualization of attack trees which enhances the effectiveness of communicating this risk model to non-expert stakeholders.

Having a high engagement and motivation of stakeholders in the explanations of attack trees can contribute to stakeholders further exploring the meaning of the model. This is crucial because as different stakeholders gain a better understanding, it can help them to improve their decision-making and policy [2]. However, there is not a singly universally accepted definition of user engagement, specifically in the context of attack trees. Therefore, it is important to define it so that it can be measured. This thesis proposes that user engagement in the context of attack trees can be seen as three-step process, each step influenced by various indicators such as visual appeal, the level of feedback, and an optimally challenging problem to solve.

Effective communication also involves verifying that the non-expert stakeholders understand attack tree being explained. Attack trees model may suffer from various possible problems hindering the explanation. They possible lack contextual information, such as how specific node types operate, use technical jargon that is difficult for non-experts to grasp, lack to communicate the risk severity of specific elements in the tree and can possibly cause cognitive overload in which the user has to process too many elements to build an understanding. Addressing these challenges can possibly lead to more effective communication.

Gamification elements may be applied during the design of the interactive visualization as they seem to improve the engagement and motivation of stakeholder [3]. Careful consideration should be made when applying certain elements of gamification as they can cause undesirable side-effects to the learning process of the users.

This research begins with the background on the attack trees. Consequently, it is explored in more detail what the understanding challenges of attack trees are in risk assessment communication. Then, it continues to a more detailed definition of user engagement in this context means and which factors play a role in enhancing this engagement. Based on this definition and the nature of an attack tree, different gamification elements can be identified that appears to be relevant for the visualization. The ideation follows after the identification of these challenges of attack trees, concluding the final chosen idea of a digital visualization utilizing the analogy of a burglar breaking into a house. The logic gates are transformed into the metaphor of doors and locks, with the purpose of making it more understandable to stakeholders. Finally, the effectiveness of the design is user tested on non-expert stakeholders.

2.1 Research Questions

The objective of this research is to design an interactive visualization of attack trees which enhances the effectiveness of communicating this risk model to non-expert stakeholders. Therefore, the research question can be formulated in the following way: *How can the effectiveness of communicating qualitative attack trees to non-expert stakeholders be improved through the integration of gamification elements?*
The effectiveness of the design will be considered better if non-experts both understand the design better and have more engagement compared to original communication of this risk model.

To answer the main research question, several sub questions could be formulated:

- What are the challenges in communicating risk assessment models?
- How can user engagement be defined in the context of Attack trees?
- Which gamification elements and other aspects can be applied to improve the user engagement?

An attempt will be made in chapter 2 to partly answer these questions and in the evaluation, it is verified whether the applied elements helped to improve the engagement and communication of ATs to non-expert stakeholders.

3 Background

3.1 Qualitative analysis of Attack trees

Attack Trees (AT) are a widely used notation in security analysis [12] and offer a systematic and convenient approach to categorize the different ways the system can be attacked [8] [9]. The structure of an Attack tree should be identified before an interactive visualization can be deployed.

The structure of ATs can be represented as a tree with the root node, usually defined as the *top event*, being the ultimate goal of an attacker [8][9][10][65]. Several approaches to reaching that objective are outlined by basic attack steps (BAS), shown as leave nodes in the tree. The nodes between the BASs and the top event serve as subgoals to reach the goal. The children of these subgoals illustrate the different method to achieve that subgoal. These subgoals are represented as logical AND, OR and SAND gates [8][9][10][11]. In essence, a SAND operator is an AND gate that in which events occur. In Figure 1, it can be seen how these operators and nodes are graphically outlined in ATs.

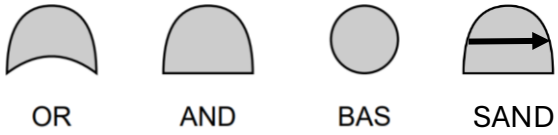


Figure 1: The logic gates used in Attack Trees [10] [11].

ATs can be used to analyse information in multiple ways. Examples are *quantitative [48], qualitative [8] and cost-damage analysis [10]*. The intention of a *qualitative* analysis is to identify the combinations of elementary events that lead to the top event, while quantitative analysis focus on how attacks may be carried out with a different probability and other metrics [8]. These can also include the damage of the attack and the required costs for the attacker [10]. Cost-damage analysis specifically analyses the interplay between the cost and damage metrics [10]. In this research, the focus emphasizes on qualitative analysis of Attack trees.

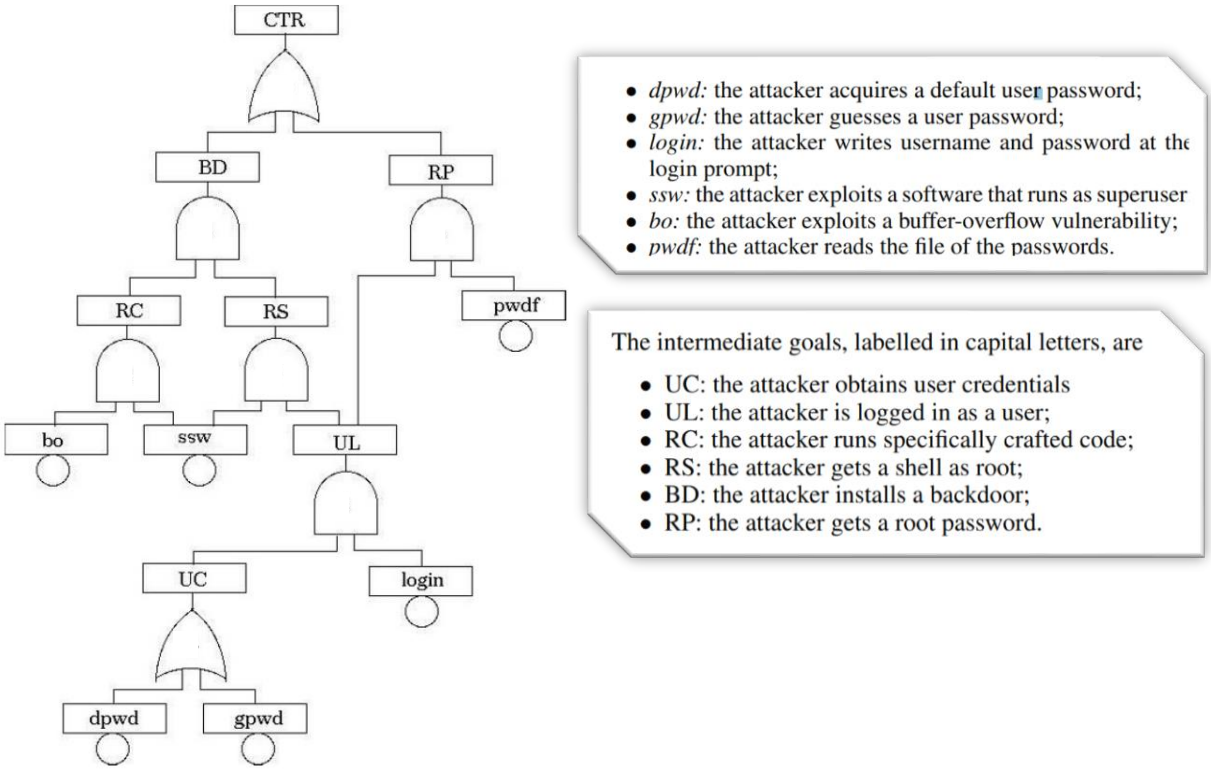


Figure 2: An Attack Trees representing an analysis about the security of Unix Server [8] The OR-gate are represented with a more pointy tip. The other gate represent an AND-gate instead of an Sequential-AND gate.

For stakeholders to improve their policy, identification of a list of minimal combinations of BASs that lead to the final objective of an attacker can reveal where a security system is the most vulnerable. A single combination of BASs in this list are defined as a minimal attack set [8]. The order or importance of a minimal attack sets is defined by a number of BASs that must be true to reach the final goal. When reanalysing the AT in Figure 2, it can be seen that there are 4 minimal combinations of BASs that reach the top event. These are the sets *{bo ∧ ssw ∧ dpwd ∧ login}, {bo ∧ ssw ∧ gpwd ∧ login}, {dpwd ∧ login ∧ pwdf}, and {gpwd ∧ login ∧ pwdf} [8].* The last two sets have 3 BASs in their sets from which the attacker can reach the goal, while the first two have four. To summarise, the order of the last two sets has an order of 3 whereas the first two sets have an order of 4. In qualitative analysis, the lower the order, the higher the importance. Therefore, from this analysis it would be beneficial for the policy makers to prioritize improving their system in the direction of the last two sets.

However, the key takeaway for the stakeholders would vary when identifying the differences in each type of analysis (e.g.: qualitative, quantitative) on ATs. For example, quantitative analysis can show that there is a small probability of a successful attack on these BASs in minimal attack sets. As a result, stakeholders should likely concentrate on other system weaknesses.

Overall, given that quantitative analysis is generally more complex than qualitative analysis, qualitative research is prioritized in this research in order to establish a foundation for future work on increasingly complex ATs.

3.2 Challenges in Risk Assessment Communication

Before the identification of fitting elements for the design, the causes for the miscommunication from engineers to non-experts about risk models should be explored. Analysing these communications problems can possibly improve the design requirements of the interactive visualization.

3.2.1 Engagement in risk communication

M.N. Ndlela [2] argues that risk stakeholders should continually participate and engage during the risk management process. One of the reasons this is important is because these stakeholders are in some manner affected by risks or strategies made revolved around risks. Moreover, stakeholders tend to change in the course of the process. Some stakeholders leave or join in different phases of the process and by continually engaging them, they feel that their interests are taken seriously. Besides, they gain a better understanding of risks management and feel involved in decision-making and actions that can affect their future [2]. The literature review section further explores engagement in the context of attack tree

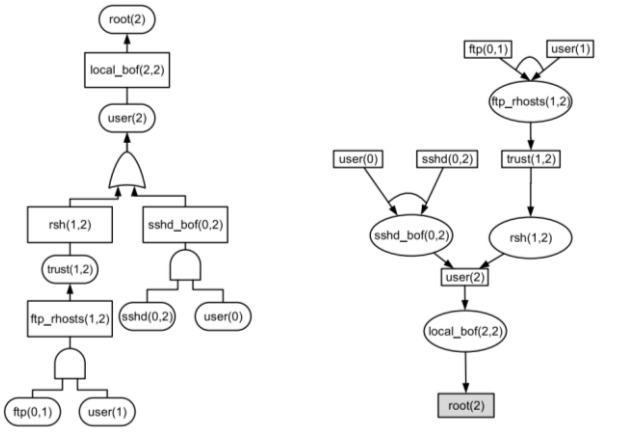
3.2.2 Lack of contextual information

Scientific ideas have a tradition to be communicated to the public as novel ideas standing on its own, without the provision of contextual information [4] [5]. Thus, it appears that the communication of risk models suffer from this lack of context as system risk assessment can be linked to science and reliability engineering. Although complex concepts, and attack trees can be seen as one, are already difficult to communicate, the communication of scientific information introduce more problems for the understanding [5]. Especially non-experts may create inaccurate assumptions and conclusions when these new structures are exposed to them. It requires them to discard those believes and that can be challenging [5].

Contextual information in attack trees that may be challenging to understand for non-expert can include node types, the functioning of logic gates, Parent-child relationships and probability and impact values (in quantitative or cost-damage analysis). For example, Prasetyono and Hariyono [28] found that less than 50% of informatics engineering students at the University of Peradaban understood logic gate material. Because of this, Lallie et al. [29] created a new visual structure of all nodes types in an adapted attack graph. For clarification, the researchers changed the symbol usage, reduced the number of symbols and utilised different representation of precondition logic. Thus, Lallie et al. made an attempt to slightly abstract the logic gates and node types. Due to their new structure, they suggest that fundamental differences in the syntax are likely to provide differences in the cognitive perception. Their adapted structure is illustrated in Figure 3. Even though, the study implies that the new structure could be suitable for aiding cyber-attacks perception, it failed to find a statistically significant result for this.

To help to aid the cognitive perception further, attack trees could use analogies and metaphors in their model. The reason to use analogies and metaphors is that it helps the non-expert to map familiar relationships onto the new material [31]. Besides a better understanding, it can also provide attention to the most prominent features [31]. This may be useful in attack trees to attract attention to the most vulnerable BAS,

possibly making it easier to identify the minimal attack sets. On the other hand, analogies are limited in terms of their instructional effectiveness. They can be misleading when the learner makes an attempt to abstract the metaphor so far that the relation between the metaphor and real concept is hardly visible [30].



Construct	Fault tree	Adapted Attack Graph
Exploit	rsh(1,2)	rsh(1,2)
Precondition	user(2)	user(2)
And		
Or		user(2)
Event Flow	Bottom to top	Top to Bottom

Figure 3: An adapted visual structure of an adapted attack tree [29]

3.2.3 Use of technical jargon

The use of technical jargon in risk communication can be challenging for non-experts to understand. This can create a barrier between the non-expert and the expert explaining their needs for making decisions on their policy. Ellermann et al. [32] found that non-experts had difficulties of understanding the risk assessments on food, feed, and other consumer products or chemicals. A part of the misunderstanding stems from the fact that their risk profile consisted of terminology such as “Acceptable and the Tolerable Daily Intake (ADI and TDI)”. Their recommendations were revised and concludes that the risk models need to reduce the technical terms. Instead of using (ADI and TDI), the term “health-based guidance value” was used to describe the limit of consumption of a certain substance. Similarly, Wu et al. [33] identified that non-expert participants had difficulty in understanding the technical terms, such as “DoS attack” and “Trojanized apps”, of the articles related to (cyber) security.

This phenomenon of technical jargon can also be related to Attack trees. Let’s look at the example of the technical terms in figure 4. According to this figure, it might be difficult for the user to understand what the description inside the various nodes means. For example, the description of *Sshd_bof(3,2)* may cause challenges in the understanding of the user.

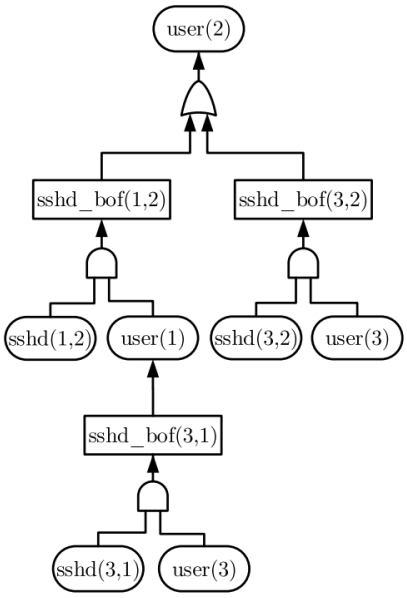


Figure 4: Attack tree with technical jargon [29]

Therefore, Wu et al. [33] developed a pop-up system that explains these terms to the participants without IT background. This is shown in Figure 5. The explanation of a term would be revealed when the user hovered on these terms. According to them, this greatly improved the users' security understanding.

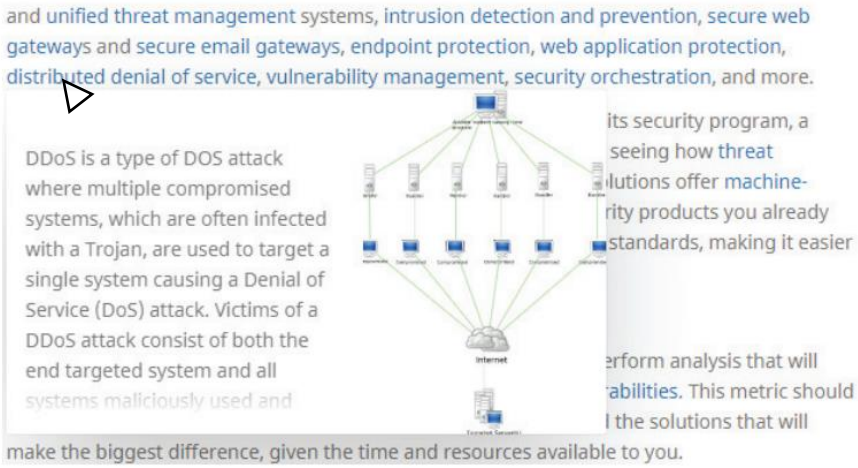


Figure 5: The term 'distribution denial of service' is explained in the pop-window.

To conclude, the new design of attack trees should include an explanation of these technical terms to prevent confusion among non-expert stakeholders.

3.2.4 Risk severity

Another possible problem with the communication of attack trees is understanding the general purpose of risk assessments. Ellermann et al. [32]. developed a risk profile prototype for communicating the severity of a health impairments of consumer raw milk to non-experts. The non-experts did not understand the reason for developing risk

assessment when there was a high level of uncertainty in the data. Furthermore, it was difficult for them to think of use cases or scenarios outside their developed risk profile.

The unclarity and uncertainty may also be more specifically related to attack trees than only general risk assessments models. However, in qualitative analysis of attack trees, uncertainty in data, such as probability of an occurrence of a basic attack step, or the uncertainty of the attack costs and damages, are only considered in other analyses (e.g.: in quantitative analysis). The general purpose of a qualitative attack tree may not be clear to the user. In this case, the purpose could be described as the provision of the most vulnerable nodes in the attack tree that needs reinforcement. Making the risk severity of the most vulnerable nodes clear can possibly aid the understanding of attack trees.

3.2.5 Cognitive overload

Risk assessment models could potentially suffer from cognitive overload. There is no general definition for cognitive or information overload. The term usually refers to when the user has a difficulty in understanding a concept when the user must process a comprehensive quantity of information [34, 35]. This comprehensive quantity of information can cause distraction from flow by when the user is faced with continuous interruptions and challenging decisions [34, 35].

Individuals usually apply different mechanisms of coping with cognitive overload. First, individuals tend to accelerate the rate at which information is processed and, but this is very cognitively demanding [36]. Second, individuals use filtering techniques to absorb only the most important information while ignoring less important information [36].

A solution to information overload is to give individuals control over the information environment [34]. To further clarify, using certain information technology techniques, such as filtering and avoidance of delivering information to users without request or control, can often offer solutions to this overload [34]. For example, a filter button in online web shops to only show male or female clothing possibly helps lowering the cognitive load.

In the context of risk assessment, information overload can be apparent. For example, Pennington and Brad [36] conducted research in which participants had to make a decision based on nine given risk factors of software development project. One of the outcomes were that the participants rarely filtered out information, which implies that the participants were unable to process all information. Potentially, this overload can also become visible in attack trees. As it can be seen in figure 6, many nodes and BASs are presented at the same time, which possibly makes it challenging for users to come to direct conclusion about the minimal attack sets and most vulnerable nodes in the system. Incorporating user control of the information can possible aid reducing the cognitive load process for users.

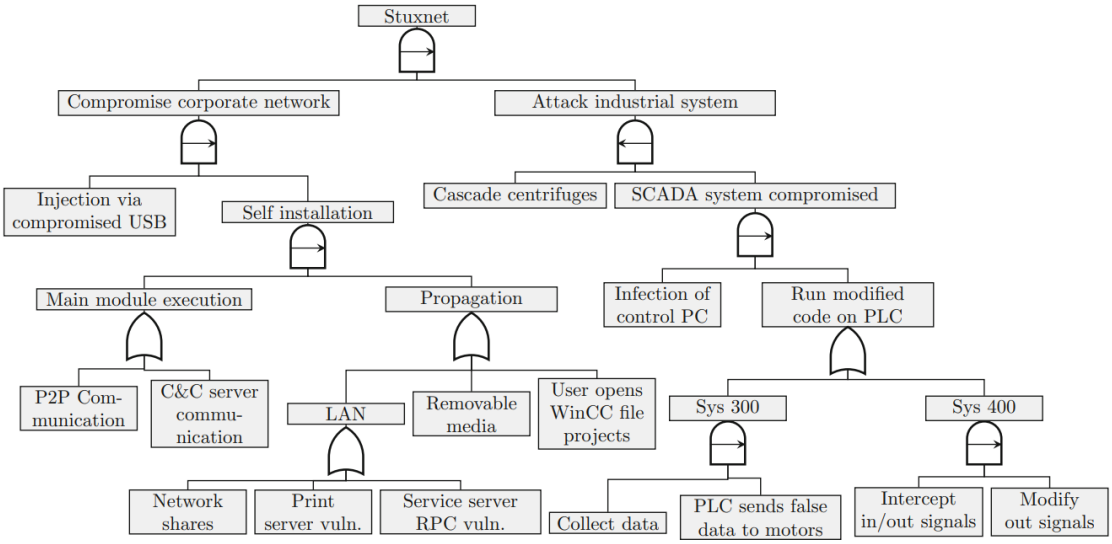


Figure 6: a part of the Stuxnet attack tree [41]. The arrow in the Sequential AND-gate defines in which order the events occur.

3.2.6 Conclusion and Discussion

In the previous section, challenges have been identified regarding the communication of risk models. From this, it could be argued that there are challenges in the engagement, lack of contextual information such as the representation of logic gates, technical jargon, the risk severity and general purpose of attack trees, and the cognitive overload. Nevertheless, these challenges in communication should be taken lightly as no direct research proposes connection of these challenges to attack trees. Some assumptions have been made about difficulties of attack trees that may or may not be true. Therefore, these challenges are used as guideline rather than a strict rule that should be fixed in the interactive design of attack trees.

3.3 Literature review

3.3.1 Introduction

In 2023, the KNVB was attacked by the Russian hacker group Lockbit and stole sensitive data such as passports and medical, with an estimated worth of approximately 1 million euros of data [1]. To prevent fatal consequences of such events risk assessment models are developed by engineers. A popular risk models used in security risk analysis are Attack Trees. Attack trees show how a target might be attacked, displayed as multi-levelled diagram: a tree, consisting of root, child nodes and leaves. An attack is considered successful when a complete path can be made from one or multiple leaves, through the children nodes' conditions, up to the root. The challenge of attack trees and other expert-based risk models is, however, that they are not always understood correctly by non-expert stakeholders.

Having a high engagement of stakeholders in the explanations of attack trees can contribute to stakeholders further exploring the meaning of the model. User engagement, which is a category of user experience, is the energy and effort that users employ towards technology, observable through different indicators [15] [16]. This active engagement is important because it can help them to improve their decision-making and policy as different stakeholders gain a better understanding [2]. Gamification elements may be applied during the design of the interactive visualization as they seem to improve the engagement and motivation of stakeholders [3]. Thus, the goal of this review is to research how various gamification elements could be utilized to Attack trees to significantly enhance the engagement of non-expert stakeholders.

The first section of the review strives to define what user engagement is in the context of Attack trees. The second section will explore which gamification elements are most commonly applied to a concept in an education context. Lastly, it is discussed how these elements could be adopted in attack trees.

3.3.2 The complexity of defining user engagement in the context of Attack Trees

It is important to provide a definition of user engagement in the context of attack trees so that it can be measured. As a result of these measurements and understanding, the design of attack trees can be altered to maximize user engagement. However, there is no single universally accepted definition of user engagement. This is because the definition of user engagement is often recognized as a complex multidimensional concept and varies based on contextual factors, leading to different perspectives within the literature [13-16]. Nevertheless, there are two main similarities of the definition of user engagement: it is often conceptualized as a process within a framework [13, 15, 16] and considers various indicators and characteristics, often described as dimensions within the framework [13-15]. In the next section, it is explored how to define these dimensions and how to shape these dimensions into process-framework for attack trees to finalize the definition.

3.3.3 The dimensions of user engagement in the context of attack trees.

User engagement is commonly viewed as having multiple dimensions.

The first dimension of user engagement is behavioural engagement. According to Bond and Bendelier [14], "behavioural engagement relates to participation, persistence and positive conduct" from the user towards the concept. However, this definition is

generally expanded on their context and provides practical implication based on their attributes [13, 15, 16]. In the context of Oh et al.'s [13] interactive media, behavioural engagement entails the physical interaction with the medium, such as swiping and scrolling, as well as the digital outreach by sharing or bookmarking the content. To illustrate their similarities, physical interaction can be seen as participation, and digital outreach as persistent, because the user is persistent with media by sharing or bookmarking content. Similarly, Heller et al. [15] only resonates with the digital outreach in their context, but combines it as reuse likelihood and word-of-mouth (WOM) intentions to others about the technology. As an interactive visualization is made of attack trees, it seems reasonable to resonate with Oh et al.'s definition due to their similar context. Therefore, behavioural engagement in the context of attack trees relates to exploration of attack tree nodes and the various ways of interacting with the attack tree. A small portion involves sharing insights about the attack tree with others, as this task falls under the responsibility of the modeler who explains the attack tree, rather than being the task of the visualization itself.

The second dimension of user engagement is psychological engagement. This dimension is usually divided into two parts: cognitive and affective engagement [13 – 15]. According to Bond and Bendelier [14], the definition of cognitive engagement relates to "deep learning strategies, self-regulation and understanding". While these terms seems relevant for the definition, Heller et al. [15] further elaborates on this definition, suggesting that cognitive engagement is a sense of heightened perception and rationality within their context of Augmented Reality. In the context of Attack trees, it seems reasonable to assume that cognitive engagement is the heightened perception and understanding of the structure of the attack tree.

Affective engagement in the context of educational technology relates according to Bond and Bendelier [14] to "positive reactions to the learning environment, peers and teachers, as well as their sense of belonging and interest". However, Heller et al. [15] criticize this definition in their context by describing it as emotional engagement, which means that the technology enables emotional connections, leading to positive feelings towards it. In conclusion, the definition of Heller el al. [15] is taken for this review due to its practicality in their definition. Both Heller et al. [15] and Oh et al. [13] provide ways to evoke this engagement for the creation of the framework.

3.3.4 A Process-Framework of user engagement for attack trees

A process of engagement considers different phases in time of these dimensions with the purpose of maximizing engagement. Frameworks of these processes differ from each other based on contexts, but there seems to be two points of agreements among these frameworks.

The process of engagement starts with a point of engagement, which is supported by different engagement attributes. According to O'Brien and Toms [16], the point of engagement begins with pleasing aesthetic elements, variety and novelty of the technology, and a motivation to accomplish a task. However, Heller et al. [15] only support visual appeal, but expand on it with the concept of information fit-to-task, which is a specific interactivity quality. Oh et al. [13] agrees both with Heller et al. and O'Brien and Toms, starting with the process with assessment which considers interactivity, novelty and ease of use, which is a characteristic of interactivity. Oh et al. and Heller et al. does not seem to include the other attributes from O'Brien and Toms as

they probably would not apply to their context. In conclusion, the point of engagement begins by the characteristics of visual appeal, novelty, interactivity and motivation.

The second phase and third stage of the process involve the two dimensions of engagement. In both the framework of Oh et al. [13] and of Heller et al. [15], the second stage begins with psychological engagement and ends the third stage with behavioural engagement. Oh et al. [13] advocates for not seeing these two stages a hard distinctive steps but rather as a gradual transition between the two. O'Brien and Toms [16] expands on the second stage of psychological engagement by discussing how psychological engagement can be stimulated. According to them, it can be stimulated with feedback, awareness, user control, positive affect, attention and the challenge to keep the user occupied. As mentioned earlier, behavioural engagement in our framework only considers further exploration of the attack tree, meaning that reengagement is not necessary.

All in all, contextual factors lead to different and diverse perspectives within the literature regarding the definition of user engagement. In the analysis of attack trees, user engagement will be considered as a process, starting with a stage of a point of engagement and flowing into the occupation of the user's attention. Figure 7 provides the user engagement process-framework for attack trees, including how the dimensions stand in this framework. With this, it can be determined how different gamification elements influence different engagement phases and attributes within the framework.

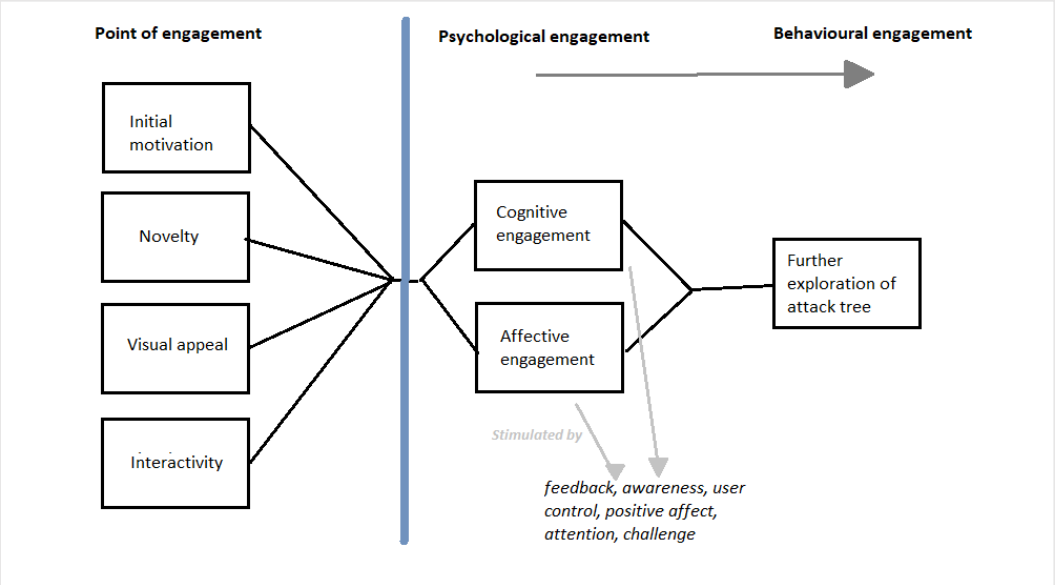


Figure 7: The user engagement process-framework for attack trees.

3.3.5 Application of gamification element in the context of attack trees

Gamification elements in literature are typically defined to improve the engagement. Selecting the the right element can be difficult as the list of gamification elements is comprehensive. Analysing all of them is beyond the scope of this review and therefore, a selection of analysed gamification elements has been made. The selection of elements includes leaderboard, points and badges. These elements were chosen based on the frequent mention in broad contexts. This is supported by both the reviews of Zhang et al. [17], Hamari et al. [18] and Dicheva et al. [19] who found that points, leaderboards, badges are the components receiving the most attention in various subjects.

Additionally, the relative ease of implementation makes them practical choices for integration into Attack trees [20].

Badges seems to lay at the beginning of the previously defined engagement process. A badge is reward provided when a certain milestone or challenge is achieved. In the context of attack trees, a badge could be achieved when the user finds out what the minimal attack sets of the attack tree are, which are the minimal basic events or vulnerabilities needed to reach the root of the attack tree. The reason that it is placed at the beginning of the engagement is process, is that they have some visual component [22] and they seem to positively influence the user's motivation to engage with the technology and complete the task [21, 23]. When considering the many attributes influencing engagement from O'Brien and Toms [16], it becomes evident that the motivation to complete a task lies inside the affective dimension of the 'point of engagement' phase, as described in our framework.

Points appear to increase the psychological engagement and stand in the middle of the engagement process of attack trees. Similar to badges, points are also rewarded when the user accomplishes certain actions. The difference between points and badges is that points continuously provide feedback to the user when completing certain activities [24]. By providing feedback, the users are more aware of what the system is doing, according to O'Brien and Toms [7]. Awareness of the system seem to align with the cognitive part of the psychological engagement dimension, as described in the framework from Figure 1. Like badges, points also have social aspects, such as user's status or reputation towards others [25, 27]. As a result, points introduce motivation aspect which resembles with the attribute in the 'point of engagement' phase. However, it may be that this influence is minimal, because providing feedback seems to be more present than increasing motivation. One way that points can be awarded in attack trees is when the user correctly chooses the most vulnerable basic event to be true.

Leaderboards appear to influence the behavioural engagement dimension, but do not fit in the process-framework. In the context of attack trees, leaderboards can rank users based on their success of the understanding the attack tree. A leaderboard brings competitiveness to the users and promote social comparison among individuals [24-26]. In small settings, people tend to play again to stay on top of the leaderboard, which is another interest besides their interest in attack trees [26]. Playing again corresponds with the intention to reuse the attack tree, which is a characteristic of behavioural engagement, according to Heller et al. [15] and O'Brien and Toms [16]. However, leaderboards may not be useful for attack trees as the main goal is to help stakeholders to understand the concept. When the stakeholder understands the concept of attack trees, repetition of the use of attack trees may not be necessary.

3.3.6 Conclusion

This literature review aimed to define user engagement in the context of attack trees and evaluate the engagement of different gamification elements based on this model. From this review, it is proposed that user engagement in the context attack trees can be seen as three-step process, starting with point of engagement, continuing with psychological engagement and ending with behavioural engagement.

As literature seems to have a scattered perspectives on user engagement, this definition may not be accurate. This is because the definition highly depends on the context it is used in, such as augmented reality, interactive technology and educational

settings. Besides, the lack of identified sources could also be a reason that an unambiguous conclusion about engagement could not be derived.

Another limitation is that alternative gamification elements are not considered in this review. There are potentially other elements, such as story and cooperation elements, that could possibly fit better in the context of attack trees, which may significantly improve engagement. Since the focus of the identified gamification elements was only on engagement, the review has also not identified the side-effects of implementing these elements. For example, a poor implementation of badges and points can result in a mental overload of the user, because there are too many features that are not related to the goal [21].

Further research could look into the communication problems that threat modelers face when communicating attack trees to stakeholders. Identification of these communication problems may help the designer in choosing and applying the proper gamification element to improve engagement. Moreover, in addition to integrating gamification elements, further research could explore other design elements and theories that can enhance the engagement in attack trees, as they may be more effective than gamification elements alone. For instance, applying colour theory to different nodes in the attack tree can highlight certain vulnerabilities, possibly making it easier for the users to understand which vulnerabilities in their system are more or less secure.

3.4 Gamification elements

Serious Gaming and gamification recently gained more popularity as tools for learning. However, researchers are unsure about the effectiveness. Admittedly, S. V. Gentry et al. [6] found some evidence that serious gaming and gamification elements compared to traditional education, in the case of health professionals, may enhance knowledge. The case for people in general is supported by C. E. Catalano et al. [7], whereas C. Girard et al. [3] is less convincing in his analysis. Although it appears from their research that the positive effect of engagement might improve the learning, the clear lack of empirical studies does not seem to support this. It should be noted that this research from Girard et al. [3] is conducted in 2012 while Gentry et al. [6] was in 2019. The research of C. Girard et al. might be for this reason less relevant.

The list of gamification elements can be comprehensive. Fortunately, Toda et al. [37] categorizes all these gamification elements into 5 categories. Please refer to figure 8 for their gamification taxonomy. During the design process, this gamification taxonomy can be utilized as helping hand for coming up with better and new element in the interactive visualization, contributing to the engagement and understanding of the attack trees to users.

Figure 8: Toda et al.'s [37] gamification taxonomy

3.5 State-of-the-art

The aforementioned challenges in the explanation of attack trees have been addressed by some analogies of logic gates and other creative risk engineering models to communicate risks better to non-experts. This is a brief section that consists of the solutions that exist. Not every solution is directly related to the problem, but identifying these items can help further define the problem and provide inspiration for creative ideas to solve the problem.

3.5.1 Analogies of logic gates for contextual information

3.5.1.1 *Pulley logic gates*

Gorischek [43] developed a physical analogy of logic gates using chessboard, eyelets, rings, weights, and strings. The idea behind this physicalization is that the user can pull or lift the two rings on the bottom left and see the output on the bottom right ring. The rings on the bottom left represent the inputs, while the ring on the bottom right represents the output. This installation was built to teach anyone who does not know much about logic gates [42] [43].

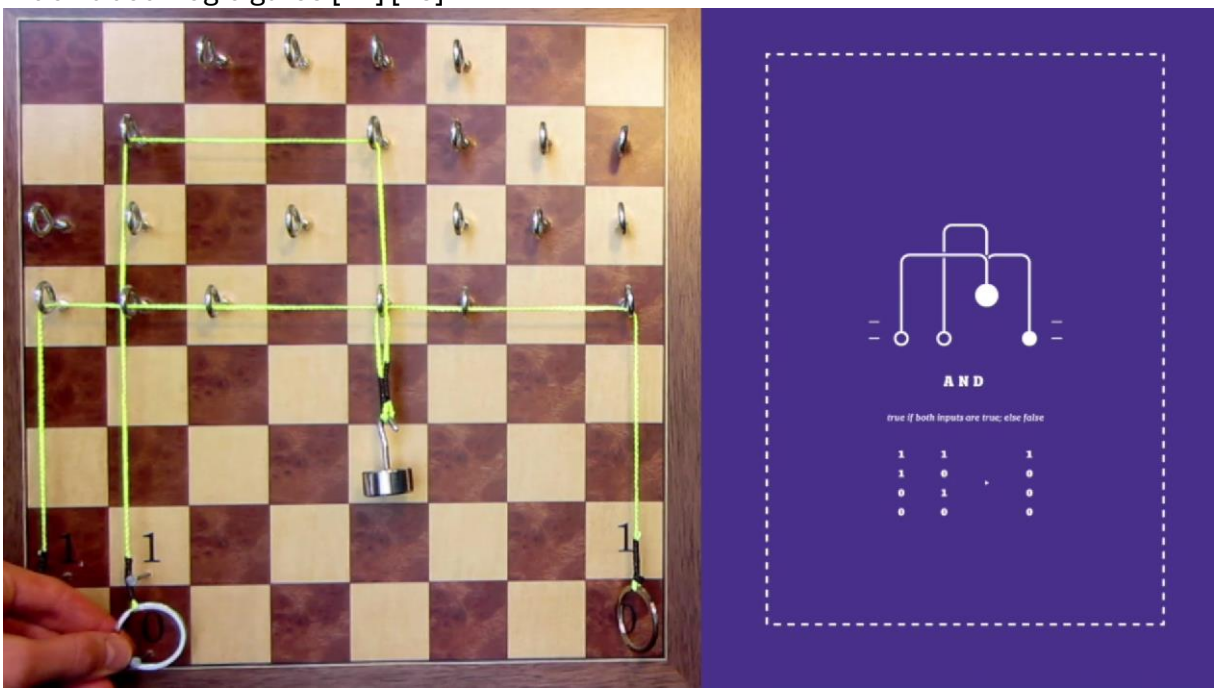


Figure 9: Pulley AND-gate using weights, strings and eyelets.

Main takeaway

This installation makes the interaction with logic gates tangible, allowing the user to see physical results, unlike plain logic gates. However, when only one of the rings is lifted, the user does not receive any feedback because the output ring will remain at the ‘0square. Besides, it is not directly intuitive how the internal system with the additional weight work, which adds more cognitive load to the user. Hiding the weights and parts of the strings behind a box would probably make more sense so that the user does not have to worry about that. Lastly, integrating multiple of these logic gates may be difficult in a large attack tree.

3.5.1.2 *Mechanical and Lego logic gates*

A similar physicalization concept involves pushing mechanical parts that will result in various outputs. Below are two similar examples of gates that utilize Lego blocks and mechanical parts. Figure 10 shows an OR-gate in which the user only has to push one of the handles to have a true output. Figure 11, on the other hand, displays an AND-gate in which the user has to push both sticks to provide a true output.

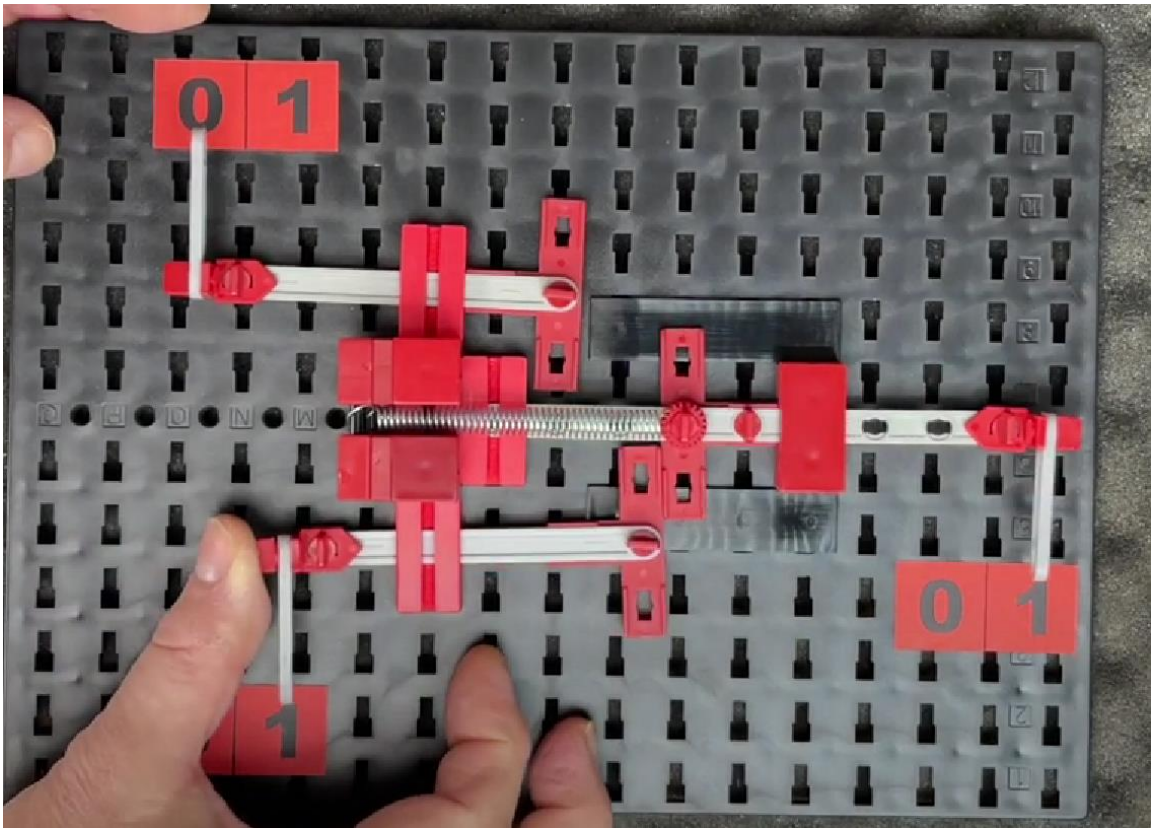


Figure 10: Lego OR-gate using a spring in which the user must push either sticks to provide a true output [44].

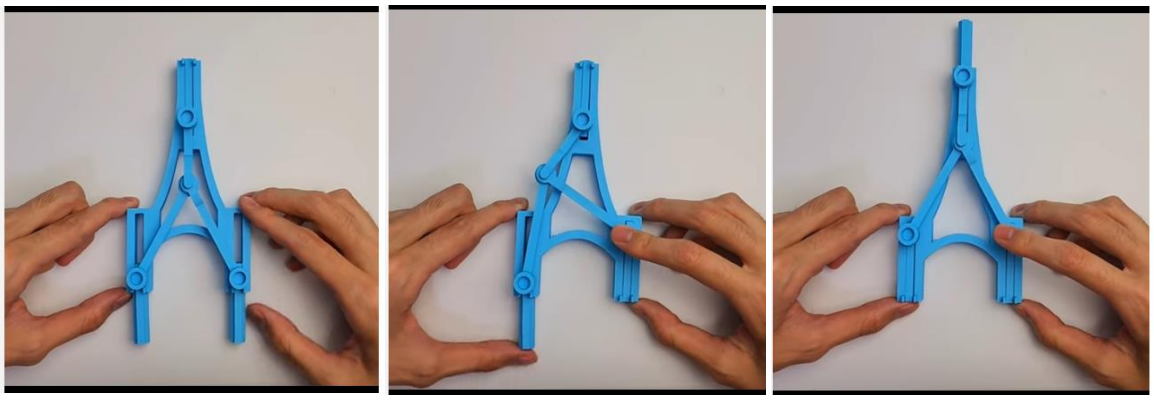


Figure 11: A mechanical AND-gate in which the user also must push either sticks to provide a true output [45]

Main takeaway

This approach can be seen as a relatively simple method of providing contextual information about various logic gates. Although the coloring seems to complicate the filtering of important information such as the inputs and outputs of the gates. Similarly to the previous example, it is not directly intuitive how the internal system with the additional weight works.

3.5.2 Risk visualizations

3.5.2.1 Graph on risks in business processes

Rasmussen et al. [46] focused on developing several interactive visual analytics for the Governance, Risk Management, and Compliance, particularly on business processes and risks. One of their visualizations was a graph-based representation related to this topic, which was similar to the representation attack trees.

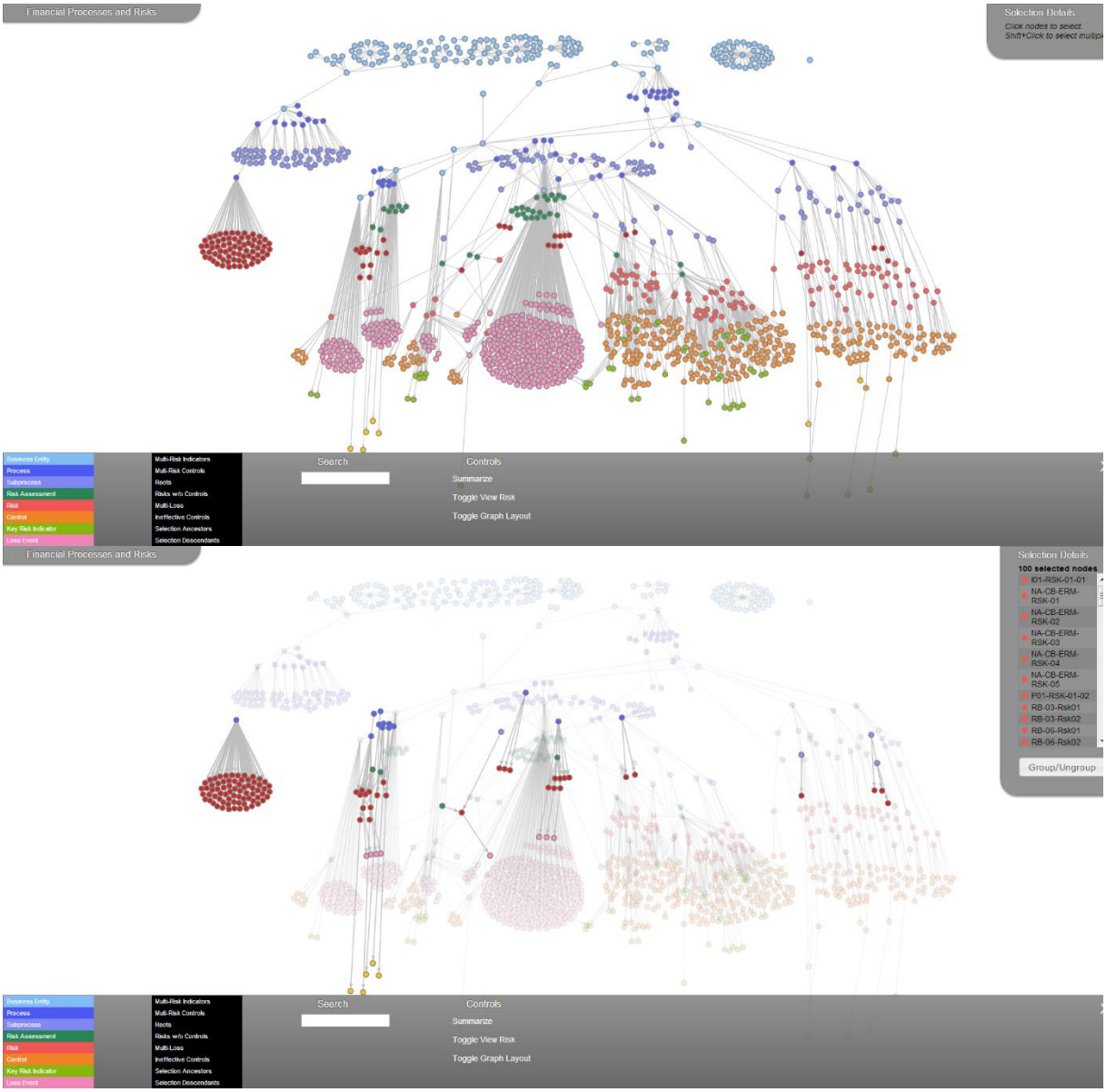


Figure 12: graph-based representation for finding deviations in the data regarding business processes.

Main takeaway
At first glance, this visualization looks rather complex and may be hard to understand. However, closer identification reveals that developers used various techniques to support the cognitive processing.

First, they grouped possible indicators and processes that belong together. This grouping allows users searching for specific processes to focus directly on their relevant group.

Secondly, each type of indicator is grouped by color. Because of this, the user can easily distinguish between risks and subprocesses.

Lastly, visualization includes functionality to filter on specific risks and subprocesses, supporting the cognitive domain of the user. This filter greyed the unselected nodes out, highlighting the selected nodes.

However, a limitation is that user cannot zoom in or out to leave out unrelated information and show the relations between the nodes better.

3.5.2.2 GradeMyDrive

GradeMyDrive is an interactive and visual application that combines accident and other risk data to help people understand their impact on the road. This project was targeted towards drivers who believe they will never cause an accident, because it has never happened to them. This audience can use the sliders to indicate what their usual driving behavior is, and how their behavior can change the outcome for themselves and others.

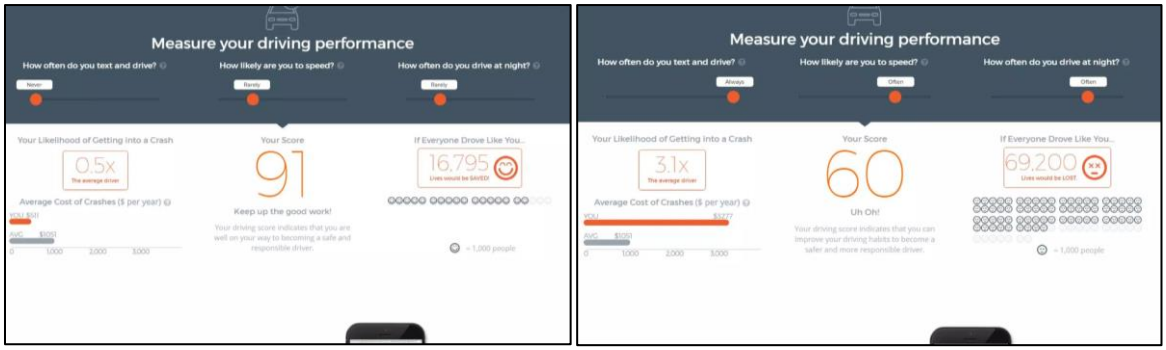


Figure 13: The GradeMyDrive UI. The left image shows the performance of a relatively safe driver, while the right image shows the performance of a relative dangerous driver.

Main takeaway
This visualization directly highlights the most important information, such as the likelihood of getting into an accident and the number of lives that will be lost. Emphasis is achieved through the use of vibrant orange colors and the size of elements, such as the driving score. Besides, by using bar charts, the behaviour of the driver is directly compared with the average driver, which makes the call to action clear. However, it may not be engaging, because the only interaction the user can do is dragging sliders. Overall, it is a quick and novel idea to show risks to users.

3.5.3 Attack tree development application

3.5.3.1 AttackTree+

The application most closely related to attack trees and this project is AttackTree+. It allows modelers to create their own attack trees and offers various functionalities, including computing minimal attack sets.

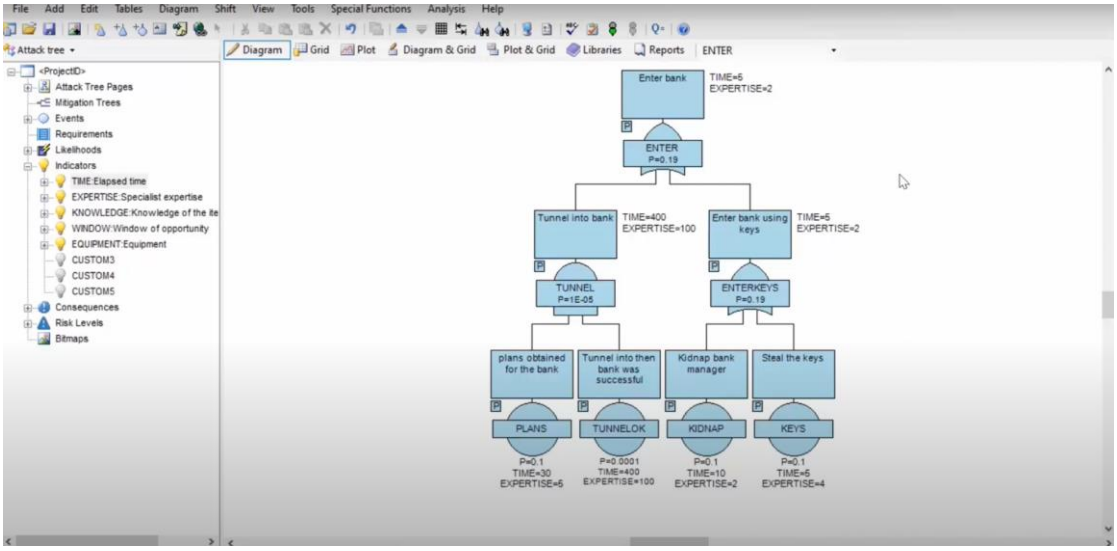


Figure 14: Isograph's AttackTree+ UI [47]

Main takeaway

The application is primarily developed for risk engineers that are building their own attack trees to increase productivity. These engineers can place BASs, and logic gates, and the software will automatically connect them and does its job for you. However, it is not user-friendly for non-expert stakeholders, because it hardly has any visual cues and distinction between elements and gates. Additionally, the minimal attack sets are presented as plain text lines. Overall, this application particularly underlines the importance of the attack tree building process for the developers.

4 Methodology

To plan and guide the design of the interactive visualization, which introduces a better understanding and increased engagement of attack trees, the Creative Technology Design process is applied as a guideline.

The reason this design process is applied instead of other design processes is because this research has overlap with various design disciplines, just like Creative Technology [38]. This research has overlap and integrates various design steps from fields such as Graphical Design, Interaction Design, Engineering Design and more, which are also exactly important components of Creative Technology [38].

Three examples can be provided with the overlap between Creative Technology and this research. First, Graphical Design part of Creative Technology is related to the Visual Appeal of the interactive visualization. Second, Interaction design part of Creative Technology is related with the interaction flow and the many ways the user can interact with the attack tree. Third, Engineering design part of Creative Technology is related with the coding schemes and UML diagrams needed for making the interactive visualization.

4.1 The Creative Technology Design Process

The Creative Technology Design process consists of four phases: Ideation, Specification, Realisation and Evaluation.

This design process is visualized in figure 15. The continuation of this thesis will be structured based on this design process. In following section, each phase will be briefly described and how it is related to this research.

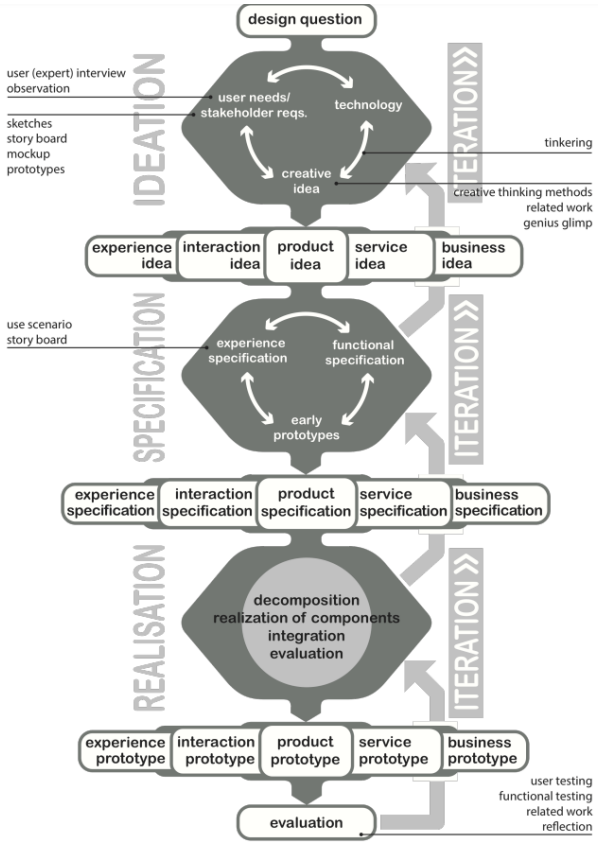


Figure 15: The Creative Technology Design Process [38]

4.1.1 Ideation

The ideation phase starts with a design question. From there, the problem definition is further specified, relevant information is gathered from literature and experts, and ideas are generated to find a solution for the question. The relevant information gathered in Chapter 2 can be used as inspiration for the idea generation. During this idea generation, the designer typically goes through a divergence and convergence phase. The idea of the divergent phase is to explore a broad range of design ideas, while the convergent phase reduces this space to come to a specific solution for the question. User requirements in this convergent phase can be defined from these specific ideas.

Well-known diverging brainstorming methods, such as Brain Writing, and the AOKI method are typically used in this phase. However, since this research is conducted individually, these methods may not be applicable. Instead, no specific process is used, but rather the sixteen brainstorming tips for Game Design described by Jesse Schell [40] are utilized. These tips are employed as gamification elements and game design principles are closely related with each other. Some tips including sketching your ideas, writing as much down as possible, on walls, papers and mixing and matching already generated ideas.

4.1.2 Specification

In the specification part, various low-fidelity (lo-fi) prototypes are evaluated to determine their effectiveness in solving the design problem. Both functional and non-functional requirements are more specifically defined. For this research, interaction user flow map, playtesting, sketches and specified requirement are utilized. To clarify, an interaction flow map is a diagram that shows the paths users can take based on their actions and interactions with the technology [39]. This tool is useful for making the user experience more concrete and facilitating the user testing. Besides, it provides a better understanding and clarity in the design of the interactive visualization. An example of an interaction user flow map of the HealthMes App of Apple Watches can be found in figure 16. From this interaction flow map, functional and non-functional requirements can be refined and ordered based on the MoSCoW method. The prioritization from the MoSCoW method is in this research useful, because of the limited time for this thesis.

Furthermore, playtesting is integrated in this phase. Playtesting is all about whether the game (or challenge) designed causes the experience for which it was designed. As time is limited and playtesting may often be needed to happen to check whether the interactive visualization does what it needs to do, only tissue testers, such as student or friends and family are used as they are highly available and comfortable talking about the research. Besides testing the things the researcher knows, it also can reveal things that the researcher is not looking for to verify. And by truly observing the playtest, the researcher can find new requirements for the design of this visualization.

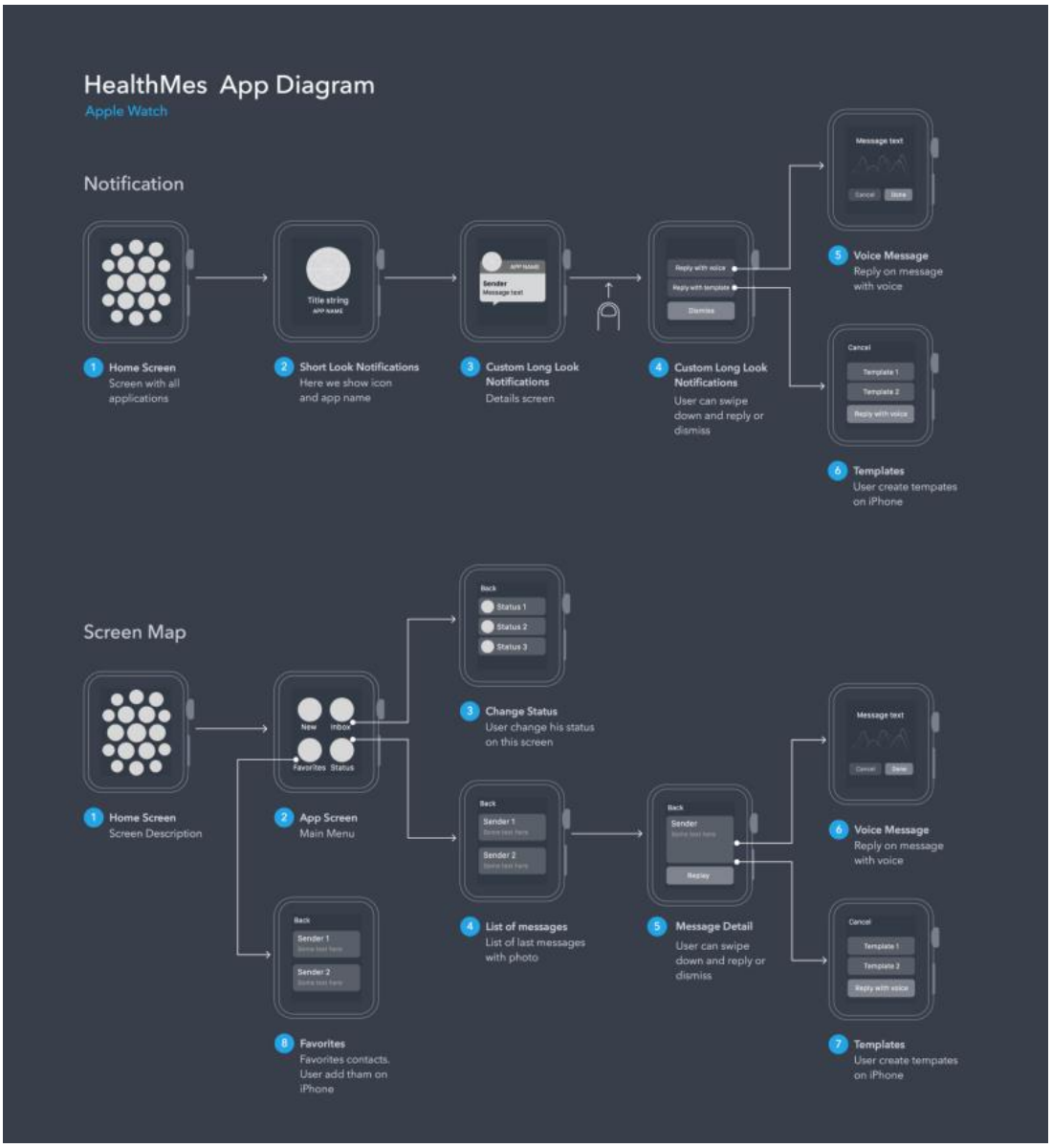


Figure 16: Interaction user flow map of the HealthMes App of Apple Watches [49] [50].

4.1.3 Realization

In the realization phase, the interactive visualization of the attack tree is made. The process of the development and the working behind this visualization is made clear. This can also include functional tests of specific components in the visualization, such as the working of different logic gates. This part will particularly consist of components related to the Game Engine Unity and the C# code.

4.1.4 Evaluation phase

In the last phase of the Creative Technology Design Process, user tests are the most well-known method verify the requirements of the visualization. For this research, it is convenient to make the understanding and engagement measurable, using numbers. Because of this, statistical methods, such as independent t-tests or Mann-Whitney U

tests, can be applied to draw conclusion whether there are significant improvements in the new design.

4.1.5 Iteration

The phases in the Creative Technology Design process seems linear, but it is an iterative process. To illustrate, when the researcher faces challenges in realizing the visualization, such as applying different gamification elements, the researcher can go back to the specification process and redefine the requirements. Moreover, this process is used as a guideline and not as a strict method.

5 Ideation

5.1 Preliminary ideas

In this part of the Ideation phase, several ideas will be presented that came out various ideation sessions. During these sessions, various ideas were presented from these individual sessions. In the appendix, a mood board from an ideation session can be found. The next section will showcase the three potential ideas that have been further elaborated.

5.1.1 Concept 1: Christmas Attack Tree

Using analogies, as seen earlier, can be effective way to increase the understanding. In this idea, an attack tree applies the analogy as Christmas tree with light bulbs, as shown in figure 17. The root or the target is visualized as the tree-topper and the BASs are shown as a battery holder. The user can place a battery inside the holder to toggle the BAS to true. The rest of the nodes and logic gates are represented as lightbulbs in a parallel or series circuit. When certain BASs are true, the lightbulb that relates to that BAS, the parent, will light up and continue further into the attack tree.

This concept is a variation between a physical installation and a digital visualization. The Christmas attack tree will be displayed on a screen, while the battery holders and batteries are physical elements, which communicate with the digital image of the Christmas tree. The reason for this choice was to make the interaction more tangible.

An issue with this idea is that the analogy of a parallel and serial circuit implies a cyclic connection. However, an attack tree is not cyclic, which may distort the perspective of an AND and an OR gate. In addition, it suggests a sequence. In figure 17, it can be seen that when the second BAS is set to true, it suggests that not only the *red* light bulb will light, but also the *blue* light bulb should light, as that seems a logical flow of electrons. In contrast, the *blue* light bulb should only light up when both the *red* and the *green* light bulb are lit.

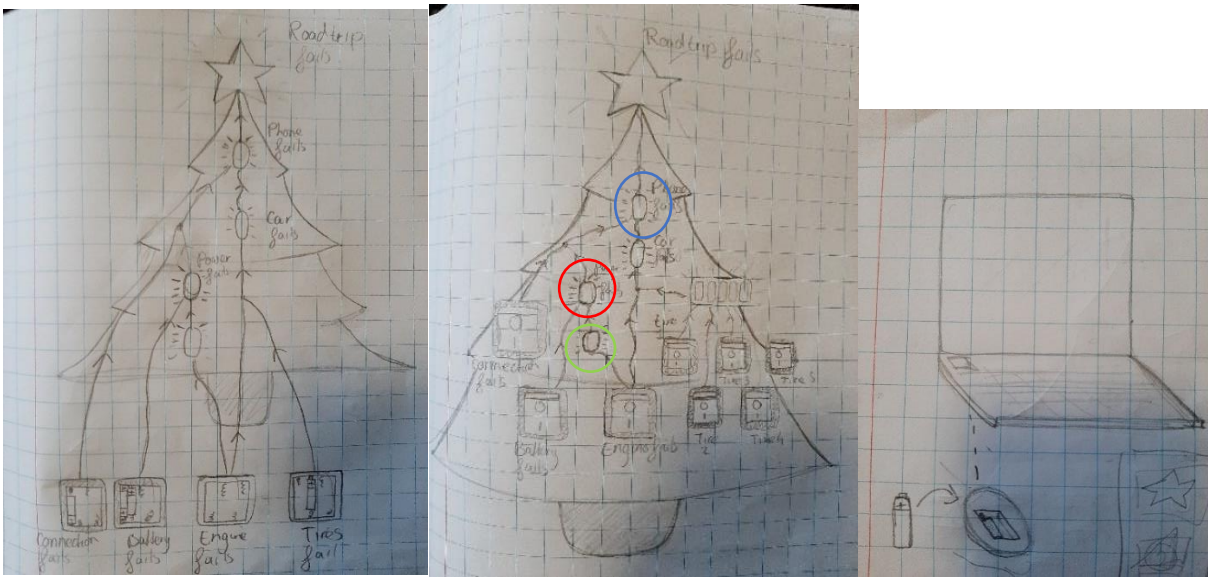


Figure 17: Sketches of the Christmas attack tree

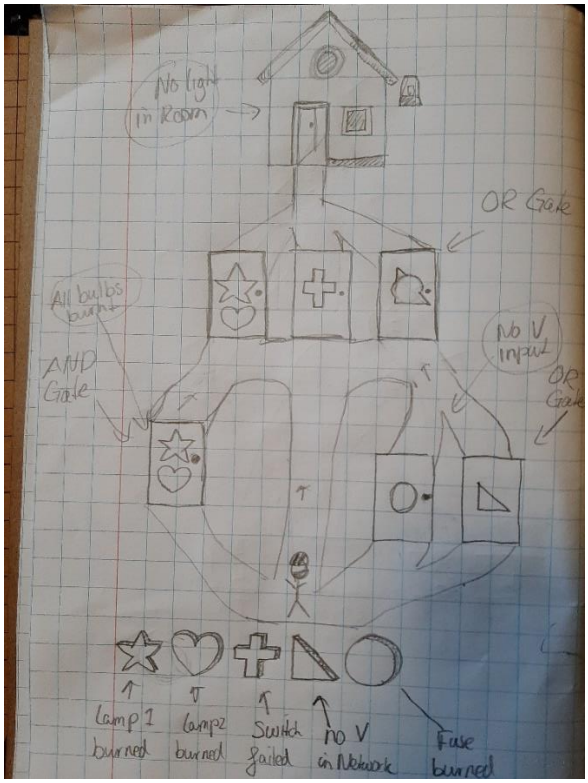
5.1.2 Concept 2: Burglar attack tree:

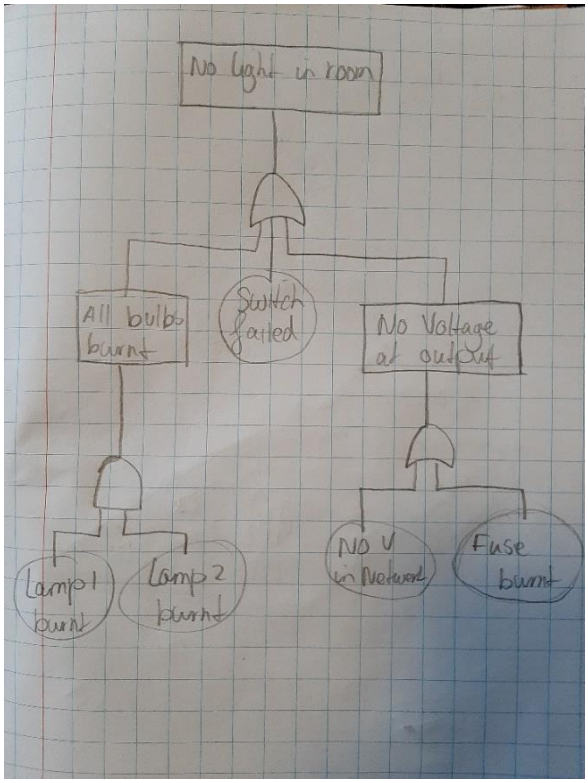
Another analogy for visualizing an attack tree is using the scenario of a burglar breaking into someone's house. The burglar starts at one of the leaves of the attack tree and the house represents the root, the target of the burglar. If the burglar can reach and make a path from the source to the destination, then an attack can be considered succesful.

The AND and OR gates are represented as doors. To recap, both BASs connected to the AND gate must be true before the output can be true. This is represented as a single door with two locks. To open the door and continue the journey, the burglar needs to use two keys that fit on those locks. So, the keys in this visualization can be seen as BASs. Therefore, a BAS is true when a key is used to open a door.
For an OR-gate to be true, only one of the BASs connected to the gate must be true. This is represented with two doors, each with their own lock, leading both to the same output lane.

In contrast with the previous concept, this idea is completely digital. Developing a physical burglar character with fluent physical interactions may not be feasible within the time contraint. Instead of this, the user can move the burglar with the mouse and keyboard towards a door and click on specific User Interface (UI) keys to unlock certain doors to reach the root. The goal for the user is to reach the target by selecting as few keys as possible. This will create affinity with the concept of minimal attack sets within attack trees.

The challenging part of this visualization lies in the depth of the attack tree. This is because each door needs a different lock and there are only so many distinct shapes that can represent a lock. This may or may not potentially create more cognitive load on the user.





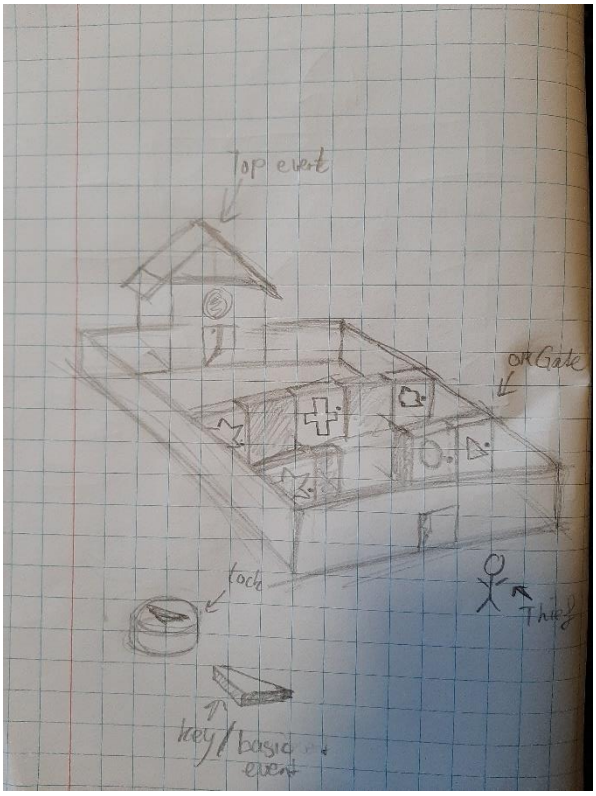


Figure 18: Sketches of the Burglar attack tree

5.1.3 Concept 3: Coloured LEDs trail tree with levers/buttons

Instead of an analogy, this idea changes the relations, normally shown as arrows from a child node to parent node, to a trail of LEDs trail going towards the parent node. The BASs of the attack trees can be toggled by a lever or a switch. When toggling a BAS to the 'true' state, then LEDs going towards the parent node, or logic gate, light up. This is done for every parent-child relationship up and until the root node.
Figure 11 shows the sketch of this idea concept.

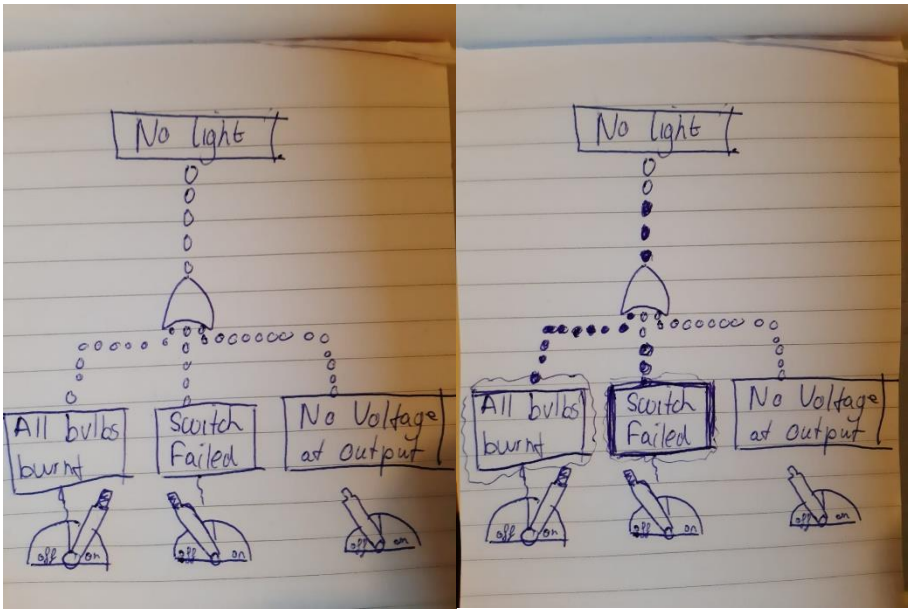


Figure 19: Sketches of the LEDs trail attack tree

5.2 Preliminary requirements check

Based on the conclusion of the literature and state-of-the-art, a brief preliminary requirement list has been developed, so that a decision can be made about the final concept. These requirements are split into three parts: engagement requirements, understanding requirement and non-functional requirements. These parts are considered to provide a wide range of requirements.

5.2.1 Understanding requirements

- The final design must provide more contextual information than the original the attack tree.
- The final design must resemble the original structure of the attack tree.

5.2.2 Engagement requirement

- The final design must fit inside the created engagement process-framework.

5.2.3 Non-functional requirement

- The final design must be flexible in adapting their structure by threat modelers and expansion to complex analysis, such as quantitative or cost-damage analysis

Based on these requirements, a brief consideration is created for each concept. This is shown in table 1.

		Christmas attack tree	Burglar attack tree	LEDs trail
Understanding	Contextualization/Intuitiveness	+/-	++	+/-
	Original structure	+	+	++
Engagement	Fit of engagement process (feedback, interactivity, visual appeal, etc.)	+	++	+/-
Non-functional	Flexibility/expandability	-	+	+

Table 1. Ranking of the ideas based on the preliminary requirements.

From this table, it is evident that the Burglar Attack tree seems to have the most potential to successfully answer the research questions.

6 Specification

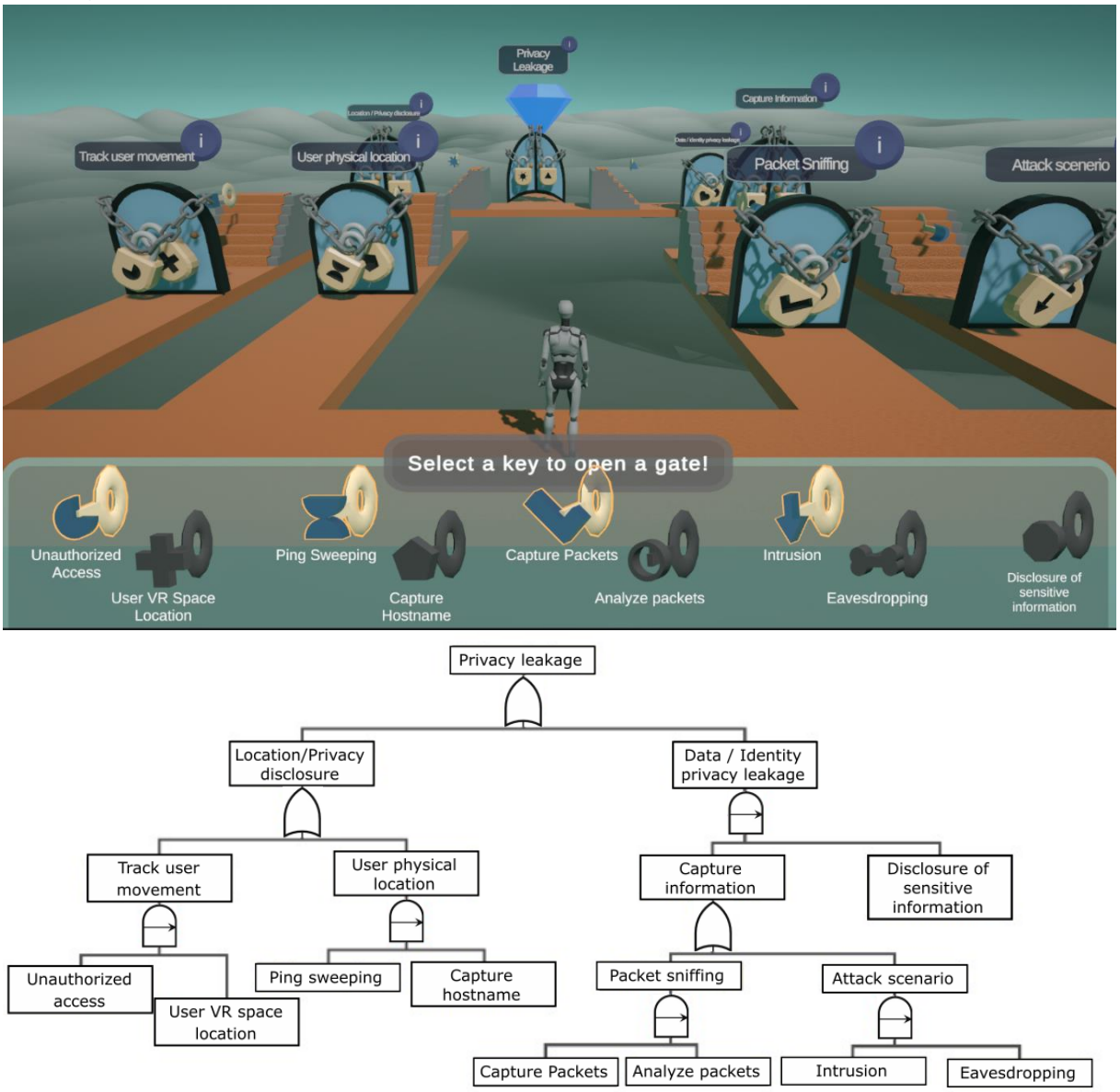


Figure 18b: The more defined overview of the visual installation. The visualization is a translation of the attack shown below. The reason this attack is chosen is because the tree is not too large, but may suffer from cognitive overload, contains technical jargon, and consists of node types, such as Sequential-AND gates which seems a bit less intuitive than its simpler version: the AND gate.

After choosing the final concept, a list of design requirements, both functional and non-functional requirements can be formulated to ensure that the research question will be answered. The MoSCoW (*Must, Should, Could, Would)* prioritization framework is applied to provide the developer with an overview of what requirements are more important to fulfil than others. This is important, because the developer has a deadline for the design to be completed.

6.1 Design requirements

The research question to be answered relates to increasing both the engagement and understanding of qualitative attack trees in general. These requirements are listed below.

6.1.1 Requirements of engagement

The requirements can be based on the three-step framework formulated in the background, so that every aspect in the framework will be integrated in the design. These steps are *point of engagement*, *psychological engagement* and *behavioural engagement*.

6.1.1.1 Point of engagement:

Must: Visual appeal

The final design must be visually appealing.

From the background, the visual appeal can initiate the level of engagement at the start of the user. Visual appeal can still be somewhat subjective, but it can be fuelled by the correct composition and use of colour theory.

Could: Interactivity

The final design must integrate multiple methods to interact with the design.

According to the background, this can foster the engagement point. Various methods could include multiple mediums such as touchscreen, keyboard and mouse, but also the way the user interacts, such as clicking, swiping and holding.

6.1.1.2 Psychological engagement requirements

Must: Instant feedback

The final design must provide instant feedback to every user's interaction.

To maintain engagement, instant feedback provides control to the user. When the user loses the sense of control of the visualization, they lose interest in interacting with the design.

Must: Optimally challenging problem

The final design must provide an optimally challenging problem for the user to solve.

This requirement particularly emphasizes on cognitive engagement. The user may lose interest when the given challenge is too simple or too difficult.

Should: User awareness and control

The user must be aware of the environment of the design.

This goes back to the user's control. The environment can be visually appealing and provide instant feedback, but their engagement can decrease if the user does not know what or why something is happening. Playtesting with others can help, but it remains difficult to accomplish this requirement in the design. Therefore, this requirement is little bit less important than the others.

Would: Repetition

The final design would minimize repetition in the provided challenge that are not related to the main material, such as BAS's, logic gates and attack sets.

This requirement is cumbersome. Repetition can result in a lower engagement, but it can also provide the user to understand the material better. Therefore, to balance the two, repetition would only be minimized in the parts of the design that are not much related to the original attack tree itself. So, this will particularly be applied in the visual appeal and not to the logic behind the attack tree itself.

6.1.1.3 Behavioural engagement requirement

Should: Further exploration

The final design should be open for further exploration of attack sets within in the attack tree.

This can include finding new minimal attack sets and new paths towards the root. As mentioned in the background, much exploration is not necessary as the main goal of the installation is to understand the concepts of attack trees.

6.1.2 Requirements for understanding

In the background of this thesis, multiple possible problems about the understanding were identified. Some requirements in the design can potentially help solving these issues.

Must: Analogy

The final design must provide an analogy of logic gates.

Analogies can shift the cognitive perception of users [29] [31]. Using analogies help users to map familiar relationships onto the new material [31]. The representation of logic gates with doors can potentially help with that.

Must: Original symbolic representation

The original symbolic representation of logic gates must be recognizable in the final design

Analogies can also be misleading when it is abstracted too far, because the relation between the analogy and concept is hardly visible [30]. By maintaining the original symbolic representation, the level of abstraction may be reduced.

Must: Progressive reveal
Each layer of the attack tree must be revealed progressively as the user interacts with the attack tree.

As this prototype will be a 3-dimensional visualization, the camera and camera could be lowered to reduce the Field of View of the layers in the top layers, while making the layers at the leaves more prominent. This approach also helps the users to focus on one part of the tree at a time.

Should: Explanation of jargon
The final design must include an explanation of jargon/technical terms or minimize its use in the design.

This can prevent confusion among non-experts about the meaning of various parts of the attack tree.

Should: Miller's law
The final design should only show a maximum of 7 Basic Attack Steps to the user.

According to Miller's Law, users can only hold approximately 7 items [51] in their working memory. Showing all elements to the users, as in the old design, may overwhelm them and may result in cognitive overload. Chunking is a technique that may help reduce the number of items.

Should: Leaderboard
The most important minimal attack sets must be ranked in a leaderboard.

This highlights the risk severity of various nodes and attack sets and may possibly help non-experts in their decision-making.

6.1.3 Additional usability requirements

Even though a sufficient number of functional and non-functional requirements have been identified to improve engagement and understanding, many requirements are not per se related the usability of the interactive visualization, which is also an important factor. If the usability is poor, there is a high chance the other requirements will not be met. Therefore, some additional requirements have been constructed.

Must: Association keys with locks
There must be no ambiguity about which key is associated with the specified lock.

If this fails, then this attack may even become more confusing than the original attack tree

Must: Communication status node.
The status of each node in the attack must be communicated to the user

This is related with the instant feedback requirement. For example, when a user activates a lock, then feedback should be provided to the user about the BASs. The same is required for doors and locks.

Could: Flexibility to make own attack tree

The final design could allow threat modelers to make their own attack tree.

This is specifically convenient for widespread use and scalability. With clean code and tools, threat modelers can create their own attack trees in the design, which can then be shown to stakeholders to illustrate system vulnerabilities.

6.2 Software components

Developing the final design requires various software tools. Each of these tools serve their own purpose for the development process. The essential software components are listed below for the final design.

1. **Unity version 2022.3.22f1 (Long-term Support)**

Unity is the game engine platform and provides developers with features such as scene editing, physics simulation, animation and rendering. A game engine such as unity accelerates and facilitates the development of the interactive visualization. It can be used to manage game environments, designing levels and scripting various mechanisms.

2. **Visual studio (Integrated Development Environment) 2019 + C# language**

In Visual Studio, developers write and debug code. Unity automatically connects and integrates with Visual studio. This environment is used to write C# scripts for game mechanics and other functionalities for the application.

3. **Blender 3D 3.6.12**

Blender is a free and open-source 3D asset creation software. In this software, 3D models can be created, textured, rigged and animated, which can then be imported to Unity. To illustrate their usefulness, the visual representation and animations of locks, keys and doors are created in this program.

4. **Inkscape**

Inkscape is a Vector Graphics Editor and is utilized for designing 2D sprites and UI elements. Like Blender, these assets can be imported to Unity to bring them together in the application.

6.3 Hardware components

Running unity applications require certain amount of computing power to run it smoothly on devices. Therefore, the recommended hardware components include for computer devices:

- **CPU (processor): Intel core i5 (or higher)**
- **GPU (Graphics card): NVIDIA GeForce GTX 700**
- **RAM: 4GB (or more)**
- **Display: a monitor of at least 1920x1080**
- **A keyboard and a mouse**

6.4 Persona

Thinking about situations in which the prototype could be utilized can help the design process of attack trees. Storyboards illustrate how the interface should work without locking in specific design elements. They ensure that every stakeholder, including the designer, shares the same understanding about the design [52].

However, the problem with storyboards is that they do not specify the exact placement of buttons or other specific interactions. This can be solved with an interaction flow map.

Before making storyboards, personas should be created to provide a clear picture of who the users are and what their goals and challenges are. Furthermore, personas allow for the creation of realistic scenarios regarding the design. This can ultimately lead to more effective design solutions.

Below a persona is formulated of a policymaker from a made-up insurance company.

Bart Mulder

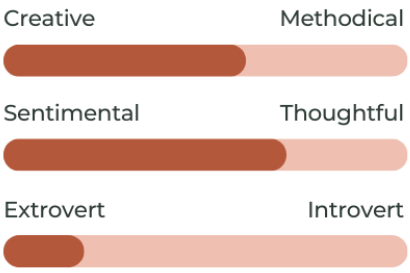
Co-founder AEGGON Insurance

Age: 64 years	Country Netherlands
Sex: Male	Education: Master's degree in Business Administration
Income level: €150,000 / year	Occupation: Co-founder AEGGON

PERSONALITY

Bart is a strategic and analytical thinker, always looking for new ways to lift the business to a higher level. He is social and likes companionships from others. He enjoys being in charge.

Free time: he enjoys golfing and reading business strategy books,



GOALS AND OBJECTIVES

- **(Business) growth**: Compete with other insurance companies by innovating
- **Customer trust:** satisfy customers with reliable services.
- **Risk management:** control risks such as financial, operational, and cybersecurity
- **Efficiency**: improve efficiency and reduce costs

PREFERENCE IN COMMUNICATION

- **Familiar with basic laptop** and smartphone **usage** for business operations, but l**acks experience** with with **IT systems.** Only works with digital dashboards and reports for decision-making
- **Prefers visual and easy-to-understand summaries.**

SKILLS

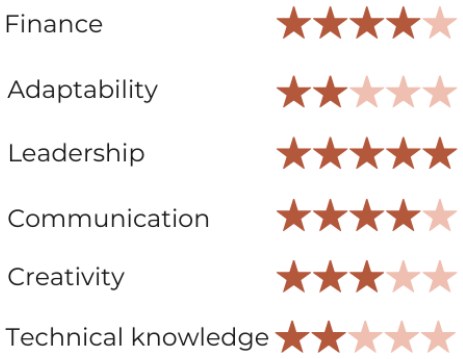


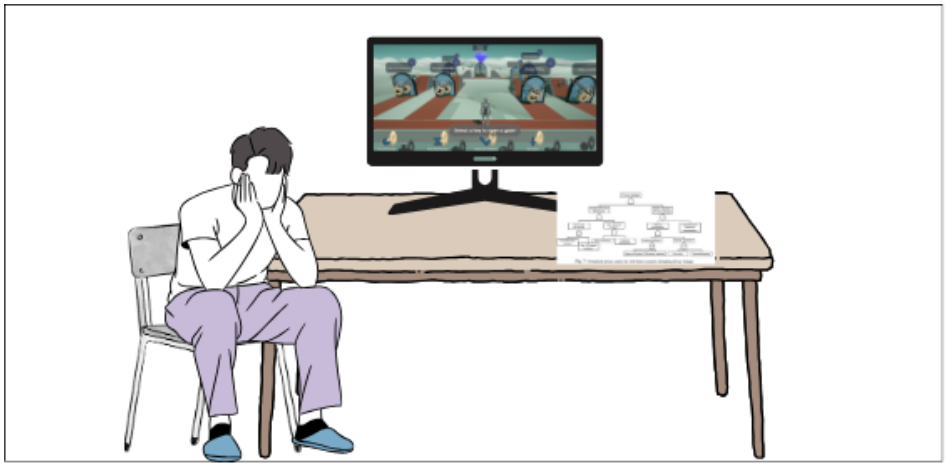
Figure 20: A persona of a co-founder relevant for the attack tree design. The portrait is generated by the website 'This Person Does Not Exist' [53].

6.5 Storyboard

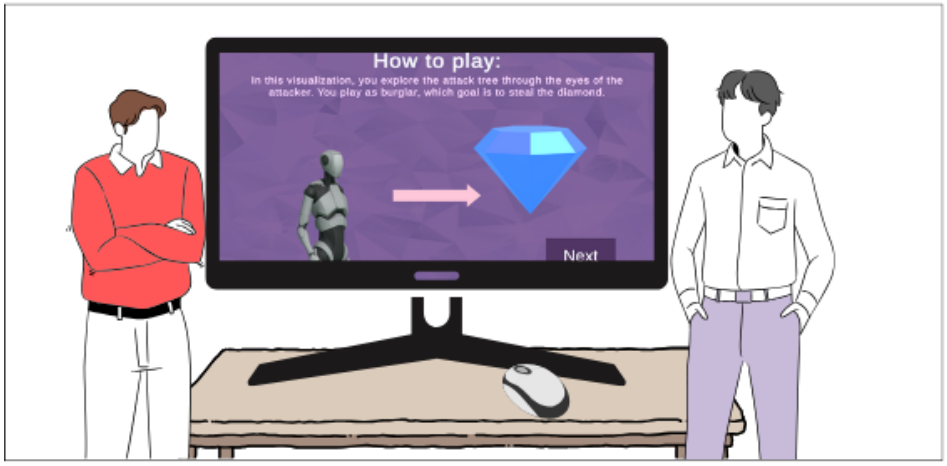
With the identification of the personas, the storyboard can be crated. This makes sure that every stakeholder is on the same page.

TITLE — An explanation about attack trees

NAME — Casper ten Holder

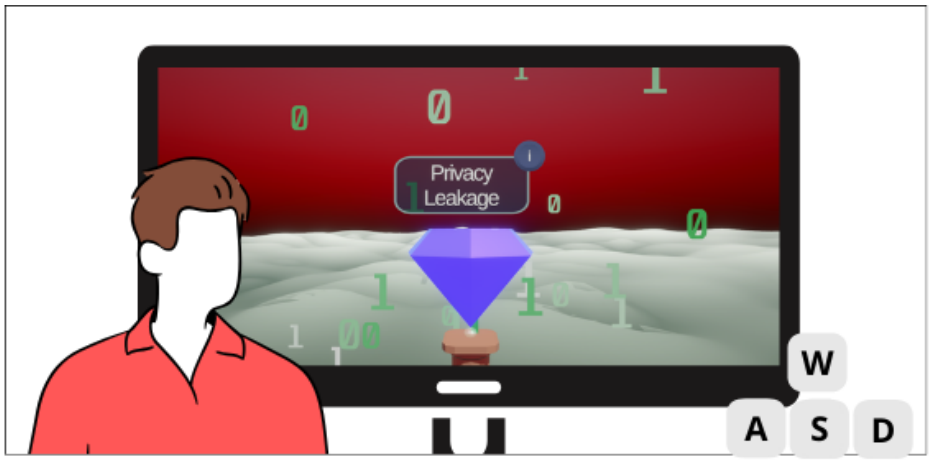


DIALOGUE: Erik is a risk engineer who is analysing the vulnerabilites in the IT-system of the insurance company. He shapes his analysis into an attack tree and would like to inform the policy-makers in the company about it. However, he is unable to effectively communicate the attack tree to Bart. If Bart misunderstands the attack tree, then it could result in uneffective policy regarding their security system. Therefore, Erik decides to use a new tool shape his analysis into a more understandable attack tree.

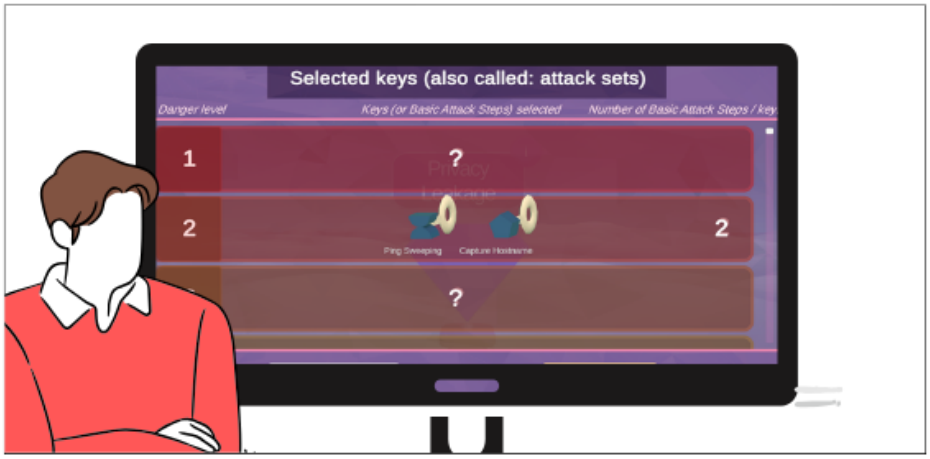


DIALOGUE: Instead of Erik explaining the attack tree to Bart, he shows him the interactive visualization. He shows Bart the burglar that can move in the environment with doors, by pressing arrow keys and computer mouse. Erik gives Bart the task to reach the diamond with the burglar, by using as few keys as possible. Bart clicks through the instructions and starts his journey

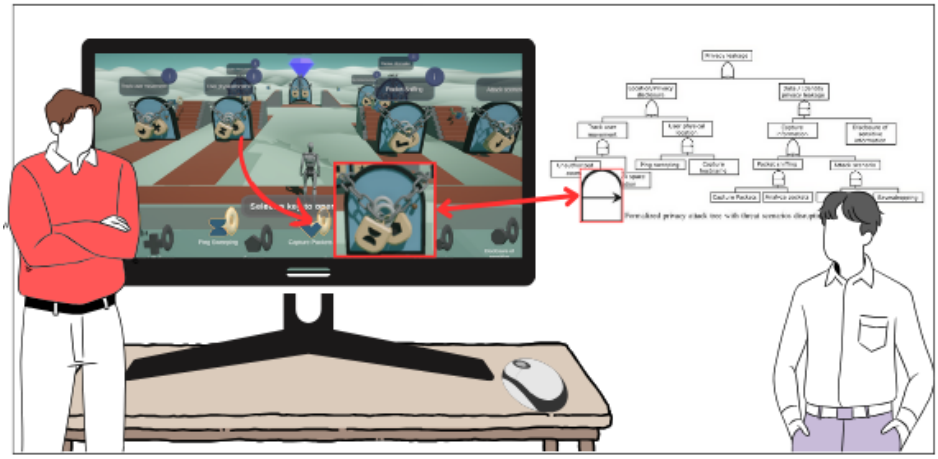
DIALOGUE: Bart moves the burglar through the tree by clicking on different keys. He sees that the some doors behave differently than other doors. Bart hovers over the text balloons to get more information about the keywords.



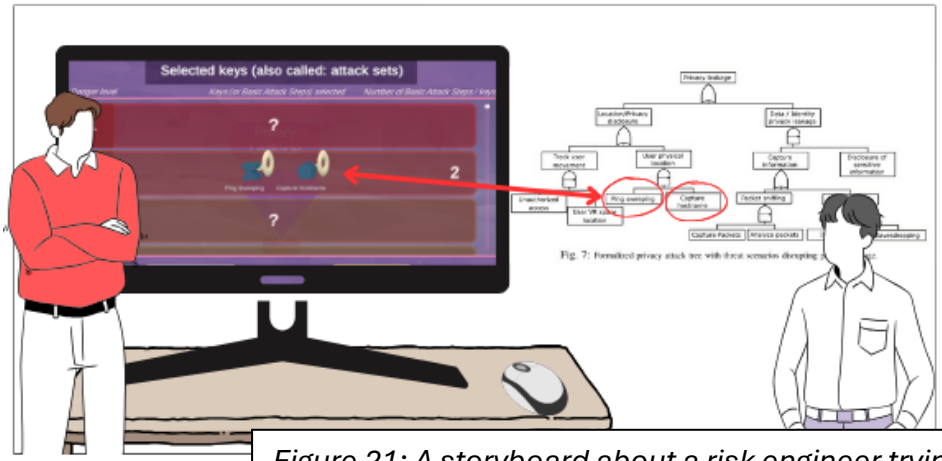
DIALOGUE: Bart reached the diamond and sees the bits leaking and blinking red sky, resembling a alarm. Sensitive data is leaked by the burglar.



DIALOGUE: Bart reached the diamond and sees the leaderboard of found attack sets. He examines the leaderboard and notices that the keys he had selected were high up in the leaderboard. He sees from the danger level that the path he walked was relatively dangerous. However, he does not understand why he is given this task from Erik. At the end, Erik decides to click the return butting and he sees a debriefing section about the explanation of the new model in relation to the original attack tree.



DIALOGUE: Both the debriefing of the visualization and Erik can explain that this interactive visualization is abstracted representation of the vulnerabilities in their IT-system. Erik compares this tool with the original attack tree and tells Bart that these doors, locks and keys represent the logic gates in the attack tree.



DIALOGUE: Erik continues his explanation by associating the found keys of the visualization with the attack sets of the original attack tree. Bart understands this as he maps his known relationship with locks and doors onto the new material of attack trees. Bart also understands the severity of the different parts of the attack tree now. Bart can make an informed decision about the vulnerabilities in their IT system and decide how to allocate their resources.

Figure 21: A storyboard about a risk engineer trying to explain attack tree about their IT-system to a non-expert Co-founder from an insurance company.

6.6 Interaction flow map

The storyboard provides a visualization of the chosen idea and provides the flow of the project. However, it lacks the exact technical 'microinteractions' the user has with the design. It does not explain the behaviour of the interface of the attack tree. Therefore, an interaction flow map is created to give an extensive explanation about this.

A microinteraction consists of 4 elements according to Saffer [54, 55]: Triggers, Rules, Feedback and Modes.

Trigger: This is the event that starts the microinteraction. An example of the trigger is clicking a button or scrolling on a screen.

Rules: Rules specify what happens next when the microinteraction is triggered. For example, when the user triggers the submit button by clicking it, the outcome will be that the form will be sent to the system's database

Feedback: This is the response that the user sees after the trigger occur, which helps them understand their action. For instance, the submit button becomes green after the user clicks on the button.

Mode/Loops: This defines the lifecycle of the microinteraction, managing duration and repetition of the interaction.

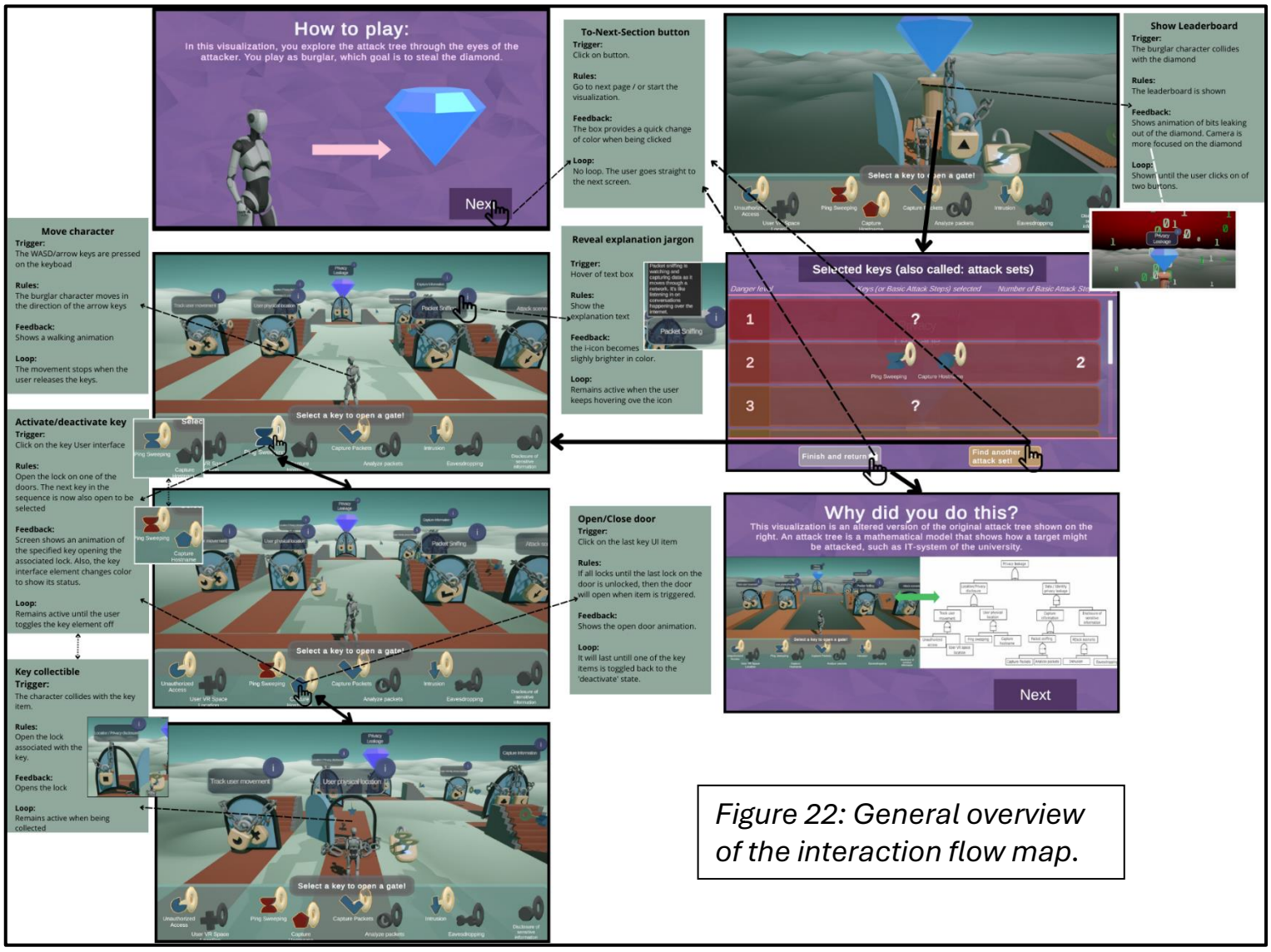
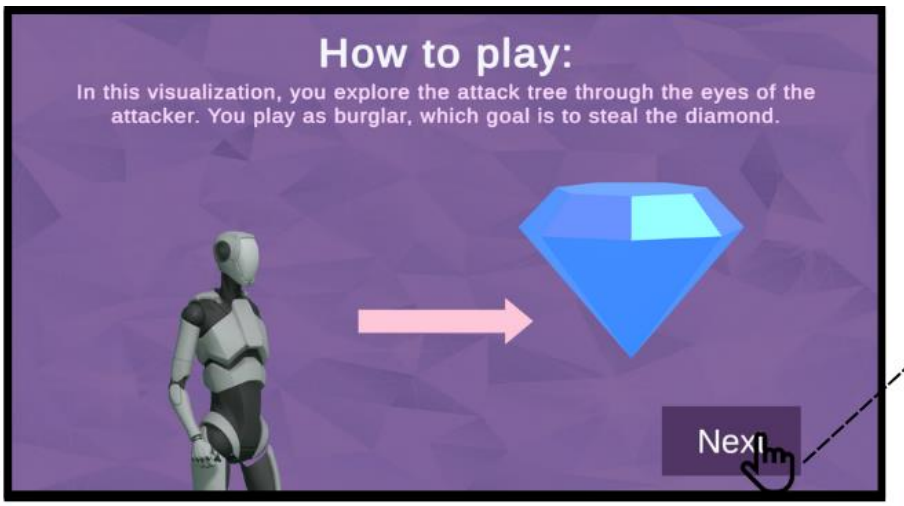


Figure 22: General overview of the interaction flow map.



To-Next-Section button

Trigger:
Click on button.

Rules:
Go to next page / or start the visualization.

Feedback:
The box provides a quick change of color when being clicked

Loop:
No loop. The user goes straight to the next screen.

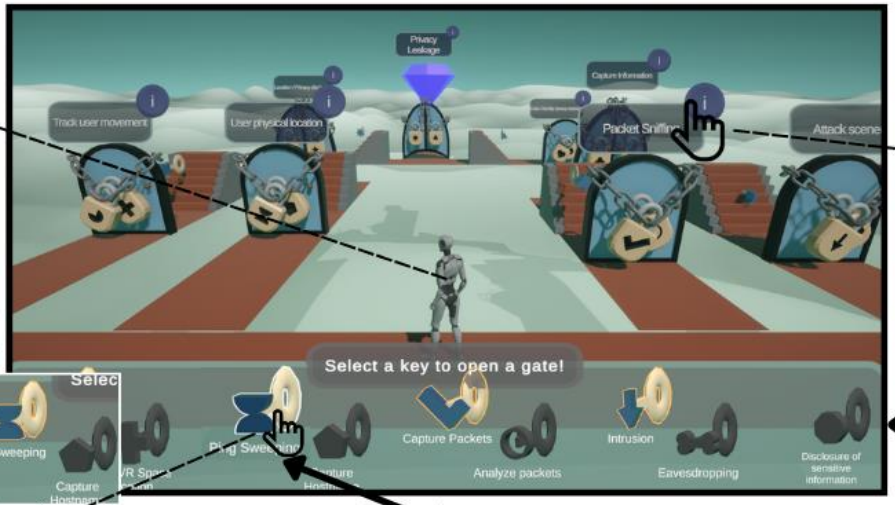
Move character

Trigger:
The WASD/arrow keys are pressed on the keyboad

Rules:
The burglar character moves in the direction of the arrow keys

Feedback:
Shows a walking animation

Loop:
The movement stops when the user releases the keys.



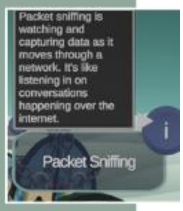
Reveal explanation jargon

Trigger:
Hover of text box

Rules:
Show the explanation text

Feedback:
the i-icon becomes slighly brighter in color.

Loop:
Remains active when the user keeps hovering ove the icon



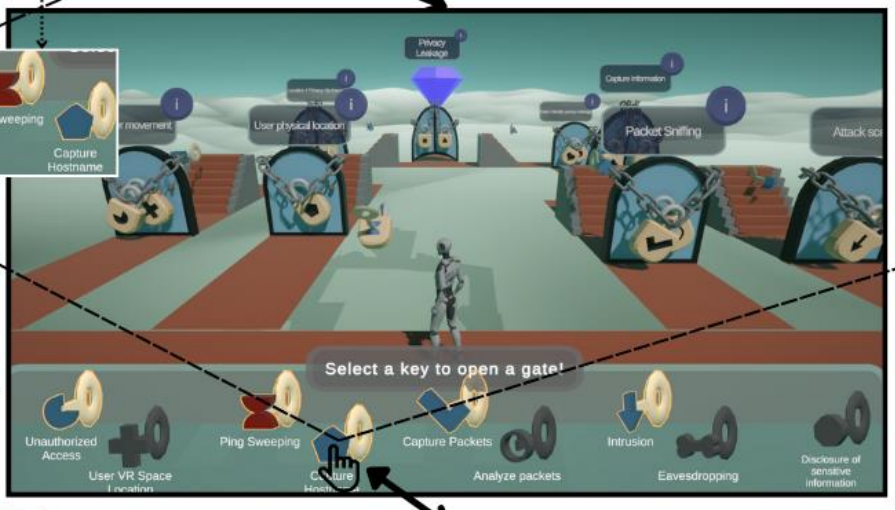
Activate/deactivate key

Trigger:
Click on the key User interface

Rules:
Open the lock on one of the doors. The next key in the sequence is now also open to be selected

Feedback:
Screen shows an animation of the specified key opening the associated lock. Also, the key interface element changes color to show its status.

Loop:
Remains active until the user toggles the key element off



Open/Close door

Trigger:
Click on the last key UI item

Rules:
If all locks until the last lock on the door is unlocked, then the door will open when item is triggered.

Feedback:
Shows the open door animation.

Loop:
It will last untill one of the key items is toggled back to the 'deactivate' state.

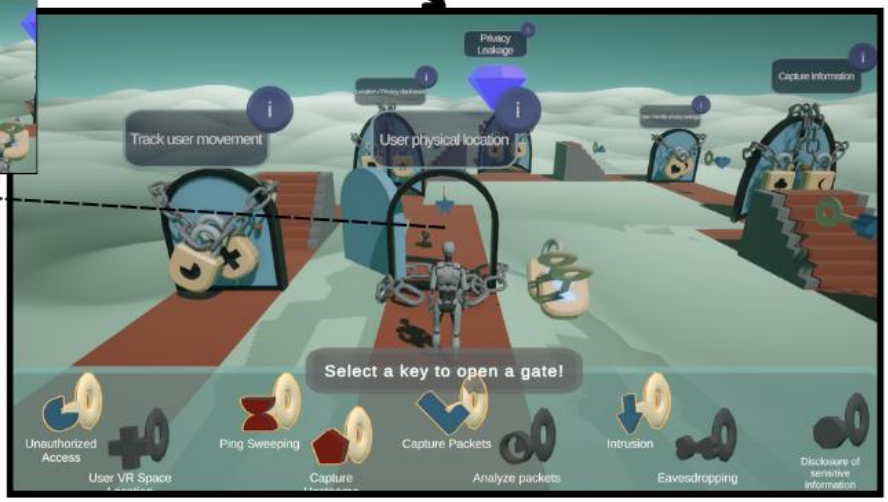
Key collectible

Trigger:
The character collides with the key item.

Rules:
Open the lock associated with the key.

Feedback:
Opens the lock

Loop:
Remains active when being collected



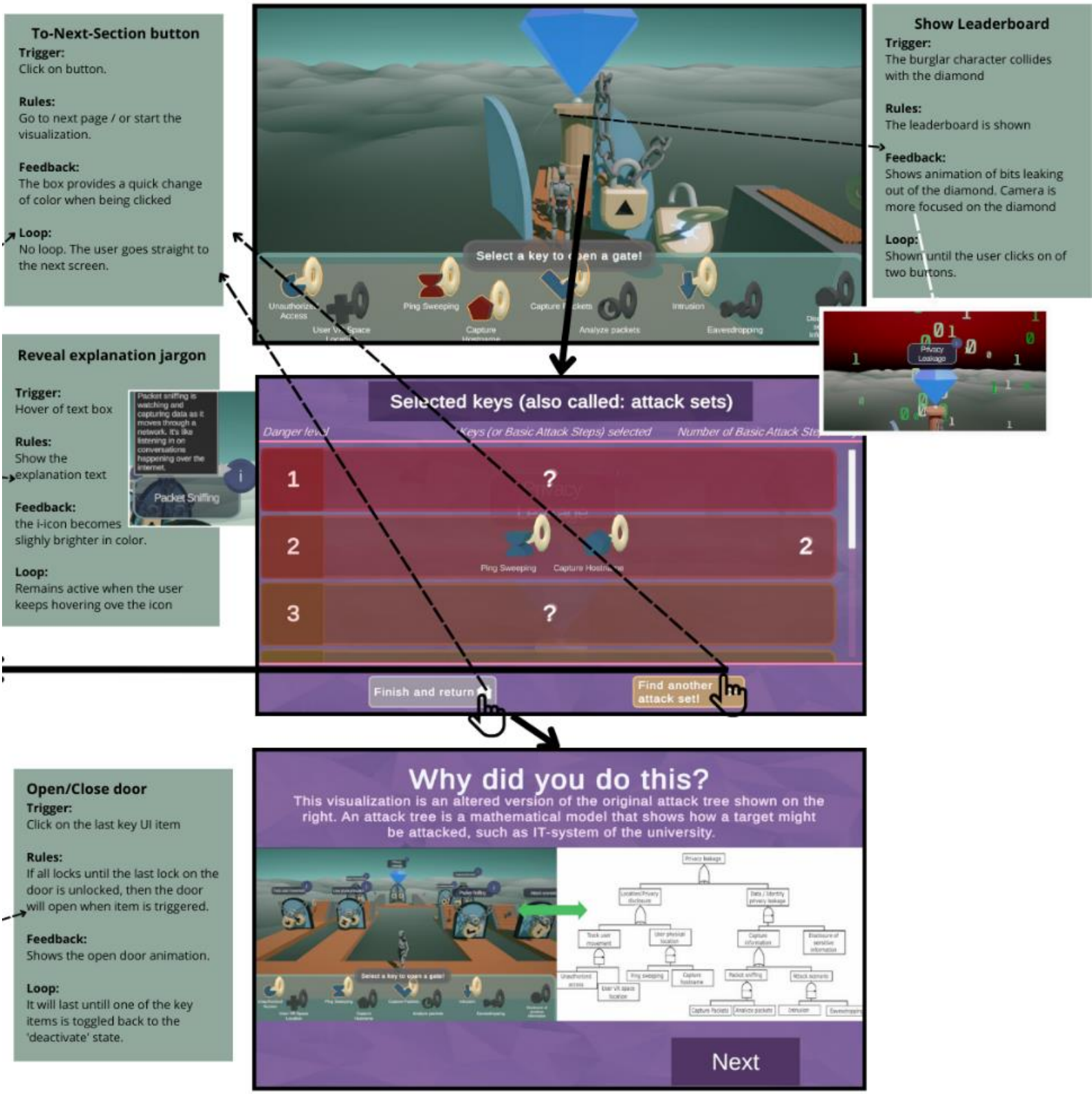


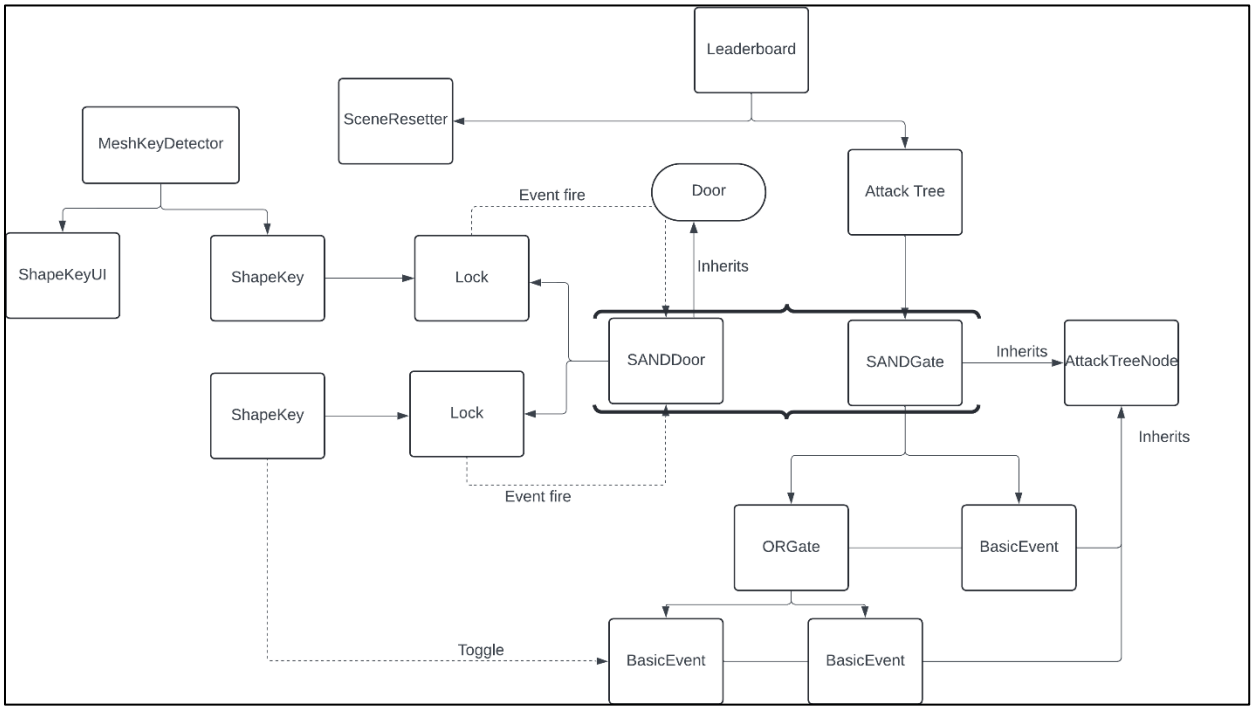
Figure 23: A more zoomed-in version of the interaction flow map, for readability purposes. To improve readability further, lay this page and the previous one next to each other, just like in figure 21.

7 Realization

Now that the requirements for the design are specified, the building process for the new design of the attack tree can begin. Reporting on the building process is important, because it shows others how they can potentially recreate this visualization.

As mentioned in the specifications, many software components, such as Unity and Blender, are utilized for the building process. These software components have their own internal workflow, which can be confusing to those with no knowledge about these libraries and components. While this chapter may explain part of this workflow, not all details will be covered due to the scope of the graduation project. The explanations will be supported with screenshots and code snippets. To view the full application, refer to the UTwente archive.

7.1 System architecture



Schema 1: The general overview of the system architecture.

7.1.1 Attack tree architecture

In coding, the qualitative attack tree as a whole can be seen as a list of nodes, with each node referencing another node, and having its own behavior. This behavior depends on the type of node, which can be a Basic Attack Step (BAS), a Sequential-AND gate (SAND gate), or an OR gate. Regardless of their different behaviors, each node has child nodes (except for the BASs). Therefore, each of these nodes can be derived from an abstract superclass, making the code more manageable and scalable. The reason it is an abstract class is that a generic node is not part of the attack, but a BAS, SAND gate, and OR gate is. Below is a code snippet of the attack tree node. The functions appear self-contained and are therefore not explained further.

A node in the Attack tree

```
public abstract class AttackTreeNode : MonoBehaviour
{
    public abstract List<HashSet<BasicEvent>> GetCurrentTrueEvents();
    public abstract void GetAllBasicEvents(List<BasicEvent> basicEvents);

    public abstract bool IsAttacked();
    public abstract string GetMathematicalRepresentation();
}
```

Snippet 1: The AttackTreeNode class.

Then each specific element derives from this super class as seen below.

Basic Event Class:

```
public class BasicEvent : AttackTreeNode, IEquatable<BasicEvent>
{
    //public string eventName;
    public bool isTriggered { get; set; }

    public override bool IsAttacked()
    {
        return isTriggered;
    }

. . .

}
```

Snippet 2: The basis of the Basic Event class. The class also inherits from IEquatable interface so that the BasicEvent can be utilized for comparison equations. This is necessary for later.

OR Gate Class

```
public class ORgate : AttackTreeNode
{
    public List<AttackTreeNode> children = new List<AttackTreeNode>();

. . .

}
```

Snippet 3: The OR gate is a subclass from the AttackTreeNode. References are made to their associated child nodes.

SAND-Gate Class

```
public class SequentialANDgate : AttackTreeNode
{
    public List<AttackTreeNode> children = new List<AttackTreeNode>();

. . .

}
```

Snippet 4: The SAND gate is a subclass from the AttackTreeNode. References are made to their associated child nodes.

In the snippets, it is not explained how the gate classes obtain a reference to each of their child nodes. This is done in the Unity Inspector. Since all of these nodes reside on a GameObject in the scene, the programmer can simply drag the gameobject into the inspector of the class, as displayed in the figure below.

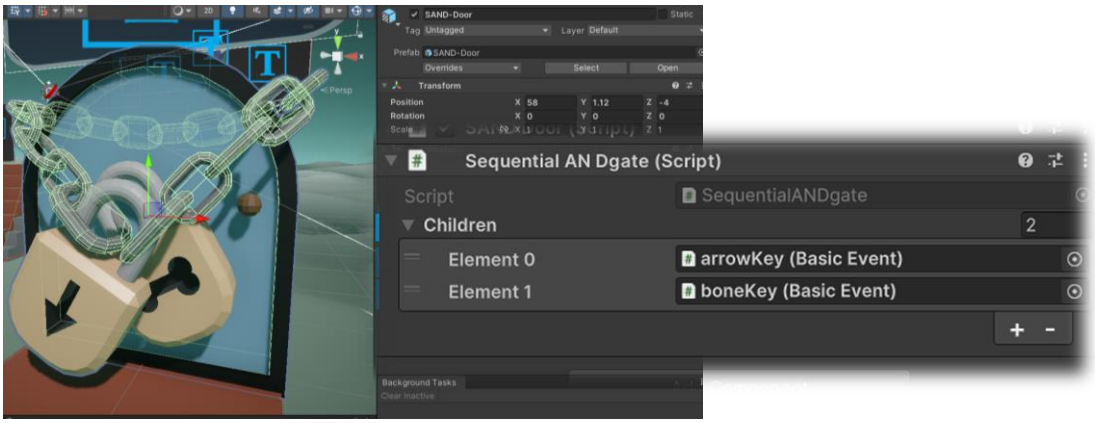


Figure 24: The BASs, bonekey and arrowkey, are dragged in the inspector onto their associated parent.

Then the full attack tree class just only needs a reference to the top attack tree node to complete the relationships between all the nodes. The root node can also be dragged and dropped to the component in the Unity inspector.

Attack Tree

```
public class AttackTree : MonoBehaviour
{
    public AttackTreeNode root;

       . . .

    private List<BasicEvent> basicEvents;
    public List<HashSet<BasicEvent>> AllRankedCutSets { get; private set; }

    private void Awake()
    {
        basicEvents = new List<BasicEvent>();
        root.GetAllBasicEvents(basicEvents);

        List<HashSet<BasicEvent>> allCutSetsEvents = GetAllCutSets();
        AllRankedCutSets = allCutSetsEvents
            .OrderBy(set => set.Count)
            .ToList();

       . . .

    }

}
```

Snippet 5: The AttackTree class: It holds a reference to the root node, all the BASs (basicEvents), and the attack sets in the attack tree.

From this root node, a list of all the BASs can be created by using `root.GetAllBasicEvents(basicEvents)` function. A reference to these BASs is convenient for resetting all states of the BASs to false. Additionally, it is useful for generating unique situations in the attack tree, as the programmer can simply set some BASs to true, while set other BASs to false.

As seen in the abstract class, each `AttackTreeNode` implements this function as a depth first traversal from the root node to the leaf nodes. This traversal is achieved

with recursion, with each node calling their child nodes' function. This approach is chosen, because it is relatively simple to implement in code, and no performance optimization is needed in this context, as the tree is relatively small.

With this, all attack sets can be found by checking every situation of the BASs. In other words, the `GetAllCutSets()` uses a brute-force method to find all attack sets to ensure simplicity of the code in the application. In the current context of 9 BASs, 2^9 situations can be created in the Attack tree.

Attack set method

```
private List<HashSet<BasicEvent>> GetAllCutSets()
{
        List<HashSet<BasicEvent>> allCutSets = new List<HashSet<BasicEvent>>();
        int numberOfCombinations = 1 << basicEvents.Count; // 2^basicEvents.Count

       . . .
}
```

Snippet 6: the 'brute-force' method for finding all attack sets.

The attack sets are found by calling the `GetCurrentTrueEvents()` function from the root node. This function also uses a depth-first traversal to check whether there are true BASs that can reach the root. However, this method can only find minimal attack sets, not all attack sets in general. This is because the function stops searching in the child nodes when the output of that specific node returns false. Figure 25 illustrates this problem.

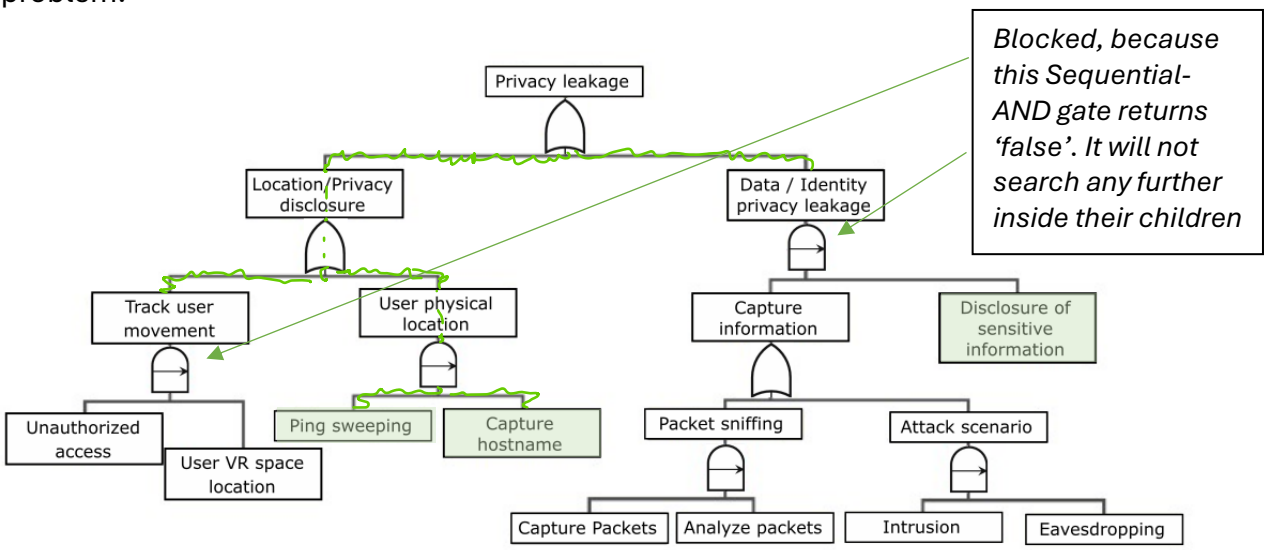


Figure 25: Demonstration of the search method applied. The tree stops searching further in nodes that return 'false', such as the 'Track User Movement' and 'Data / Identity privacy leakage' nodes. The BASs highlighted with the green colour are true.

In the previous snippet, all the found attack sets are ranked based on the number of BASs in the set. This is useful for the leaderboard implementation explained later.

7.2 Attack tree elements

7.2.1 Shape Keys

Each key shape is modelled in Blender and only the shape is exported to Unity. Since the casing or loop of the key remains the same, only the model is necessary for exportation to Unity.

Each of this Shapekey GameObjects is then attached with a `Shapekey` script. Consequently, this class has a reference to the lock GameObject that resembles the shape of the key. This linkage is done in the Unity Inspector.

This class has many tasks, which can be considered a software design flaw. Its particular jobs are to set the BAS to true or false, animate the ShapeKey GameObject and trigger the animation of the lock when toggled by either picking it up, or clicking the UI element. In the future, this could be improved.

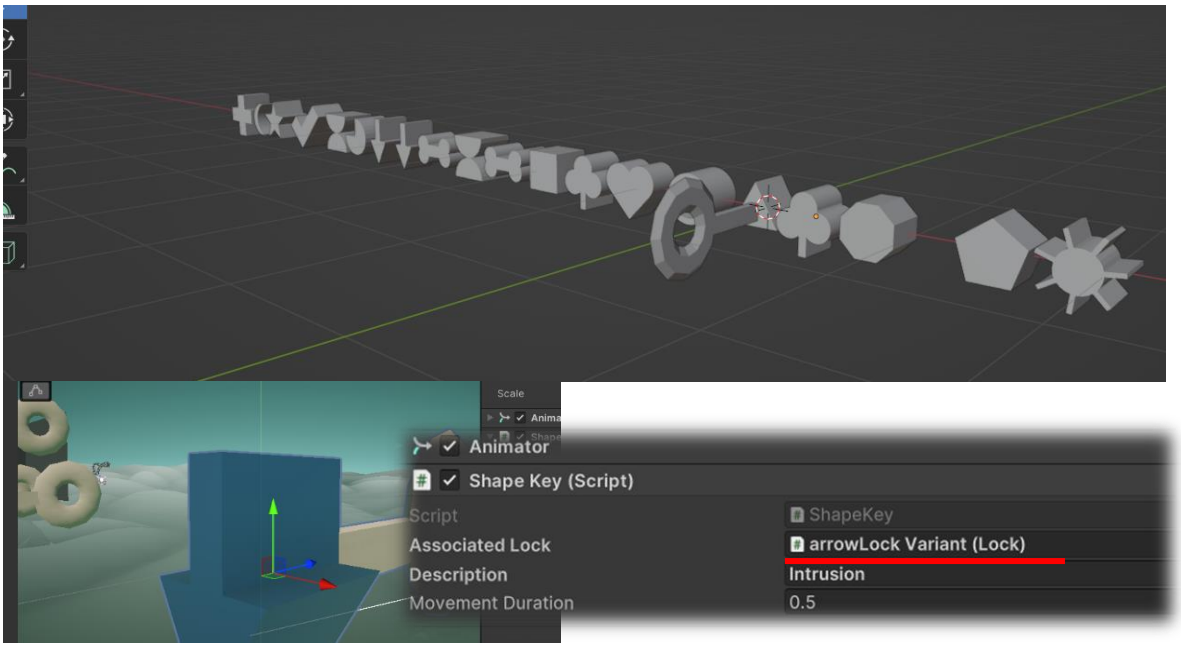


Figure 26: All the shapes are modeled in Blender. In this example, the `arrowkey` has a reference to the `arrowLock`.

The `ShapeKey` Class

```
public class ShapeKey : MonoBehaviour
{
    [SerializeField] private Lock associatedLock;

. . .
    public void ToggleKey(Vector3 UIposition)
    {
       . . .
        basicEvent.isTriggered = !basicEvent.isTriggered;

        if (associatedLock.IsOpen)
        {
            transform.position = UIposition;
            StartCoroutine(MoveKeyToTarget());
        }
        else
        {
            ResetKey();
        }
    }

}
```

Snippet 7: The `ToggleKey` function inside the `ShapeKey` script. As each UI element represents a BAS, the `basicEvent.isTriggered` is set to true (or false depending on its previous state). The Coroutine initiates the animation of the key.

Most of the animation is done in a coroutine `StartCoroutine(MoveKeyToTarget())`. A Coroutine in Unity is a function that can pause execution and return control to Unity. This is convenient for when the programmer wants to spread the task across multiple frames. Animation is almost always done over multiple frames. Defining our animations through code instead of using a timeline with keyframes provides the flexibility to move the `ShapeKey` from any position towards their specific target lock. This flexibility is needed as `ShapeKeys` can be positioned anywhere in the scene.

Coroutine for animating the keys

```
private IEnumerator MoveKeyToTarget()
{
      . . .
    float elapsedTime = 0f;

    while (elapsedTime < movementDuration)
    {
        float factor = elapsedTime / movementDuration;
        factor = Mathf.SmoothStep(0, 1, factor);
        transform.position = Vector3.Lerp(initialPosition, targetPosition, factor);

        elapsedTime += Time.deltaTime;
        yield return null; // Pause execution until the next frame
    }

    transform.SetParent(associatedLock);
    associatedLock.ToggleLock();
}
```

Snippet 8: This function animates the key towards the target. `Yield return null` pauses the execution of the code until the next frame. The `transform.position` of the key is linearly interpolated between its start and end position, which based on the amount of time that has elapsed in the scene.

The coroutine is defined with `IEnumerator` and `yield return` somewhere in the function. The position of the key is linearly interpolated between their start position and the position of their associated lock, with the elapsed time as the determining factor for this position. After the animation of the key has been completed, the lock will execute their animation.

7.2.2 Locks

A lock does not have much functionality other than animating and firing an event to the door when the animation is completed. Additionally, locks also apply coroutines to move them to the side. This event is utilized to notify the door that a lock has been unlocked and that it should check whether it should open. The reason that the lock fires an event instead of the instance holding a reference to the door is to prevent circular dependency. The doors already have references to the locks for the purpose of initializing, setting and checking their status.

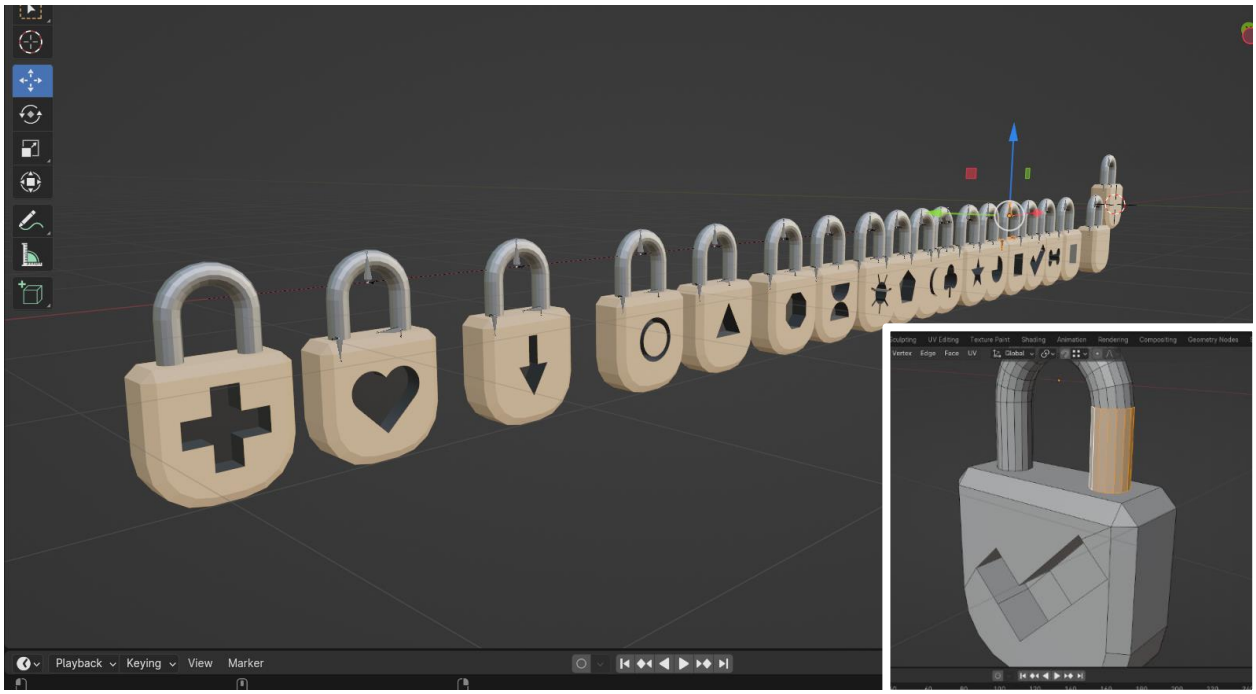


Figure 27: Modeling process of the locks

Lock class

```
public class Lock : MonoBehaviour {
    public event Action<Lock> OnLockToggled;
    . . .
    public void ToggleLockLogic()
    {
        OnLockToggled?.Invoke(this);
       . . .
    }

. . .

}
```

Snippet 9: Invoking the event when lock is being opened.

7.2.3 Doors

In the attack tree, there are two types of doors: SAND doors and OR doors. They both are subclasses from the *Door* superclass. As mentioned in the lock section, doors have references to the locks. The doors subscribe to the `OnLockToggled` events of the locks that reside on the door. The door will open when both the `OnLockToggled` event is fired and when all the locks on the door are unlocked.

Door class

```
public class Door : MonoBehaviour
{
   . . .
   protected virtual void Start()
   {
        if (locks != null)
        {
            for (int i = 0; i < locks.Length; i++)
            {
                locks[i].OnLockToggled += ToggleDoor;
            }
        }
   }

   public virtual void ToggleDoor()
   {
        SetDoorOpenAfterUnlock();
        ActivateAnimation();
   }
 . . .
}
```

Snippet 10: The Door class. These instances subscribe to the lock event.

7.2.3.1 Stairs

In this prototype, a key is floating behind every door, which fits on the next door the burglar encounters. Conversely, this setup has a problem when two paths converge at a new gate. When the burglar opens one of the gates, he can walk around to the door on the same layer and access the key he is not supposed to pick up. This problem is also illustrated in figure 28.

To solve this issue, dynamic stairs have been placed behind every door so that the burglar cannot access these keys. These stairs indicate one way direction. Please refer to figure 29 for the stairs.

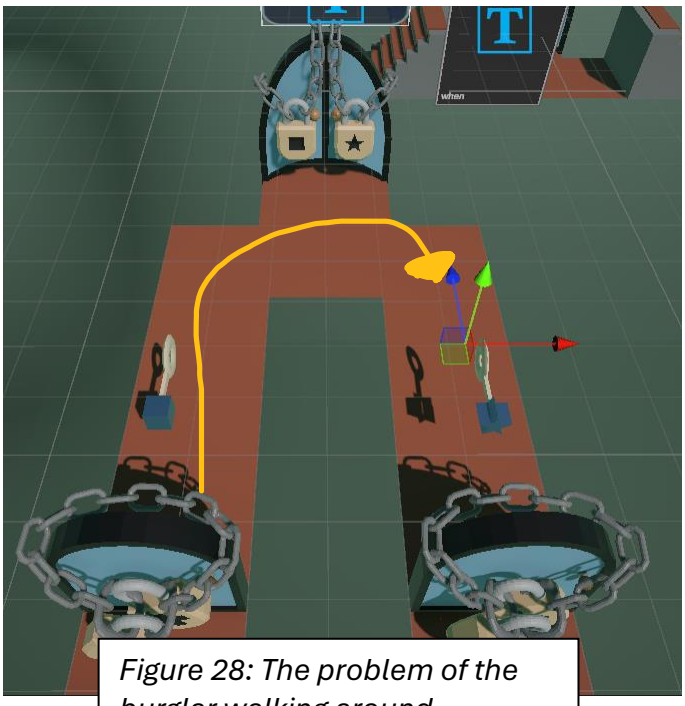


Figure 28: The problem of the burglar walking around

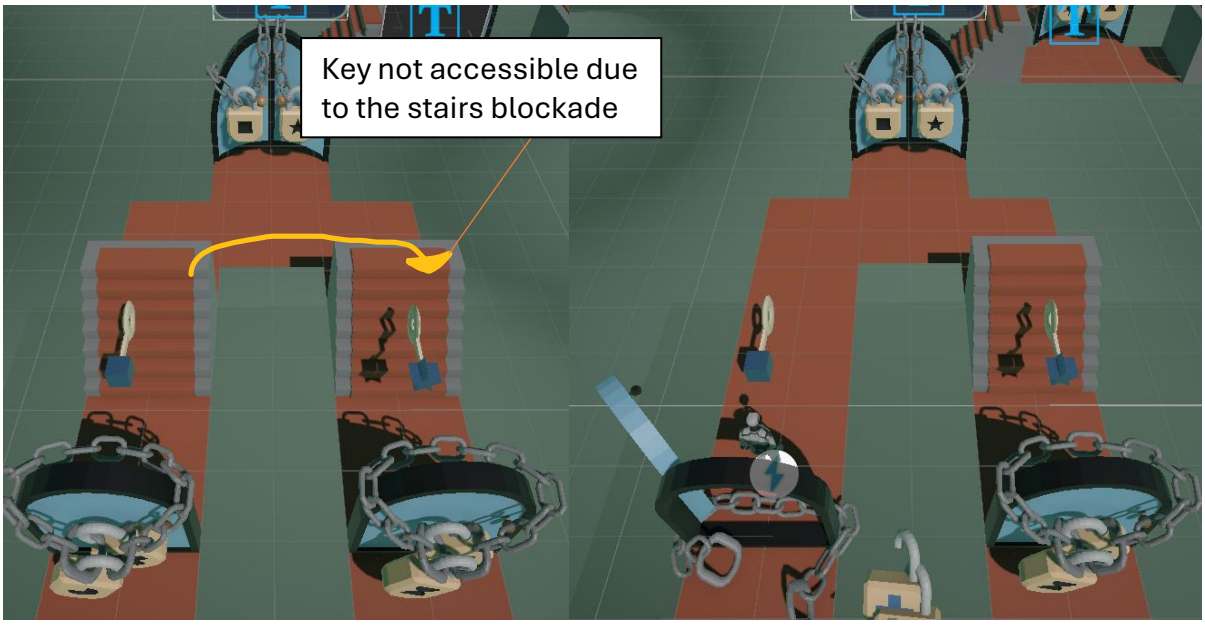


Figure 29: Stairs prevent the burglar from the keys he is not supposed to pick up.

These stairs are made dynamic. That means that the stairs will disappear when the door behind it opens. This gives the user the opportunity to go back and analyse the other doors when they want to do that.

7.2.3.2 Chains

The visual and behavioural aspects of the chains are important. If the chains are not prominently visible, the user might misinterpret the analogy of the doors and locks with the logic gates. Specifically, it may appear that the locks are not connected to the door at all, leading to the incorrect assumption that the door is open when it is.

The chains around the doors have Rigidbody components. These components enable movement in the physics engine of Unity, allowing gravity and other forces that can be applied to move objects as they would move in the real world. Each chain consists of 18 chain part GameObjects, each programmed to move each individual part through the physics engine. This gives a more natural movement to the chains. And yet, each part of the chain is not completely free, because it is connected to each other by Hinge Joint component. This component links two rigidbodies together to constrain their movements with respect to each other, which in simple terms means that each chain part is connected to the previous chain part.

The entire chain carries a custom ChainController script, which is responsible for controlling the forces applied to the chain parts. Before the door is opened, no forces, such as gravity, are applied to the chain. When the door is about to open, gravity on all Rigidbody chain parts is activated, causing them to fall naturally.

ChainController Class

```
public class ChainController : MonoBehaviour
{
    private Rigidbody[] chainLinksRigidbodies;

. . .
    public void ActivateGravity() {
        if (resetCoroutine != null) {
            StopCoroutine(resetCoroutine);
            resetCoroutine = null;
        }

        ResetTransformOfAllChainLinks();
        SetGravityChain(true);
    }
    private void SetGravityChain(bool useGravity) {
        for (int i = 0; i < chainLength; i++) {
            chainLinksRigidbodies[i].useGravity = useGravity;
            chainLinksRigidbodies[i].isKinematic = !useGravity;
        }
    }
. . .
}
```

Snippet 10: The ChainController class. It has a reference to all parts of the chain and will activate the gravity on these objects when the door is opened, causing the chain to fall down.

7.2.3.3 Sequential AND door

The Sequential-AND door overrides the superclass door. The `SANDDoor` class adds the functionality of updating the `CanBeUnlocked` status of the locks. This Boolean applies a rule that the locks can only be unlocked in a certain order, just like a sequential AND-gate. The `SANDDoor` maintains an index variable maintaining the information about how many locks have been opened already and until which unlocked lock.

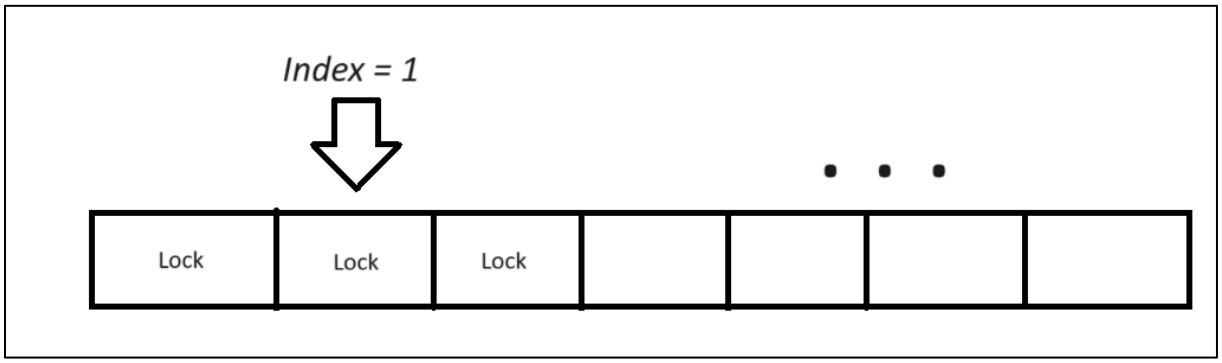


Figure 30: Every lock up to and including index 1 is opened. The order of opening the locks is from left to right.

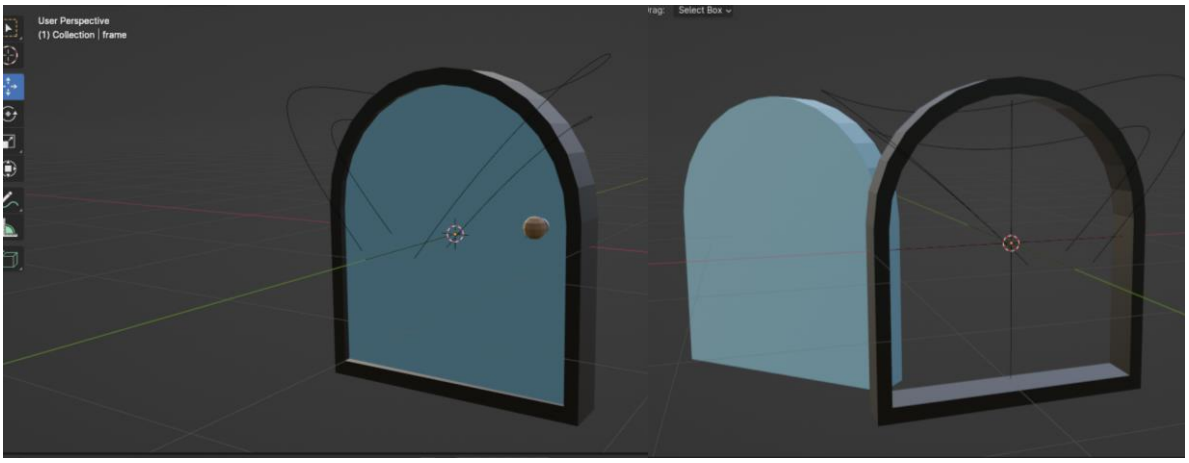


Figure 31: Modelling process of the SAND door.

7.2.3.4 OR-door

The OR-door just have two `Door` script components on either door object. Therefore, only changes regarding the visual representation of the OR-door must be made.

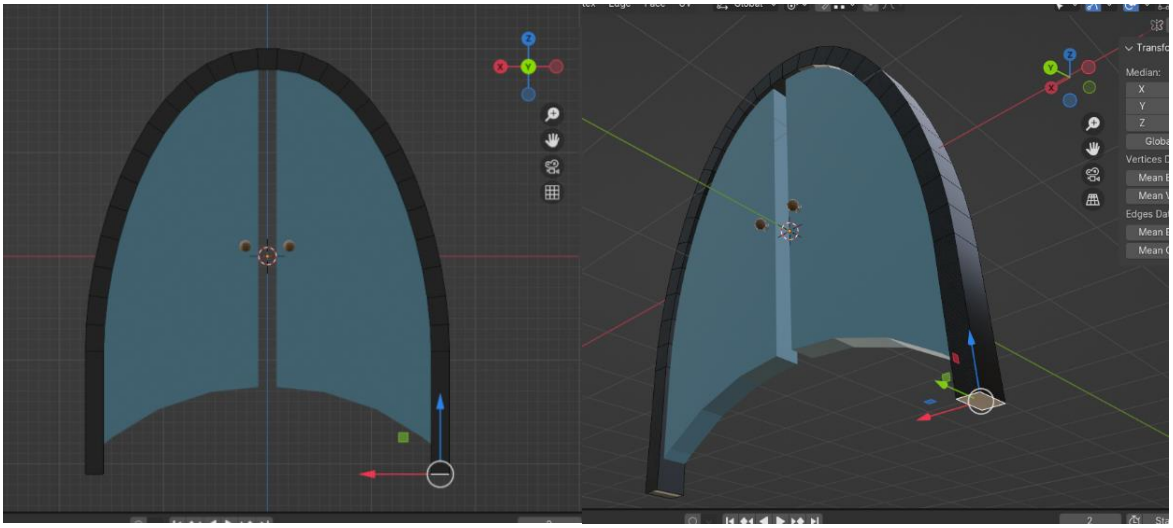


Figure 32: Modelling process of the OR door.

7.2.4 Root

There are several reasons why the root of the attack tree should behave differently from the other nodes. Firstly, the root should be prominently visible so that the user can clearly see his goal of finding the optimal path toward it. Secondly, the root should indicate that reaching it is damaging for the system in question. Thirdly, the root should raise some emotional responses in the users for the purpose of keeping them engaged.

To satisfy the first condition, a large diamond has been modelled, because diamonds generally hold significant value. The second and third condition is met by using Unity's particle system. The diamond will emit bits for a few seconds when the burglar reaches the root. This represents sensitive data, such as passwords, being exploited or shared with others. The combination of this effect with the red alarming environment should bring the user some emotional response and understanding about the severity of reaching it.

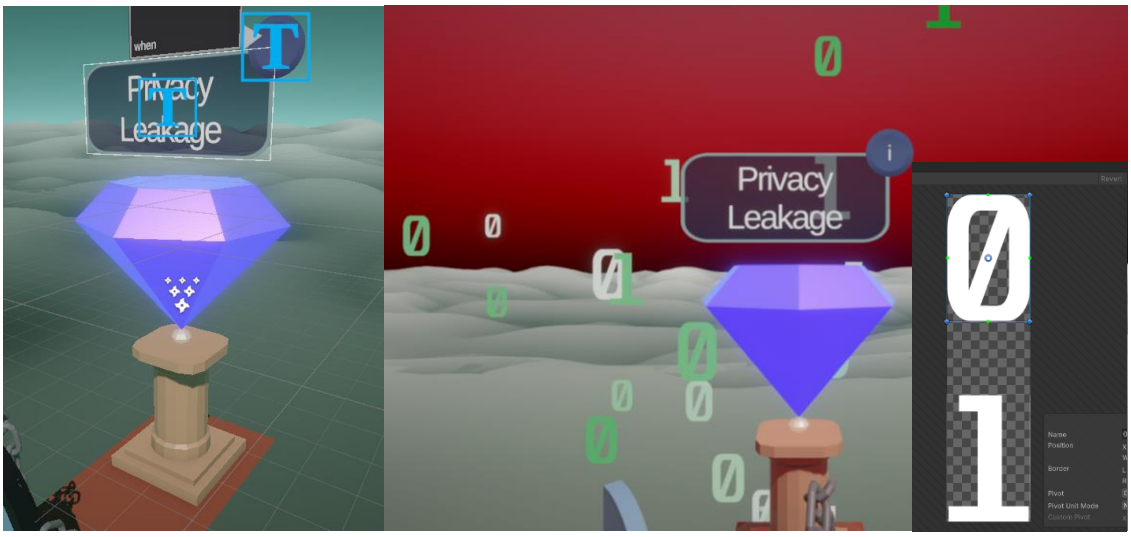


Figure 33: The root. The diamond will emit bits when the burglar reaches the root

7.3 User Interface

The user interface (UI) of our application is designed to be intuitive and to improve the overall usability. The UI of this prototype consists of three canvas

7.3.1 Home Screen and Debriefing

The user starts their experience with the home page in which it is explained what the controls and goals of the visualization are. The user only must read the text and click on the purple buttons to go through the next canvas. Because this interaction is simple, the code is not explained further.

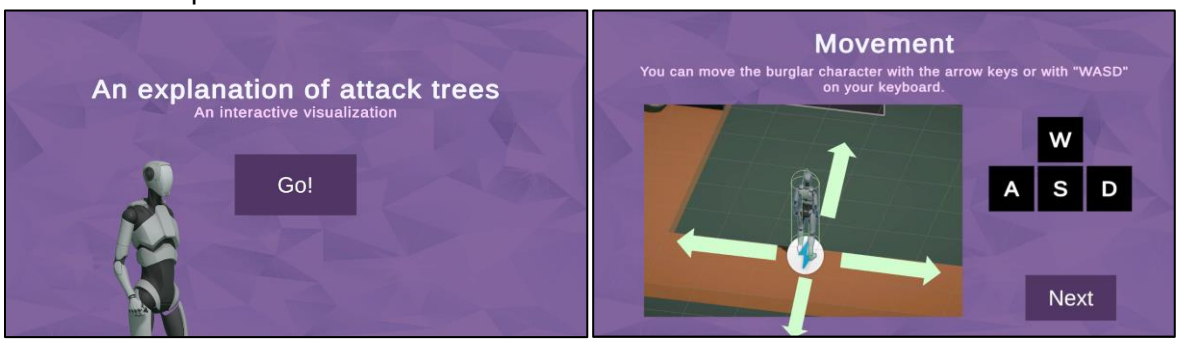


Figure 34: The Home screens explaining the goals and controls

Upon finishing the experience, the user receives a final explanation of the relationships between the original attack tree design and the current design of the prototype: the debriefing phase. The interactions in the debriefing phase are the same as in the home page.

7.3.2 Leaderboard

A leaderboard of the completed (minimal) attack sets is displayed to the user when the burglar successfully reaches the root. This leaderboard encourages the user to reflect on their own actions by showing them their selected BASs. They can assess further how well they did in finding the most severe attack set. The most severe attack sets, the ones with the least keys in the set, are shown at the top of the leaderboard, while the attack sets that contains a quantity of keys are placed at the bottom. This ranking is amplified

by a color scheme. The items in the top region are red, which slowly descends to green for the lower items in the leaderboard. These colours are chosen as red is generally perceived as ‘bad’ and ‘dangerous’, while green is usually seen as ‘good’ and ‘safe’. The leaderboard only shows the attack sets found by the user, possibly motivating the user to find more severe attack sets in the attack tree.

Coloring of leaderboard items

```
uiBG.color = Color.Lerp(startColor, endColor, (float)i / numberOfCutSets);
```

Snippet 11: This line linearly interpolates between red and green. The lower the item in the leaderboard, indicated by the ‘i’ variable, the greener the items appear in the leaderboard.

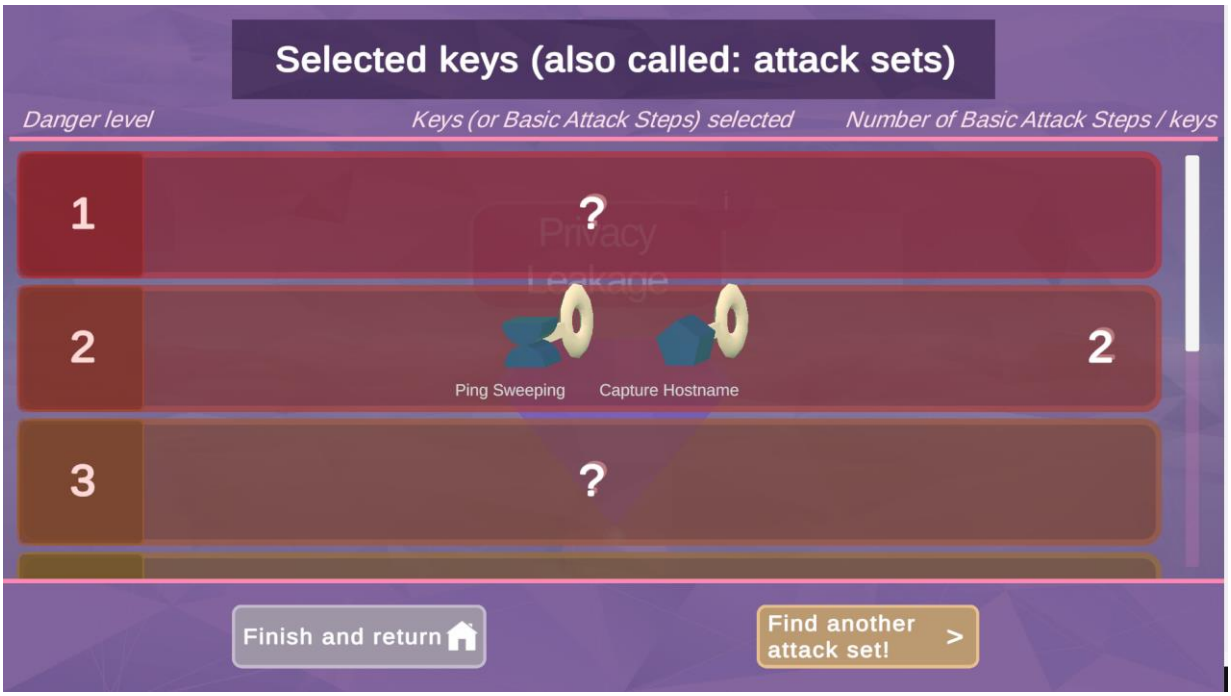


Figure 35: The leaderboard screen.

There are several functionalities that make it easy for the developer to change the visual representation of the leaderboard.

Firstly, the developer can choose how many items to display in the leaderboard. This is done in the `GenerateItem()` function. In this function, a leaderboard item is initially added and set to unknown or ‘not found’, shown as a question mark. Each leaderboard item consists of a prefab that defines its structure, its outline of the item, the descriptions of the keys and the instances of the visual representation of the keys. Because there is a prefab, it is easy to switch between the number of items the leaderboard should show. This functionality is shown in snippet 12.

Leaderboard class

```
public class Leaderboard : MonoBehaviour
{
    private List<HashSet<BasicEvent>> foundCutSets;
    private AttackTree attackTree;
    private int numberOfCutSets = 6;

    private void GenerateItem()
    {
        int i = 0;
        foreach (HashSet<BasicEvent> basicEvents in attackTree.AllRankedCutSets) {
            GameObject item = AddItem(basicEvents);
            SetItemToUnknown(item, i);

            if (i >= numberOfCutSets)
            {
                break;
            }
            i++;
       }
    }

   private GameObject AddItem(HashSet<BasicEvent> basicEvents) {
        GameObject item = Instantiate(itemPrefab, items);
        Transform keys = item.transform.GetChild(0);

        foreach (BasicEvent basicEvent in basicEvents) {
            GameObject key = Instantiate(keyPrefab, keys);
            GameObject keyAppearance = Instantiate(basicEvent.gameObject, key);
       . . .
        }

        return item;
    }
}
```

Snippet 12: the `AddItem` function is highlighted here. This method instantiates the item prefab and their associating key prefabs (i.e.: the visual representation of the key) for each attack set.

Secondly, the leaderboard can update an unknown attack set to a found attack set. This is possible because the object holds an instance variable containing a list of all found attack set. When the root is reached, the attack tree returns the found attack set from the scene, adds this set to the list of found attack sets, and determines its index in complete list of all attack sets. The item at this index is then revealed to the user.

At the bottom of the screen, two buttons are displayed. The left button initiates the debriefing phase and resets the scene, which can be automated using Unity's Scene Manager. However, resetting the entire scene to find a new attack set does not suffice for this application because it would delete information about the found attack sets. Therefore, an additional SceneResetter class has been created. This class is responsible for resetting every element in the scene while preserving essential user information.

7.3.2.1 Resetting the Scene

At the bottom of the screen, two buttons are shown. The left button initiates the debriefing phase and then resets the scene, which can be automatically done with Unity's Scene Manager. Conversely, resetting the entire scene to find a new attack set does not suffice for this application, because it would discard the information of the found attack sets. Therefore, an additional `SceneResetter` class has been created, which is responsible for resetting every element in the scene while also preserving the user information.

Figure 36: The leaderboard buttons.

7.3.3 Key selection

The real gameplay contains a key selection dashboard at the bottom of the screen, which is designed to implement various design choices that should make its working intuitive.

Firstly, each item that is selectable has been given an orange outline, to complement the neutral blue color of the keys. This makes the element pop more on the screen. Emphasis on this is important as selecting the keys is an important interaction for this visualization.

Secondly, the keys are organized based on the location in the original attack tree, helping the mapping for the user. The BASs directly connected to a Sequential-AND gate are grouped together in the UI. This is done by positioning the BAS that should be true before the next BAS, higher on the screen. For most users, this is natural as usually information is read from left to right and top to bottom.

Thirdly, the BAS that cannot be selected are greyed out, as the Sequential AND-gate requires an order of execution. This visual cue help to convey the structure of the Sequential-AND gate, possibly making it more intuitive for the user.

Lastly, a `ShapeKey` turns red when it is being selected by the user. This provides instant feedback to the user about the status of the BAS. The color red is chosen because red suggests danger, which align to the nature of the attack tree.

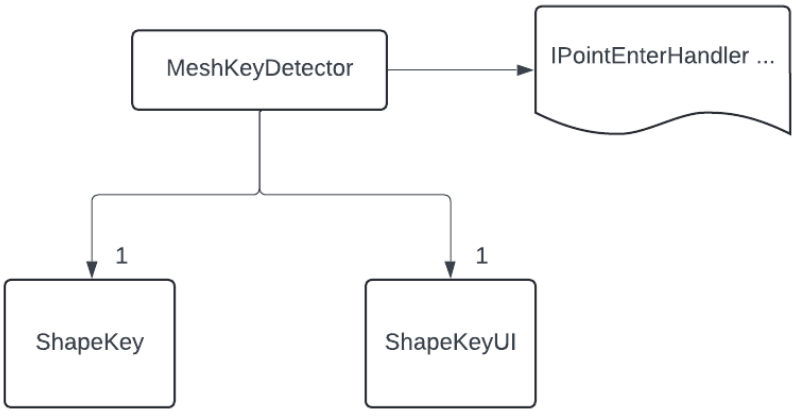


Figure 37: The structure of the selectable keys displayed at the bottom of the screen.

To select a key, the `MeshKeyDetector` class is applied to these UI elements. This class inherits from the `IPointerEnterHandler`, and `IPointerClickHandler` intefaces from Unity's event system. This inclusion ensures detection of the mouse hovering over or clicking on the mesh. This class also maintains a reference to the `ShapeKey` class and their asscíated UI class to handle the communication between the two. UI and the `ShapeKey` classes are seperated into their own classes to maintain code clean and organization. Clicking a key toggles the corresponding `ShapeKey` and therefore automatically updates the entire attack tree.

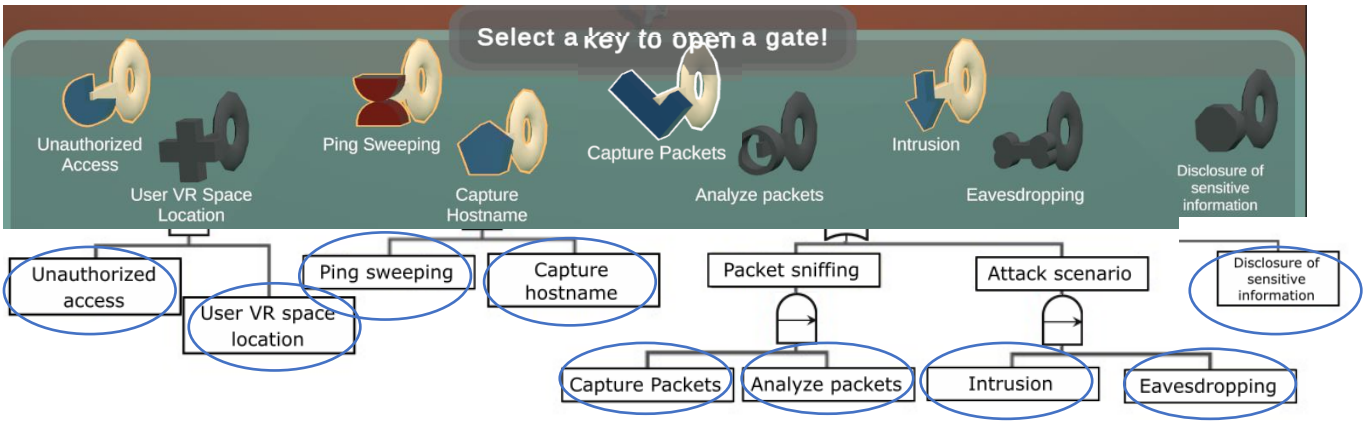


Figure 38: An overview of the relationsships between the BASs and the selectable keys.

7.3.3.1 Key collectible

Besides selecting the keys referencing the BASs, there are also keys that can be picked up in the child nodes. These keys maintain the parent-child relationships between the nodes. They float in the air, move in a sine wave pattern and have a box collider, so that they can be picked up upon collision with the burglar.

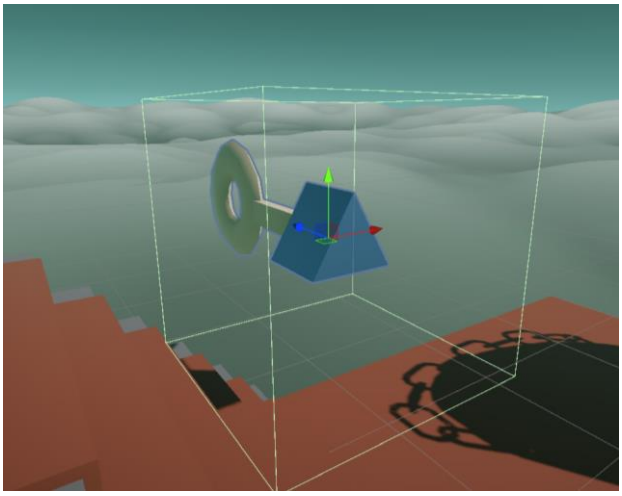


Figure 39: A floating key that can be collected by the burglar. The green box resembles the collision area.

7.3.3.2 Shape Key Canvas

Lastly, above each door is small textbox. This canvas has a billboard functionality, meaning that the text will always face the camera. Additionally, users can hover over these text boxes to access more information about it when they do not understand the technical jargon inside the box. This hover functionality is implemented using Unity Events.

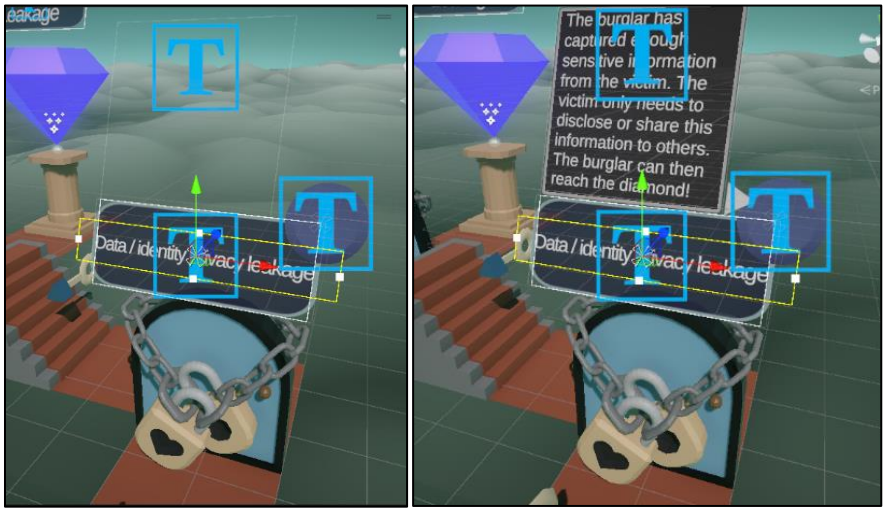


Figure 40: the left image shows the plain text boxes, while the right image also displays the explanation of the technical terms.

7.4 Shader visualization

A distinction has been made between walkable and non-walkable paths in the attack trees. To make this distinction very clear, the non-walkable floor has been designed as clouds, while the walkable paths are made like red carpets.

Creating an object that moves and looks like a cloud is challenging. To convince the audience with realistic clouds, a shader has been applied on a highly detailed plane containing approximately 25.000 vertices. The number of vertices is high, because these vertices need to be moved in a vertical direction by the custom shader. When the vertices density is low, then the clouds appear pixelated and unrealistic.

The shader is created with Unity's Shader Graph. The benefit of using a Shader Graph is that the developer does not need to code the shader in HLSL, which requires much knowledge and expertise about math and scripting notation.

The basis of the shader consists of a gradient noise pattern that moves over time over another gradient noise. The addition of two noise texture together removes the predicted behaviour of the movement of clouds.

The creation of this shader is based on the Papush's video [56]. For the full shader, refer to the appendix.

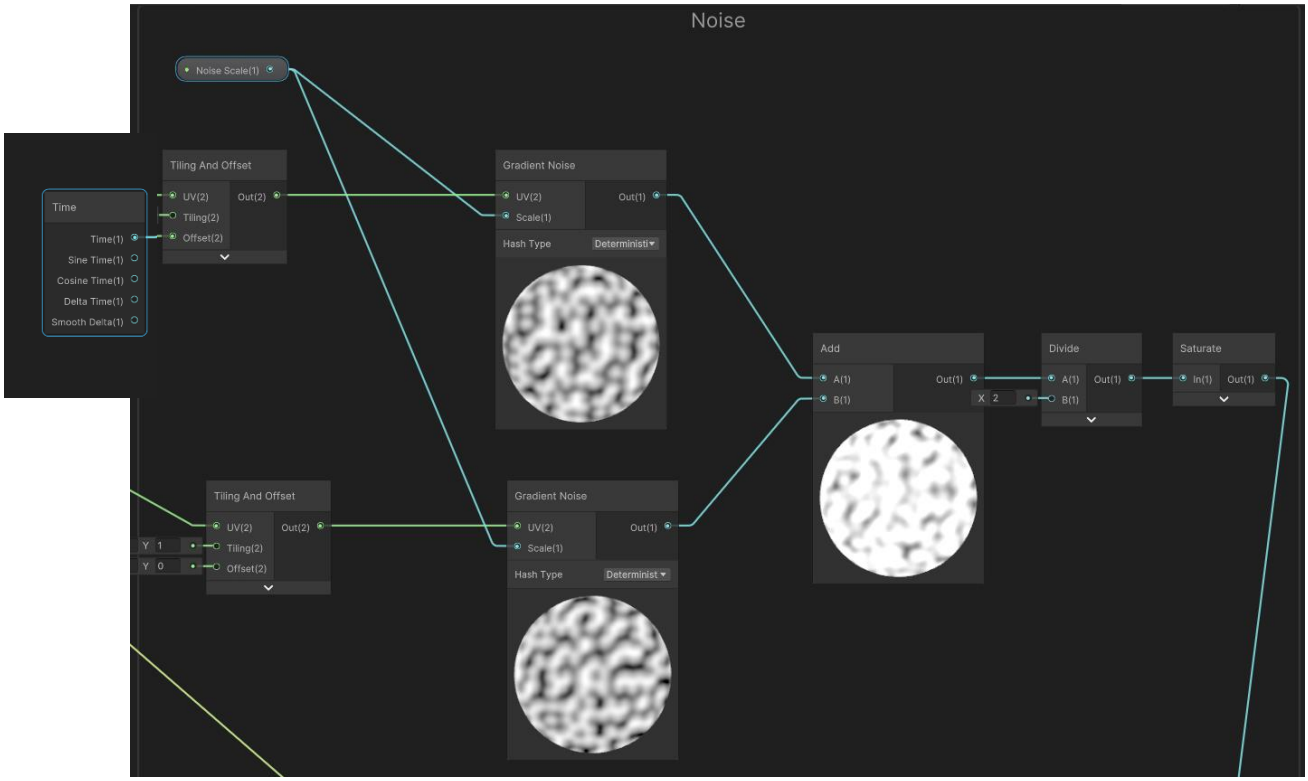


Figure 41: A moving noise texture (top element) is added with a still noise texture. The combination of is then normalized to values between 0 and 1, as colour values cannot exceed above the white colour which is (1, 1, 1) in unity.

These noise textures are then further modified with other shader graph nodes, such as colouring, until a realistic noise pattern has been found.

After that, the noise texture is transformed to data that each individual vertices in the plane can read. This is done by multiplying the noise texture, which consists of pixels between 0 and 1, with the normal vector of each individual vertex in the plane. The height of these pixels can then be modified through a parameter in the inspector. The heigher this value, the greater the heights difference is within the clouds. Lastly, the vectors that are created do not initially have a position. That means that the vector originate from the absolute zero point in the vector space. Therefore, the position of these vertices is added to the normal vector at the end.

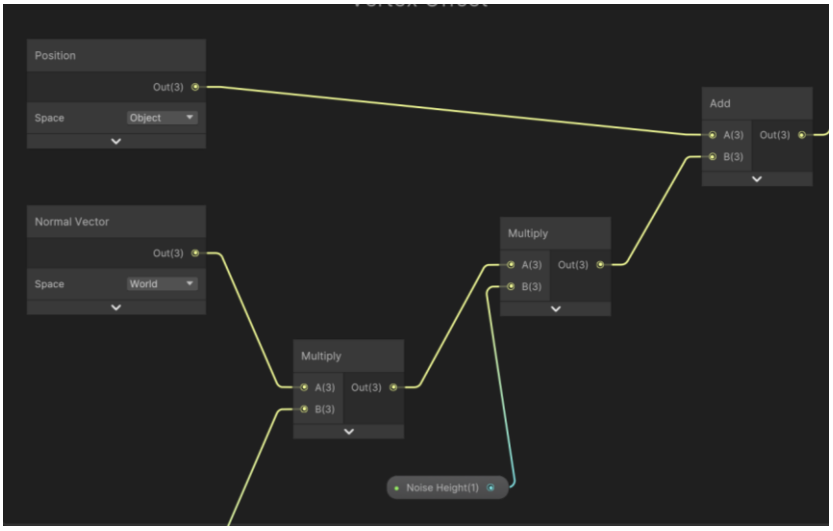


Figure 42: The normal vector is multiplied by the noise texture in the bottom-left node. At the end, all the normal vectors are translated to their position in the scene.

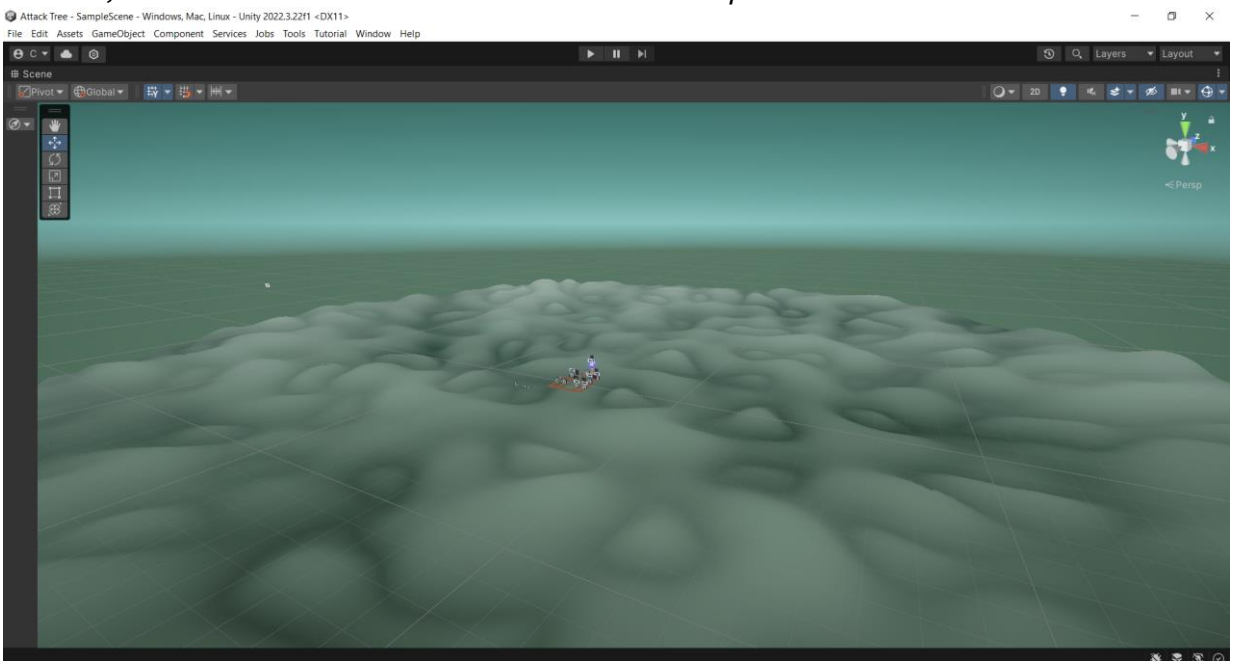


Figure 43: The result of the clouds.

7.5 Character development

The character used in this visualization is downloaded from the Unity's Asset Store. A moving character is so common in game engines that reinventing the wheel does not make sense.

7.5.1 Cinemachine

The camera movement for the character, however, cannot be premade. Unity offers external package called *Cinemachine*, which simplifies the developing of camera movement behaviour. *Cinemachine* works with virtual cameras, which are just empty *GameObjects* with predefined settings that tells the actual camera in the scene how to behave. The camera's primary task is to follow the character and always point in the direction of the x-axis.

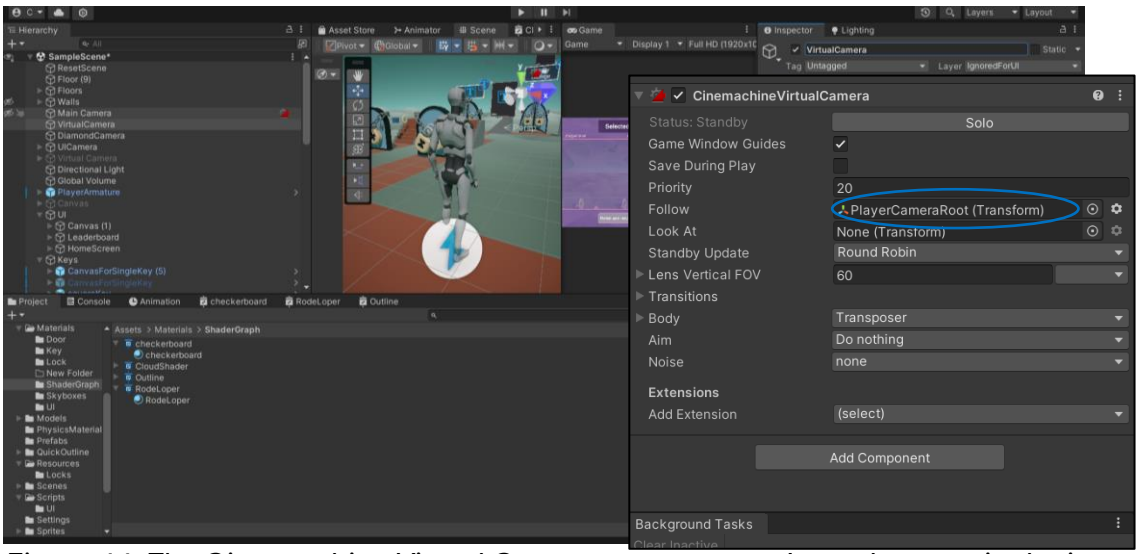


Figure 44: The Cinemachine Virtual Camera component. It can be seen in the inspector that the Virtual Camera follows the Player

7.6 Invisible fences

To prevent the character from falling, invisible box colliders have been placed at the edges the walkable paths. This is shown in figure 45.

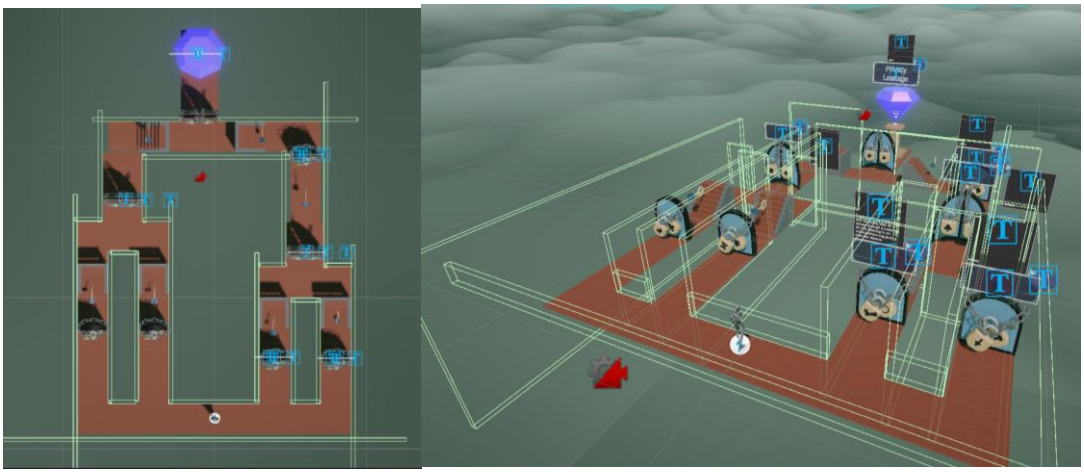


Figure 45: Setup of the box colliders to prevent the character from falling off the paths

8 Evaluation

The goal of this project is to evaluate whether the built interactive visualization is more effective in communicating qualitative attack trees than traditional communication.

Therefore, a between-subject design was tested on this effectiveness, in which one group was tested with the interactive visualization, while the other group was tested with a textual explanation of attack trees. A between-subject instead of a within-subject design was chosen, because familiarity and learning of the material of attack trees could lead to better results in later stages of the test. It is hypothesized that the interactive visualization will lead to better learning and communicating outcomes than the control group.

Furthermore, a qualitative analysis has been conducted to test on the overall usability of the prototype.

8.1 Experimental design

8.1.1 Subjects

A total of 23 participants were recruited for this project. Participants were randomly assigned to two groups. 12 participants were assigned to Group A (Experimental group) and the other 11 participants were assigned to Group B (Control group). Each participant is tested individually.

8.1.2 Variables

8.1.2.1 Independent variables

For this experiment, only one independent variable is used, which is just the condition the participants are assigned to. Specifically, Group A uses interactive visualization, while Group B receives a textual explanation about qualitative attack tree.

8.1.2.2 Dependent variables

There are three dependent variables utilized in this experiment.

First, the understanding is considered. This is measured by a test asked by the researcher to assess comprehension of attack trees.

Second, the engagement is measured. Various methods have been applied to measure this, such as through behavioural observations and questionnaires.

Third, the overall usability of the product is tested. A shortened version of the System Usability Scale (SUS) is applied to verify whether the engagement value is valued. The researcher will also observe and ask the participant questions about whether certain interactions were intuitive. These insights can further improve the design and are part of the qualitive analysis of the experimental group.

8.1.3 Qualitative analysis

It remains to be a challenge to ask the right questions when the researcher wants to know the overall usability and whether his design works or not. Testers are there to validate the design further. Since only the general design and user interactions is tested, as it is the final playtest, and since the researcher has limited time, the FFWWDD questionnaire from Jesse Schell [40] appears to be a good solution for this issue. This questionnaire consists of 6 basic questions which should give the researcher a general idea of what the user thinks about the design. FFWWDD stands for the following:

- "What was the most **frustrating** moment or aspect of this visualization?
- What was the **favourite** moment or aspect of the visualization?
- Was there anything you **wanted** to do, but couldn't?
- If you had a magic **wand** where you could add/change/remove in the visualization, what would you add/change/remove?
- What would you say you were **doing** in the visualization?
- How would you **describe** the visualization to friends and family?" [40]

During the evaluation, not all questions were asked to prevent burden on the users. Usually, the first two questions were asked as they seemed to fit the most to this prototype. Some participants got different combinations to receive a wide scala of answers to the design.

8.1.4 Procedure:

As the experimental design considers a between-subject design, the evaluation of the two groups differs slightly. The only real difference is that the experimental design is also asked about the usability and specific interactions of the prototype. The table below explains the procedures of both groups further.

	Group A (Experimental Group)	**Group B (Control group)**
Introduction	A small explanation is given about what attack trees are to the participant (2+/- sentences).	
Prototype	The participant is asked to interact with the prototype. The controls are briefly explained, and the participant has been given the challenge to reach the diamond by using as few keys as possible. They can explore as much as they or stop early. The researcher observes the interactions and though-process of the participant and makes notes about the completion rate, retention rate and error rate.	-

Provision of information	Upon satisfaction of the participant, the participant is asked to view the debriefing phase of prototype. Here the connection between the original attack tree and current attack tree is explained.	The participant is asked to read the textual explanation about attack trees. They see an example of an attack tree. This is a less extensive and more puzzling version of the debriefing from Group A.
Understanding questions (intro)	Two questions of in each layer of Bloom's taxonomy are asked in each layer: Knowledge, Comprehension and Application. It is an open book test, so the participant is allowed to revise the *prototype* and *information*.	Two questions of in each layer of Bloomy's taxonomy are asked in each layer: Knowledge, Comprehension and Application. It is an open book test, so the participant is allowed to revise the *textual information*.
Understanding questions	Please refer for the understanding questions to table 2 in section of "Measurement instruments"	
Understanding questions (intro)	The participant is also given the question what he or she thinks the leaderboard represents and how the ranking of this leaderboard works. Furthermore, the situation of the questions is shown in both the prototype and the information,	-
Usability	The participant is asked two FFWWDD questions from the to ask about the overall usability [40].	-
Survey	Both groups are presented with a survey. In this survey, they fill in the SUS, questions about engagement such as visual appeal of the attack tree based on a Likert scale, and end with demographics about their technical background.	

Table 2: The experimental design setup.

8.1.5 Measurement instruments

8.1.5.1 Understanding

The questions about the degree of understanding of Qualitative Attack trees are based Bloom's framework called Bloom's Taxonomy. The cognitive domain of Bloom's taxonomy [63-64] is a hierarchical classification of the different levels of knowledge and thinking about a topic. This project only considers the first three domains: Knowledge, Comprehension and Application.

Knowledge is about specific facts, classifications and terminology of, in this case, attack trees.

Comprehension focuses, among other things, on organizing, translating or stating the main ideas of the issue in question.

Application is, as the name of the domain implies, the layer that applies the knowledge in new situations to solve problems.

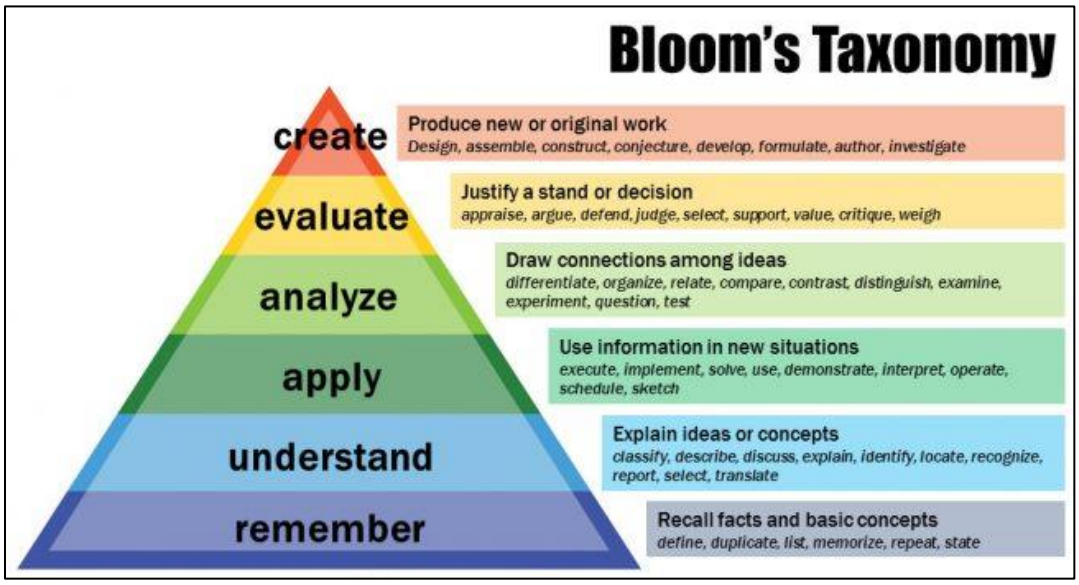


Figure 46: Bloom's taxonomy [63-64]

Each level includes two or three questions. Since each question is an open question, the question must be encoded to a score. For simplicity, three scores can be given to an answer to a question: 0%, 50% and 100%. A wrong answer is classified as a score of 0%, a correct score to 100%, and a partly correct score is 50%. There are various reasons an answer could be partly correct, such as lack of identifying certain element. Below you can find a table of how scoring works when an answer is partly correct.

8.1.5.1.1 Questions

1. **Knowledge**
   - Could you have an idea what this attack tree is trying to show to you?
   - In this example, there is an element named Packet Sniffing. Do you have an idea what "Packet Sniffing" could mean?
2. **Comprehension**

   There are various symbols in this attack tree.
   - Can you point to me the elements called Basic Attack Steps in this example?

   What do you think this symbol means? What is their task?
   - Sequential AND-gate
   - OR-gate
3. **Application**
   - Given the following situation in which three BASs are true, is this an attack set?
   - Can you explain to me what the most important minimal attack set would be if you must include the BAS named "Capture Packets"?

Table 2: The understanding questions asked.

Bloom's layer	*Question*	*Scoring*
1 Knowledge	Attack tree explanation	**100 points:** • Refers to privacy leakage as goal for the attacker. • Shows that Several steps are needed for the attacker to leak privacy. **0 points:** • No reference to privacy leakage as goal.
	Packet Sniffing	**100 points:** • Refers to eavesdropping on a conversation over a network. • Packets = data. **0 points:** • No reference to eavesdropping.
2 Comprehension	BASs	**100 points:** • Refers to all BASs. **50 points:** • Refers to the lowest layer, missing out on the last BAS named "Disclosure of sensitive information". **0 points:** • Refers to an element that is not a BAS or misses selecting multiple BASs.
	SAND gate	**100 points:** • Correct explanation that all conditions under it must be true in a specific sequence for the attack to proceed. • Points to correct element in attack tree. **50 points:** • Makes the mistake of either referring to the wrong element in the attack tree or forgets to mention the sequential part of the SAND gate. **0 points:** • Does not refer to that both conditions must be true.
	OR gate	**100 points:** • Correct explanation that any one of the conditions under it must be for the attack to proceed. • Points to correct element in attack tree. **50 points:** • Referring to the wrong element in the tree **0 points:** • Does not mention that one of the two conditions must be true.

3 Application	BAS situation	**100 points:** • Answers 'no' and provides the answer that the current situation does not reach the root **0 points:** • Answers 'yes' or answers 'no', but provides an incorrect explanation why.
	Creation of attack set	**100 points:** • Identification of the BASs 'Capture Packets', 'Analyse packets' and 'Disclosure of sensitive information' **0 points:** • Anything else is incorrect

Table 3: the scoring of the understanding questions

8.1.5.2 Engagement

Engagement is difficult variable to measure, because it depends on many factors. As mentioned in the background, engagement in this context consists of three phases: the point of engagement, psychological engagement and behavioral engagement. Each phase is briefly measured with different methods. The first phase and second phase are measured with questions on a Likert scale between 1 and 5. This includes questions about the level of the challenge, about the visual appeal and about the novelty. These values can then be compared between the two different groups.

8.1.5.3 Usability (SUS)

The overall usability is measured with the SUS. This is one of the industry's standards in measuring usability. Because usability is not super important and the users are asked many questions, the number of questions halved from 10 questions to 5 questions. Due to this change, an adapted formula has been created to calculate the score:

$$SUS = 5\left((R_1 - 1) + (5 - R_2) + (R_3 - 1) + (5 - R_4) + (R_5 - 1)\right)$$

To clarify this formula, all uneven variables are positive statement about the prototype, while the even variables are negative statements A SUS score above a 68 would be considered above average [57].

8.1.6 Hypothesis

Based on the goal of this project and the experimental design, the following hypotheses have been proposed.

8.1.6.1 Engagement

Engagement will be measured by indicators, such as the level of challenge and visual appeal.

User awareness

Null hypothesis ($H0_{EUA}$): The means score of the user awareness of the attack tree is equal across both groups.
Alternative Hypothesis ($H1_{EUA}$): The means score of the user awareness of the attack tree is higher in the experimental group than in the control group.

Visual appeal

$H0_{EV}$: The means score of the visual appeal of the attack tree is equal across both groups.
$H1_{EV}$: The means score of the visual appeal of the attack tree is higher in the experimental group than in the control group.

Level of challenge

$H0_{EV}$: The means score of the level of challenge of finding minimal attack sets is equal across both groups.
$H1_{EV}$: The means score of level of challenge of finding minimal attack sets differs s across at least one of the groups

Novelty and engagement

$H0_{N}$: The means score of the novelty of the attack tree is equal across both groups.
$H1_{N}$: The means score of the novelty of the attack tree is higher in the experimental group than in the control group.

8.1.6.2 Understanding

$H0_{U}$: The means score of the understanding is equal across both groups.
$H1_{U}$: The means score of the understanding is higher in the experimental group than in the control group.

$H0_{U\_N}$: The means score of the understanding of layer N from Bloom's taxonomy is equal across both groups.
$H1_{U\_N}$: The means score of the understanding of layer N from Bloom's taxonomy in the experimental group than in the control group.

8.2 Results

8.2.1 Engagement

Report

	Prototype?								
	No			Yes			Total		
	Mean	N	Std. Deviation	Mean	N	Std. Deviation	Mean	N	Std. Deviation
The given challenge (finding the minimal attack sets) was optimally challenging	.36	11	1.120	-.58	12	.793	-.13	23	1.058
User Awareness	4.00	11	.775	4.00	12	.603	4.00	23	.674
Visual appeal	2.73	11	.786	4.42	12	.669	3.61	23	1.118
Novelty and engagement	2.45	11	.934	4.33	12	.888	3.43	23	1.308

Table 4: The descriptive statistics of various engagement indicators. The mean score of the give challenge should be as close to zero as possible, while the other indicators should be as high as possible. 5 is the maximum score for these indicators.

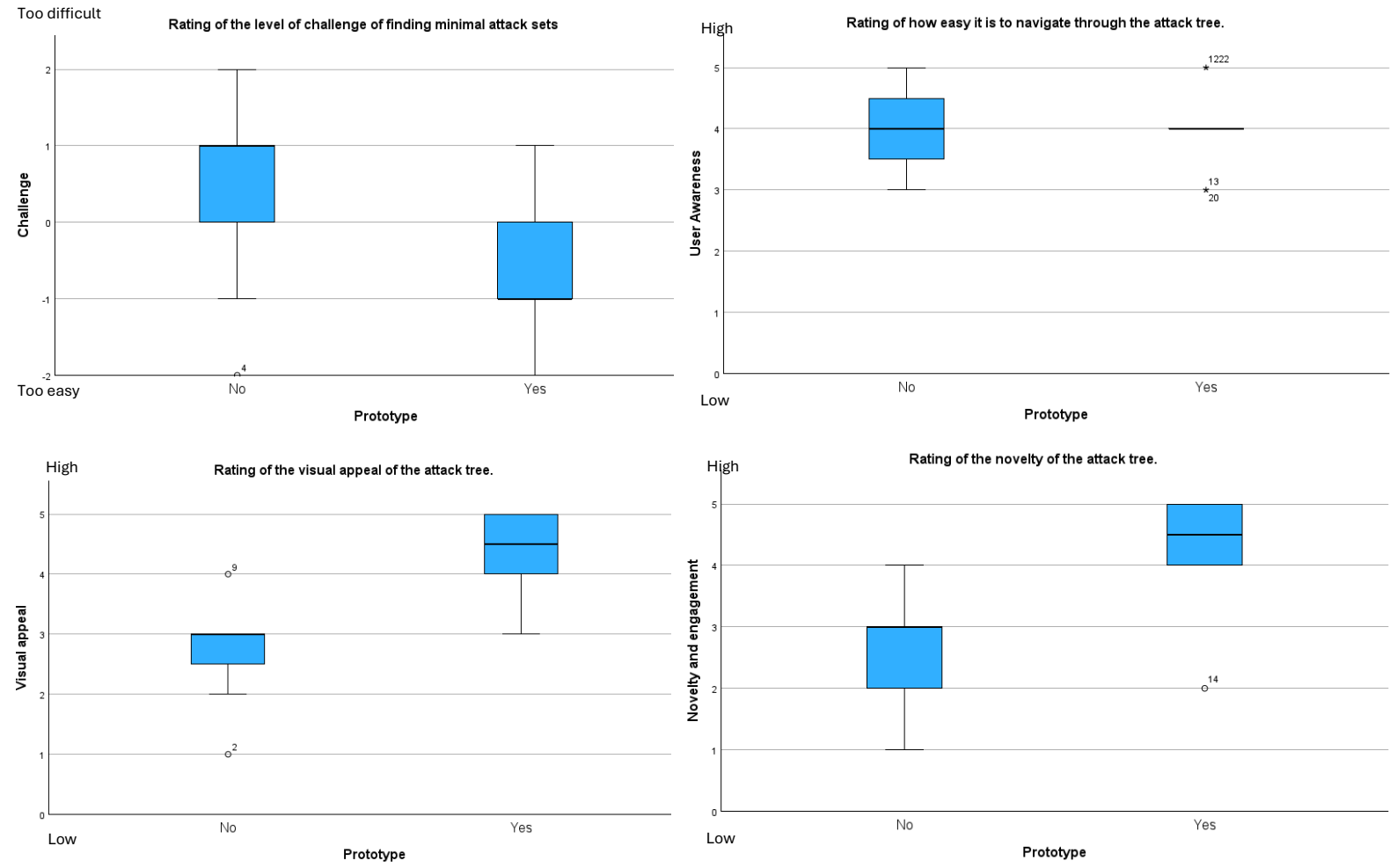


Figure 47: Boxplots of each engagement indicator. The score in top-left boxplot is different than the other boxplots. This boxplot aims for a score as close to zero while the other indicators should be as high as possible.

The descriptives and boxplots show the mean scores for the engagement factors. The score for each factor is based on a Likert scale (1 = strongly disagree and 5 = strongly agree), except for the challenge factor, which uses a Likert scale from -2 (challenge too easy) and 2 (challenge too difficult). Ideally, the challenge factor should have a score as close to zero as possible, indicating that the giving problem is optimally challenging.

The boxplots also reveal potential outliers in the data, which is shown as a dot with the participant number. Since the sample size is relatively small, outliers are kept in the analysis.

From the boxplots and the table, it can be observed that the mean scores for novelty and visual appeal are higher in the experimental group compared to the control group. However, the mean score of user awareness seems roughly similar between the groups.

It is also important to solely observe the mean scores of the experimental group. Their mean scores are 4.0 or above, which seems to suggest that these factors foster the overall engagement. According to the boxplot, the given challenge of finding minimal attack sets, however, were considered easier in the experimental group than in the control group. The mean score from the experimental group is further away from the zero point than the mean score from the control group. Nevertheless, it is preferred in this case to have slightly easier level of challenge than slightly more difficult level, as it can indicate that they understood the material of attack trees better.

Overall, the descriptive statistics show promising results about the level of engagement of the experimental group compared to the control group.

8.2.1.1 Instant feedback

Instant Feedback

	N	Range	Minimum	Maximum	Mean	Std. Deviation
Instant Feedback	12	4.00	1.00	5.00	3.6667	1.15470
Valid N (listwise)	12					

Table 5: The descriptive statistics of the 'feedback' indicator.

In table 5, the mean score for the feedback indicator is shown. This is only measured for the experimental group, because a paper version of an attack tree cannot provide feedback to the user. The mean score is approximately 3.66. This is on the higher end, but it also suggests that some improvements are needed in the design. These problems are outlined in the next section

8.2.1.2 Significant results with non-parametric tests

Tests of Normality

	Prototype	Kolmogorov-Smirnov[a] Statistic	df	Sig.	Shapiro-Wilk Statistic	df	Sig.
Challenge	No	.260	11	.036	.890	11	.141
	Yes	.284	12	.008	.875	12	.077
User Awareness	No	.227	11	.117	.833	11	.025
	Yes	.333	12	<,001	.774	12	.005
Visual appeal	No	.363	11	<,001	.810	11	.013
	Yes	.309	12	.002	.768	12	.004
Novelty and engagement	No	.266	11	.029	.887	11	.127
	Yes	.274	12	.013	.716	12	.001

a. Lilliefors Significance Correction

Table 6: The Shapiro-Wilk test of normality on each indicator. Only the last column is relevant for whether normality can be assumed.

To test if there is a significance difference, an independent sampled t-test is usually conducted. However, the values from the participant were restricted to a Likert scale from 1 to 5, which is ordinal data. Independent samples t-test typically requires the data to be interval or ratio in nature, because the test assumes that the differences between values are meaningful. To double-check, normality can be assessed using the Shapiro-Wilk test for small sample sizes to determine if an independent samples t-test can be conducted. From the last column from Table 6, most significance values are below the $\alpha = 0.05$. Therefore, normality cannot be assumed.

For ordinal data, it is more appropriate to apply the non-parametric Mann-Whitney U test to see if there are significant differences between the groups. From Table 7, the p-values of the *challenge*, *visual appeal* and *novelty* are below 0.05. This means the null hypotheses from these indicators can be rejected.

To sum up, the level of challenge in the interactive visualization is simpler than in the control group. The visual appeal and the novelty of the visualization is higher than in the original attack tree. There is no difference in user awareness regarding navigating through the tree.

Test Statistics[a]

	Challenge	User Awareness	Visual appeal	Novelty and engagement
Mann-Whitney U	31.000	66.000	7.000	10.000
Wilcoxon W	109.000	144.000	73.000	76.000
Z	-2.237	.000	-3.779	-3.536
Asymp. Sig. (2-tailed)	.025	1.000	<,001	<,001
Exact Sig. [2*(1-tailed Sig.)]	.032[b]	1.000[b]	<,001[b]	<,001[b]

a. Grouping Variable: PrototypeNum

b. Not corrected for ties.

Table 7: Mann-Whitney U test on the various indicators. The last row shows the p-values of the test. Only the null hypothesis of User Awareness cannot be rejected.

8.2.2 Understanding

As the understanding scores can also be seen as ordinal data, The Mann-Whitney U test is applied again. The difference in each layer of Bloom's taxonomy is analyses.

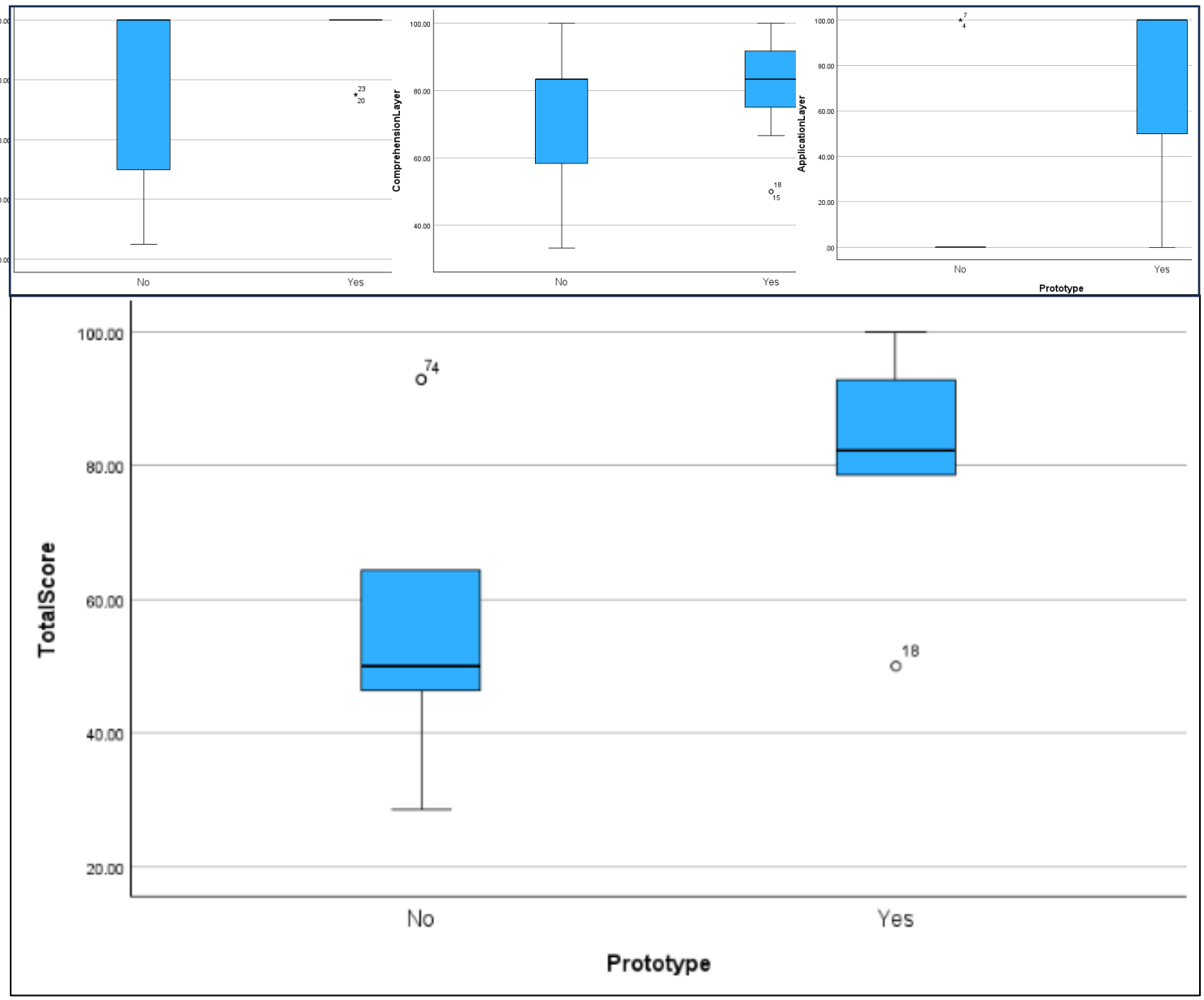


Figure 48: Boxplots of the understanding scores. The three small boxplots are the mean score of each layer. The large boxplots show the means of the total score of the participants between the control group and experimental group.

	TotalScore	KnowledgeLayer	ComprehensionLayer	ApplicationLayer
Mann-Whitney U	22.500	49.000	46.500	21.500
Wilcoxon W	88.500	115.000	112.500	87.500
Z	-2.704	-1.357	-1.278	-2.963
Asymp. Sig. (2-tailed)	.007	.175	.201	.003
Exact Sig. [2*(1-tailed Sig.)]	.006[b]	.316[b]	.235[b]	.004[b]

a. Grouping Variable: PrototypeNum

b. Not corrected for ties.

Table 8: The p-values for the Mann-Whitney U tests.

From table 8, the p-values of the first two understanding layers of Bloom's taxonomy are 0.175 and 0.201 respectively. These values are higher than $\alpha = 0.05$, so $H0_{U_1}$ and $H0_{U_2}$ cannot be rejected. The p-value of the Application layer of Bloom's taxonomy is lower than $\alpha = 0.05$, which means that $H0_{U_3}$ can be rejected. The total score of all the questions from all layers is also lower than 0.05, so $H0_U$ can also be rejected.

Thus, the understanding of the attack sets, and consequently the overall understanding, is significantly higher when participants use the prototype compared to reading the textual explanation. No significant increase is found in the understanding of the terms, knowledge and the various nodes of the attack trees among the participants in the experimental compared to the control group.

Thus, the understanding of the attack sets, and consequently the overall understanding, is significantly higher when participants use the prototype compared to reading the textual explanation. However, there is no significant increase in the understanding of the terms, knowledge, and various nodes of the attack trees among participants in the experimental group compared to the control group.

8.2.3 Overall usability

System Usability Scale

	N	Range	Minimum	Maximum	Mean	Std. Deviation
SUS	12	40.00	60.00	100.00	84.5833	12.69544

Table 9: The System Usability Scale. The mean score is approximately 84.6

From the table, the mean score of the SUS is 84.6. This score is on the higher end, because a score above 68 would be considered an above usability average score [57]. Therefore, the overall usability of the prototype can be considered high.

8.3 Qualitative results and observations

From the FFWWDD questions and observations, it became evident that the visualization had a few challenges regarding the overall usability, despite the high SUS score. There was still some room for improvement in the specific functionality of the prototype. Below is a list of the minor problems and possible solutions identified during the evaluation.

8.3.1 Information boxes

8.3.1.1 Problem: Hovering cue

Problem:

Many participants found it unclear that it was possible to hover over the textboxes to gather more information about specific technical jargon in the attack tree. They noticed it after the developer mentioned that it was possible.

Possible solution: animating outline

Next to the 'I' icon, a visual cue such as an animating outline, can suggest that the textboxes are interactable.

8.3.1.2 Problem: Explanation text

Problem

When the participants did hover over these textboxes, the participants felt that it was too small and hard to read. In addition, some mentioned that the explanations were too long while others indicated that the explanations were too short. This was because they did not have the concentration to read everything.

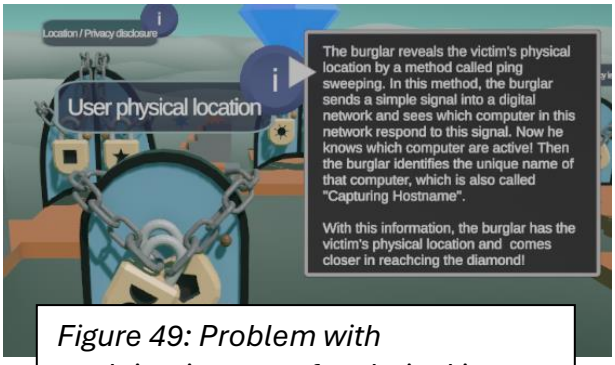


Figure 49: Problem with explaination text of technical jargon

Possible design solution: Read more option

A possible solution is to shorten the text and add an additional button on which the participant can click if they would like to know more about the terminology. By shortening the text, the font size can be made larger so that it is more readable.

8.3.1.3 Problem: No explanation about BAS keys

Problem:

Every textbox is explained except for the BASs. This choice was made to reduce the clutter on the screen. The BASs were explained to their related logic gate. However, it was noticed that not everyone understood the terminology in the BASs on its own.

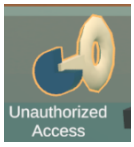


Figure 50: BAS with no information icon

Possible solution:

Add a small 'I'-icon, similarly to the other textboxes with an additional visual cue, such as an animating outline.

8.3.2 Leaderboard

8.3.2.1 Problem: Ranking

Problem:
It was observed that the participants perceived some minimal attack sets as more important even though they had the same number of BASs in their set. This was probably because the minimal attack sets had a lower danger level in the leaderboard. This numbering was done automatically through code, so this was not an explicit design choice to it in this way.

Possible solution: Equal danger level number
This is easily solved by making both items the same colour red and give them the same danger level number.

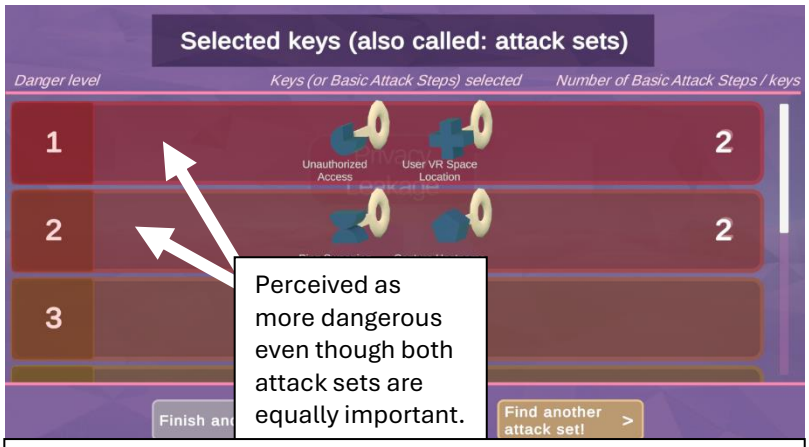


Figure 51: Leaderboard ranking problem

8.3.2.2

8.3.2.3 Problem: No call-up interaction

Some participants noticed that there was no functionality to view the leaderboard while interacting with the attack tree itself. The leaderboard is only shown after completing the task of finding a minimal attack set.

Possible solution
They would have liked to have that functionality, so that they can compare their current path with the path they just have completed.

8.3.3 Basic Attack Steps (selectable keys)

8.3.3.1 Problem: Distinction between BASs and collectible keys

Problem
During the evaluation, it was recognized that some had trouble distinguishing between the BASs in the UI and keys that can be collected on the ground after opening a door. They thought during the evaluation that these also counted as keys for reaching the diamond. Most of them realised during the showcase of the leaderboard that there was a difference in the BASs and collectibles, but it should already be clear by intuition.

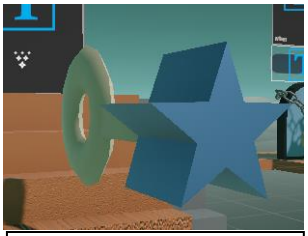


Figure 52: Key collectible looks too like the BASs in the UI.

Possible solution:
The keys can be distinguished by various colours, shapes and sizes. Currently, they look too similar.

8.3.3.2 Problem: Collision boxes UI elements

Problem

It was surprisingly discovered that the collision boxes around the UI elements were too small. This resulted in the users clicking on keys with nothing happening as result.

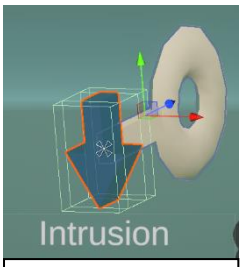


Figure 53: A small collision box in the UI element.

Possible solution:

This is easily solved by making the collision boxes larger.

8.3.3.3 Problem: Feedback

Problem

There were occupations where the participant clicked on a key and it opened a lock on the door, which was not visible in the scene. This confused the player about where they were located in the attack tree.

Possible solution

There are various of ways to solve this. For example, camera movement should follow the key that is selected instead of having a fixed location on the burglar. Additional textual or visual pop-ups or animations could reveal on screen.

8.3.4 Stairs

8.3.4.1 Problem: Stairs have no functionality

Problem

Many participants commented that they did not know why there were stairs in the prototype. It felt like a distracting feature of the visualization.

Possible solution: Removal of collectible keys

Instead of making the keys collectibles, the keys should automatically be given by the prototype when a door is opened. In this way, the burglar cannot collect the keys by walking around the door. With this, the stairs are no longer necessary and can be removed from the scene.

8.3.5 Camera movement

8.3.5.1 Problem: Seeing the next attack tree layer

Through the angle of the camera, it is hard to see the next stages of the attack tree. This was done intentionally to break the information down into smaller chunks, layer by layer. However, because the next layer was still somewhat visible, the participants try to follow the best path even though it was difficult to see.

Possible solution: Top-down view and fog

An option of changing the camera angle could be crated, so that the participant can toggle between third person view and the top-down view. The top-down view would look like the original 2D representation of the attack tree. Another option would be to implement some fog, so that it is not possible to see the next layers in the attack tree, possibly reducing the cognitive load. It would make it harder to decide on which path to choose, but it is not that important as it is an introductory visualization about attack trees. This is one of the solutions that should be playtested to see if it worked.

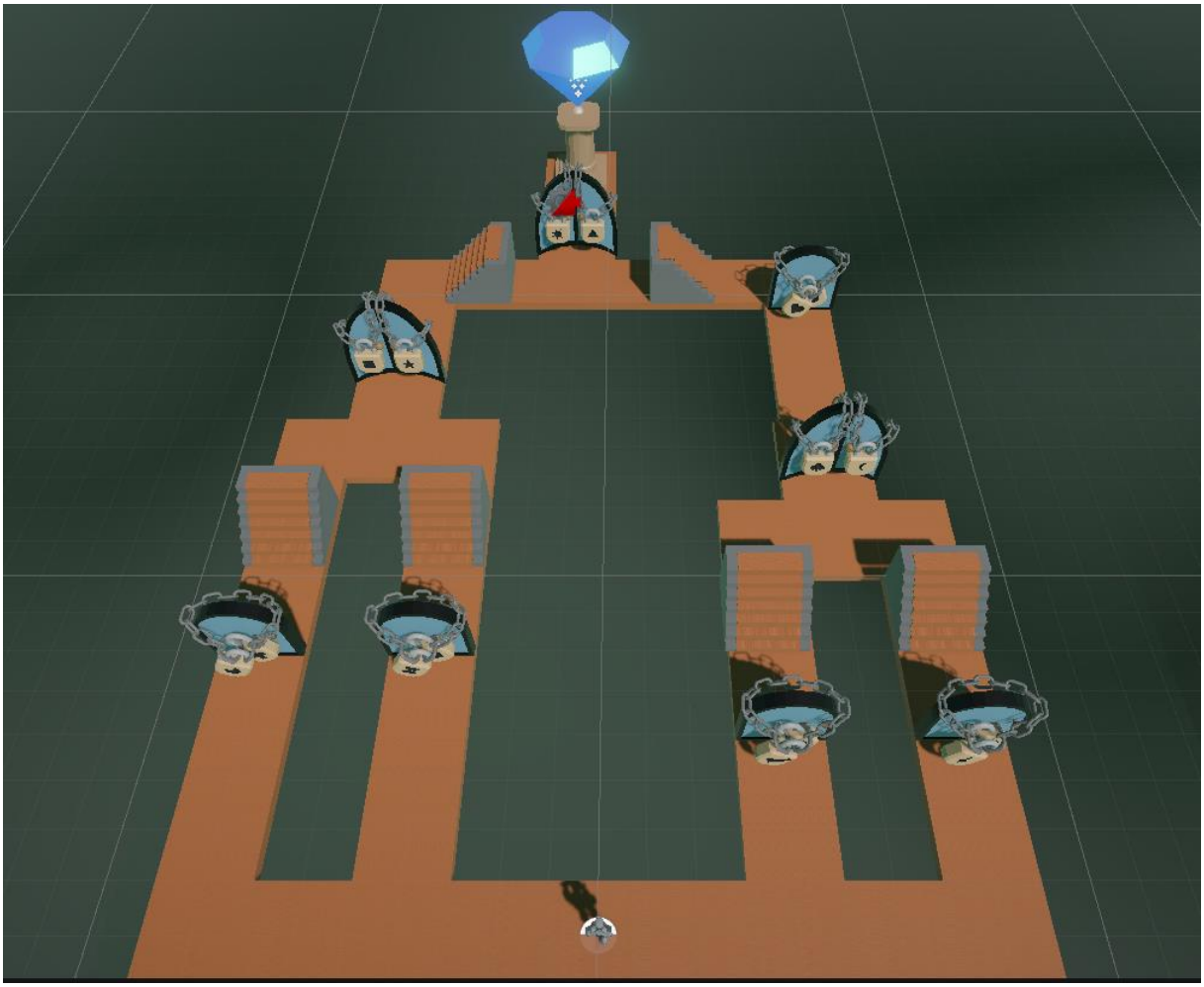


Figure 54: The alternative camera angle for the user.

8.5 Discussion

The results of the evaluation were promising, but there were some limitations that might have affected the interpretation of the results

Firstly, most participants in the user tests were young undergraduate students from the University of Twente with backgrounds in Creative Technology or Industrial design. Some were even familiar with some basic logic gate calculation. This group is not a realistic representation of the population of non-expert stakeholders. This could have potentially led to higher scores in the understanding category.
Besides, it was hardly possible to compare the data based on specific demographics, because these demographics were so similar.
Furthermore, the young audience were likely more advanced in the usage of specific interactions with the computer, such as the movement with arrow keys. It could have been the case that an older audience would have found the system cumbersome to use, because they are not familiar with these interactions and movement buttons. A possible solution for this is the integration of a pathfinding algorithm. With this, the user can click on the walkable floors to move in that direction.

Secondly, the small sample size was a limitation. This may have affected the generalizability of the results about understandability and engagement. Future research will larger sample size is recommended to confirm the findings in the evaluation.

Thirdly, participants from the experimental group slightly outperform the control group on the questions in the comprehension layer, particularly on the question about naming the BASs. Most participants in the control group overlooked the last BAS called “Disclosure of sensitive information”, potentially due to its placement in the attack tree. In contrast, the experimental group had to opportunity to also point to the BASs in either the original model or the interactive visualization. These BASs or selectable keys in the interactive visualization were just all listed into a box, so it was harder to forget one BAS. Although no significant effect was found in this layer, it is worth noting.

Fourthly, engagement has no standardized definition, meaning that it remains difficult to test to what extent the participants were engaged. The engagement was evaluated on several indicators, such as visual appeal and user awareness. Other factors in the prototype could have also influenced the engagement without being aware of it. Future research could look further in detail into the meaning of engagement in the context of explaining attack trees.

Fifthly, during the evaluation, it was also noticed that the participants from the experimental group took slightly more time to answer the questions than those in the control group. This was probably because they attempted to translate the visualization model to the original model. This suggests a potential issue, as the translation from model to model should flow naturally. Time could have been measured, but that also brings some issues with it. Measuring time could be influenced by other factors like concentration and the overall testing duration, impacting participant responses.

Sixthly, the scoring method for the questions could influence the perceived understanding of attack trees. Each question was treated with equal importance, but prioritizing attack sets, or specific nodes over technical jargon questions might provide deeper insights. Alternatively, the scoring could include more detailed criteria, so that partial points can also be given. Additionally, the coding of the scores from the

researcher can also be biased. The researcher determined the grades based on his notes of the participants' answers. Another researcher could have coded the answers differently, potentially leading to different results.

All-in all, there are various biases that could and may have influenced the results. It seems reasonable to see the results more as guidelines than as definitive facts about the visualization.

8.6 Compliance with design requirements

Requirement section	Requirement	Priority	Is the design requirement met?	Explanation
Engagement (point of engagement)	Visual appeal	Must		The visualization adheres to the principles of colour theory and has high score seen in the results. However, there were some individuals who did not enjoy it as much as others. So, there is always room for improvement.
	Interactivity	Could		There are not that many ways to interact with the visualization. The user can click, hover over elements and move with WASD keys. However, there could be method integrated that the user can also click on the floor to move to that direction, which is called pathfinding.
Psychological engagement	Instant feedback	Must		The mean was not particularly high. This was because minor interactions were not flowing smoothly, such as the small collision boxes of the UI elements.
	Level of challenge	Must		From the survey, most users did find the problem relatively close to optimally challenging. From observations, users did not find the most important minimal attack sets in one attempt and were engaged throughout the test. The challenge was considered simple by some participants and that could have various reasons, such as the small size of the attack tree or the nature of a qualitative attack tree.
	User Awareness	Should		The user was aware of the environment, but there were some small flaws mentioned in the qualitative analysis of the evaluation. The mean score is relatively high.
	Minimization of Repetition	Would		There was repetition, such as opening doors, clicking and collecting the same keys. This is important for understanding the material, but can be damaging for the overall engagement
Behavioural engagement	Further exploration	Should		The user can find as many attack trees as they want when clicking the ‘find new attack set’ button in the leaderboard. It is also possible to reset the keys and relock the doors.
Understanding	Analogy of nodes	Must		The animation with locks on doors seems to work well in the application. The participants liked the analogy

	Original symbolic representation	Must		The outline of the doors represents the original symbolic representation. However, in the evaluation, there were still people that missed the fact that the Sequential-AND doors were in order. From the evaluation, the arrow sign in the original attack tree did not help them to identify the sequence factor either. Conversely, implementing another way of ordering, just like numbering the locks may possibly help.
	Progressive layer reveal	Must		As mentioned in the qualitative analysis of the evaluation, there was a problem that the user can still see the next layers, making them curious about what steps the burglar should take next. Fog or top-down view may be a possible solution.
	Explanation of Jargon	Should		There were explanations, but the content was for some too short or too long.
	Miller's law	Should		By grouping BASs based on their related parent by adjusting their position in the UI, it seems easier for users to chunk information, reducing the number of elements in the memory of the users. For larger Attack trees, this solution may not hold anymore, because it may become hard to group all BASs
	Leaderboard	Should		The most important minimal attack sets were shown at the top of the leaderboard. Conversely, some minimal attack sets were perceived as more important even though they had the same number of BASs in their set. This was because the danger level was just a discrete number ascending in magnitude.
Additional requirements	Association keys with lock	Must		Many participants liked the association and found intuitive which key belongs to which lock. There was only occurrence that a participant clicked on a key but opened a wrong door. Therefore, there is still room for improvement by using visual cues. Overall, this requirement is met.
	Communi-cation status node.	Should		The doors and the keys did communicate their status, but this requirement goes together with the instant feedback requirement. Visual cues could help increase the status.

	Flexibility to make own attack tree	Could		The current implementation is hard-coded, meaning that threat modelers cannot implement their own attack tree in the application yet. This could be done in the future.

8.7 Conclusion

The evaluation of the built interactive visualization provides several insights.

The prototype did not provide any significant better understanding regarding the first two layers of Bloom's taxonomy. However, the understanding of attack sets was better in the experimental group than in the control group.

According to the participants, the prototype was engaging, but the participants were not always aware of the environment and feedback given by the system. Through observations and interviews, various design flaws were identified related to this that can be improved.

The overall usability of the prototype was high, meaning that only small changes must be made to refine the prototype.

9 Conclusion

This thesis aimed to explore the impact of interactive and gamification elements on the effectiveness of communication of qualitative attack trees to non-expert stakeholders. It was found that the possible challenges of attack trees include the lack of contextual information, the usage of technical jargon, the lack of highlighting on the risk severity of attack trees, such as minimal attack sets, and the natural structure possibly leading to cognitive overload to users. Additionally, an engagement framework has been formulated to know what design choices must be made to optimally contribute to effective communication of attack trees.

To answer the research question, a gamified interactive visualization is built. Various gamification elements have been implemented into the design, including a leaderboard of minimal attack sets, a challenge for finding the minimal attack set by moving choosing keys for doors, UI elements with instant feedback, a visually appealing environment and a movable character to improve the engagement and understanding for non-expert stakeholders. An analogy of doors and keys for the logic gates and BASs have also been applied for a better understanding of these logic gates.

It was found that the experimental group understood attack sets better than the control group and the prototype was according to the participants engaging. There were a few design flaws identified that can be improved to elevate the engagement even more.

9.1 Future work

In the future, the observations identified in the evaluation can be solved with the provided advice in that section.

9.1.1 Extended user test

Most participants in the user tests were young undergraduate students from the University of Twente with backgrounds in Creative Technology or Industrial design. This is not a realistic representation of the non-expert stakeholders involved in the cybersecurity scene. Therefore, extended user test with a more represented group should be conducted. This includes stakeholders from various age groups, various companies and technical backgrounds.

9.1.2 Tool

Currently, the implementation of the attack tree is hard-coded. This means that the application does not allow users to modify the structure of the attack tree and create their own attack trees. In the future, this should be programmed in such a way that other threat modelers can create their own attack trees in it, just like in AttackTree+. They can then show their design to others and then the design could have an impact.

9.1.3 Complexity

The prototype can also be extended to use more complex elements. Qualitative analysis is useful, but quantitative and cost-damage analysis can reveal even more details about the system and risk severity of it. For example, BASs in the design could have some cost to use it on a lock. The goal of the user can be adapted in a way that tells them they should minimize the cost to reach the root of the attack tree. Possibly similar probability concepts could be integrated into the design. Moreover, the brute-force method of finding minimal attack sets should be changed to a more efficient algorithm. This function may take too much processing power when the designed attack tree contains many nodes.

Through the addition of more complex analyses, the design can automatically implement gamification elements. For instance, an introduction of costs to the keys automatically introduces the element of economy and points. These elements can possibly further increase the engagement of the user. However, this implementation should be done carefully as the opposite effect can also happen when integrated poorly into the design.

10 References

[1] “KNVB betaalt losgeld aan hackers om vertrouwelijke gegevens te beschermen NOS.nl - Nieuws, Sport en Evenementen https://nos.nl/artikel/2490181-knvb-betaalt-losgeld-aan-hackers-om-vertrouwelijke-gegevens-te-beschermen (accessed Feb. 22, 2024).

[2] M. N. Ndlela, “A Stakeholder Approach to Risk Management,” in *Crisis Communication: A Stakeholder Approach*, Switzerland: Palgrave Pivot Cham, 2019, pp. 53–77

[3] C. Girard, J. Ecalle, and A. Magnan, “Serious games as new educational tools: How effective are they? A meta-analysis of recent studies,” *Journal of Computer Assisted Learning*, vol. 29, no. 3, pp. 207–219, Jun. 2012. doi:10.1111/j.1365-2729.2012.00489.x

[4] A. Sundin, K. Andersson, and R. Watt, “Rethinking communication: Integrating storytelling for increased stakeholder engagement in environmental evidence synthesis,” Environmental Evidence, vol. 7, no. 1, Feb. 2018. doi:10.1186/s13750-018-0116-4

[5] J. S. Downs, “Prescriptive scientific narratives for communicating usable science,” Proceedings of the National Academy of Sciences, vol. 111, no. supplement_4, pp. 13627–13633, Sep. 2014. doi:10.1073/pnas.1317502111

[6] S. V. Gentry et al., “Serious gaming and gamification education in Health Professions: Systematic Review,” *Journal of Medical Internet Research*, vol. 21, no. 3, Mar. 2019. doi:10.2196/12994

[7] C. E. Catalano, A. M. Luccini, and M. Mortara, “Guidelines for an effective design of serious games,” International Journal of Serious Games, vol. 1, no. 1, Feb. 2014. doi:10.17083/ijsg.v1i1.8

[8] A. Bobbio, L. Egidi, and R. Terruggia, “A methodology for qualitative/quantitative analysis of weighted attack trees,” *IFAC Proceedings Volumes*, vol. 46, no. 22, pp. 133–138, Sep. 2013. doi:10.3182/20130904-3-uk-4041.00007

[9] S. Mauw and M. Oostdijk, “Foundations of attack trees,” *Information Security and Cryptology - ICISC 2005*, pp. 186–198, 2006. doi:10.1007/11734727_17

[10] M. Lopuhaä-Zwakenberg and M. Stoelinga, “Cost-damage analysis of attack trees,” *2023 53rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, Jun. 2023. doi:10.1109/dsn58367.2023.00057

[11] R. Jhawar, B. Kordy, S. Mauw, S. Radomirovic, and R. Trujillo-Rasua , “Attack Trees with Sequential Conjunction,” in *ICT Systems Security and Privacy Protection: 30th IFIP TC 11 international conference, SEC 2015 Hamburg, Germany, May 26-28, 2015 proceedings*, 2015, pp. 339–353

[12] H. Mantel and C. W. Probst, “On the meaning and purpose of attack trees,” *2019 IEEE 32nd Computer Security Foundations Symposium (CSF)*, Jun. 2019. doi:10.1109/csf.2019.00020

[13] J. Oh, S. Bellur, and S. S. Sundar, “Clicking, assessing, immersing, and sharing: An empirical model of user engagement with interactive media,” *Communication Research*, vol. 45, no. 5, pp. 737–763, Sep. 2015. doi:10.1177/0093650215600493

[14] M. Bond and S. Bedenlier, “Facilitating student engagement through educational technology: Towards a conceptual framework,” *Journal of Interactive Media in Education*, vol. 2019, no. 1, pp. 1–14, 2019. doi:10.5334/jime.528

[15] J. Heller *et al.*, “Tangible service automation: Decomposing the technology-enabled engagement process (TEEP) for augmented reality,” *Journal of Service Research*, vol. 24, no. 1, pp. 84–103, Jun. 2020. doi:10.1177/1094670520933692

[16] H. L. O’Brien and E. G. Toms, “What is user engagement? A conceptual framework for defining user engagement with technology,” *Journal of the American Society for Information Science and Technology*, vol. 59, no. 6, pp. 938–955, Feb. 2008. doi:10.1002/asi.20801

[17] W. Zhang, A. Ghanbaripour, and T. Watanabe, “Exploring the landscape of gamification in Higher Education: A systematic mapping study,” *The Asian Conference on Education 2023: Official Conference Proceedings*, Jan. 2023. doi:10.22492/issn.2186-5892.2024.115

[18] J. Hamari, J. Koivisto, and H. Sarsa, “Does gamification work? -- A literature review of empirical studies on Gamification,” *2014 47th Hawaii International Conference on System Sciences*, Jan. 2014. doi:10.1109/hicss.2014.377

[19] D. Dicheva, C. Dichev, G. Agre, and G. Angelova, “Gamification in Education: A Systematic Mapping Study,” *Journal of Educational Technology & Society*, vol. 18, no. 3, pp. 75–88, 2014.

[20] T. J. Brigham, “An introduction to gamification: Adding game elements for Engagement,” *Medical Reference Services Quarterly*, vol. 34, no. 4, pp. 471–480, Oct. 2015. doi:10.1080/02763869.2015.1082385

[21] M. Donnermann *et al.*, “Social Robots and gamification for technology supported learning: An empirical study on engagement and motivation,” *Computers in Human Behavior*, vol. 121, p. 106792, Aug. 2021. doi:10.1016/j.chb.2021.106792

[22] J. Hamari and V. Eranti, “Framework for designing and evaluating game achievements,” *Proceedings of the 2011 DiGRA international conference*, vol. 10, no. 1224, 2011.

[23] E. Kyewski and N. C. Krämer, “To gamify or not to gamify? an experimental field study of the influence of badges on motivation, activity, and performance in an online learning course,” *Computers & Education*, vol. 118, pp. 25–37, Mar. 2018. doi:10.1016/j.compedu.2017.11.006

[24] M. Sailer, J. U. Hense, S. K. Mayr, and H. Mandl, “How gamification motivates: An experimental study of the effects of specific game design elements on psychological need satisfaction,” *Computers in Human Behavior*, vol. 69, pp. 371–380, Apr. 2017. doi:10.1016/j.chb.2016.12.033

[25] G. Richter, L. C. Wood, T. Reiners, D. R. Raban, and S. Rafaeli , “Studying Gamification: The Effect of Rewards and Incentives on Motivation,” in *Gamification in education and business*, Bentley, Australia: Springer International Publishing, 2015, pp. 21–46

[26] M. Malone *et al.*, “To Gamify or Not?: On Leaderboard Effects, Student Engagement and Learning Outcomes in a Cybersecurity Intervention,” *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*, Mar. 2021. doi:10.1145/3408877.3432544

[27] R. S. Alsawaier, “The effect of gamification on motivation and engagement,” *The International Journal of Information and Learning Technology*, vol. 35, no. 1, pp. 56–79, Jan. 2018. doi:10.1108/ijilt-02-2017-0009

[28] R. N. Prasetyono and R. Cipta, “Development of flipbook using web learning to improve logical thinking ability in Logic Gate,” *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 1, pp. 342–348, 2020. doi:10.14569/ijacsa.2020.0110143

[29] H. S. Lallie, K. Debattista, and J. Bal, “An empirical evaluation of the effectiveness of attack graphs and fault trees in cyber-attack perception,” *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 5, pp. 1110–1122, May 2018. doi:10.1109/tifs.2017.2771238

[30] Orgill, M.K. & Bodner, G. *“*What research tells us about using analogies to teach chemistry”. *Chemistry Education: Research and Practice,* 5(*1), 12-32, 2004*

[31] Northeastern Center for Advancing Teaching and Learning Through Research, “Using analogies to help others learn,” *Center for Advancing Teaching and Learning Through Research,* https://learning.northeastern.edu/using-analogies-to-help-others-learn/ (accessed Apr. 12, 2024).

[32] C. Ellermann *et al.*, “Identifying content to improve risk assessment communications within the risk profile: Literature reviews and focus groups with expert and non-expert stakeholders,” *PLOS ONE*, vol. 17, no. 4, Apr. 2022. doi:10.1371/journal.pone.0266800

[33] T. Wu *et al.*, “What risk? I don’t understand. an empirical study on users’ understanding of the terms used in security texts,” *Proceedings of the 15th ACM Asia Conference on Computer and Communications Security*, Oct. 2020. doi:10.1145/3320269.3384761

[34] D. Bawden and L. Robinson, “The dark side of information: Overload, anxiety and other paradoxes and pathologies,” *Journal of Information Science*, vol. 35, no. 2, pp. 180–191, Nov. 2008. doi:10.1177/0165551508095781

[35] E. Bolisani, E. Scarso, and A. Padova, “Cognitive overload in Organizational Knowledge Management: Case Study Research,” *Knowledge and Process Management*, vol. 25, no. 4, pp. 223–231, Jun. 2018. doi:10.1002/kpm.1579

[36] R. Pennington and B. Tuttle, “The effects of information overload on software project risk assessment*,” *Decision Sciences*, vol. 38, no. 3, pp. 489–526, Aug. 2007. doi:10.1111/j.1540-5915.2007.00167.x

[37] A. M. Toda *et al.*, “Analysing gamification elements in educational environments using an existing gamification taxonomy,” *Smart Learning Environments*, vol. 6, no. 1, Dec. 2019. doi:10.1186/s40561-019-0106-1

[38] A. Mader and W. Eggink, “A DESIGN PROCESS FOR CREATIVE TECHNOLOGY”, *IN 16th International Conference on Engineering and Product Design, E&PDE 2014 ,* 4 & 5 september, 2014, pp. 568-573, ISBN: 978-1-904670-56-8

[39] D. Rosenberg, “What are user flows?,” The Interaction Design Foundation, https://www.interaction-design.org/literature/topics/user-flows (accessed Apr. 21, 2024).

[40] J. Schell, *The Art of Game Design: A Book of Lenses*. Boca Raton, FL: CRC Press, Taylor & Francis Group, 2020.

[41] F. Arnold, D. Guck, R. Kumar, and M. Stoelinga, “Sequential and parallel attack tree modelling,” *Lecture Notes in Computer Science*, pp. 291–299, 2015. doi:10.1007/978-3-319-24249-1_25

[42] J. Hobson, “Using pulleys and weights to explain binary logic gates,” Hackaday, https://hackaday.com/2014/05/30/using-pulleys-and-weights-to-explain-binary-logic-gates/ (accessed May 2024).

[43] A. Gorischek, “Pulley logic gates on Vimeo,” Vimeo | Pulley Logic Gates, 2014, https://vimeo.com/93042377 (accessed May 2024).

[44] Whizzbizz, “Mechanical Logic Gates,” Youtube, 20 Dec. 2022, https://www.youtube.com/watch?v=bzYcMvdYKyU.

[45] keishiroueki2158, “Mechanical logic gate AND” Youtube, 31 Aug. 2023, https://www.youtube.com/shorts/L3_I3n2BaFE.

[46] J. Rasmussen et al., “Risk Visualization and Simulation ,” Jamie Rasmussen : Risk visualization and simulation, IBM OpenPages and D3, http://www.jamierasmussen.com/portfolio/risk.html.

[47] Isograph, “AttackTree+,” *AttackTree*. https://www.isograph.com/software/attacktree/ (accessed 2024)

[48] C. E. Budde and M. Stoelinga, “Efficient algorithms for quantitative attack tree analysis,” *2021 IEEE 34th Computer Security Foundations Symposium (CSF)*, Jun. 2021. doi:10.1109/csf51468.2021.00041

[49] E. Nkemchor, “Designing the perfect user flow diagram,” Articles on everything UX: Research, Testing & Design, https://blog.uxtweak.com/user-flow-diagram/ (accessed 2024).

[50] Apple, “Healthcare - Apple Watch,” Apple, https://www.apple.com/healthcare/apple-watch/ (accessed 2024).

[51] G. A. Miller, “The magical number seven, plus or minus two: Some limits on our capacity for processing information,” *Essential Sources in the Scientific Study of Consciousness*, pp. 357–372, Apr. 2003. doi:10.7551/mitpress/2834.003.0029

[52] S. Klemmer, “Lecture 3.1 Storyboard, Paper Prototypes, and Mockups,” *Interaction Design Specilalization*, Youtube, uploaded by IxD Online: UCSD & Coursera, 2 Feb. 2016, https://www.youtube.com/watch?v=12OpiFIF26Y.

[53] This Person Does Not Exist. https://thispersondoesnotexist.com/, accessed 2024.

[54] D. Saffer, “Designing Microinteractions,” in *Microinteractions: Designing with Details*, Sebastopol: O’Reilly Media Inc., 2013, pp. 14–19

[55] S. Kumar, “An introduction to interaction flows,” Medium, https://medium.com/apploitte/an-introduction-to-interaction-flows-8332677d976f (accessed 2024).

[56] R. Papush, “Ultimate Clouds with Shader Graph in Unity, Made Easy,” Youtube, uploaded by Roman Papush, 21 Jul. 2019, https://www.youtube.com/watch?v=Y7r5n5TsX_E.

[57] J. Sauro, “Measuring usability with the system usability scale (SUS),” MeasuringU, https://measuringu.com/sus/ (accessed 2024).

[58] T. Stafford, “The psychology of tetris,” BBC News, https://www.bbc.com/future/article/20121022-the-psychology-of-tetris (accessed 2024).

[59] Partyverhuur-Verkoop, “Oudhollands Jenga Stapeltoren: Partyverhuur-Verkoop,” verkoop.nl, https://www.partyverhuur-verkoop.nl/product/8088/oudhollands-jenga--stapeltoren (accessed 2024).

[60] J. Moossdorff, “Barricade Hout (26X26CM) - Kopen Bij spellenrijk.nl,” Spellenrijk.nl - Online Spellenwinkel, https://www.spellenrijk.nl/artikel/26903/barricade-hout-26x26cm.html (accessed 2024).

[61] Speelgoedpaleis, “AquaPlay 1660 - AquaPlay ’N go,” Het Speelgoedpaleis, https://www.hetspeelgoedpaleis.com/AquaPlay-1660-AquaPlay-n-Go (accessed 2024).

[62] Spelenopzeilen, “Home,” Spelenopzeilen Slangen en Ladders, https://www.spelenopzeilen.nl/spelzeilen/ladders-en-slangen-india-levend-ladders-en-slangen%E2%80%83 (accessed 2024).

[63] P. Armstrong. “Bloom’s Taxonomy.” Vanderbilt University Center for Teaching. 2010, https://cft.vanderbilt.edu/guides-sub-pages/blooms-taxonomy/. (accessed 2024).

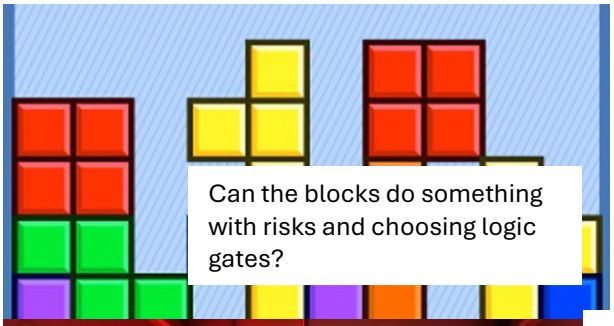
[64] L.W. Anderson, D.R. Krathwohl, “A Taxonomy for Teaching, Learning, and Assessment”, Longman: New York, NY, USA, 2001; pp. 1–333.

[65] Marlon Fraile, Margaret Ford, Olga Gadyatskaya, Rajesh Kumar, Mariëlle Stoelinga, Rolando Trujillo-Rasua: Using Attack-Defense Trees to Analyze Threats and Countermeasures in an ATM: A Case Study. PoEM 2016: 326-334.

11 Appendix

11.1 A1: Ideation

Moodboard Ideation

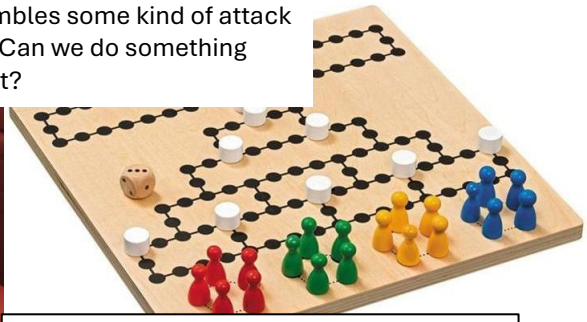


Can the blocks do something with risks and choosing logic gates?

Risks with Jenga? Probability of a node can be defined by the length and width of a wooden piece.

Risk of cutting a wire? Very interactive 2-player game. Goal is to prevent the bomb from exploding (Keep Talking and Nobody Explodes)

The game Barricade resembles some kind of attack tree. Can we do something with it?

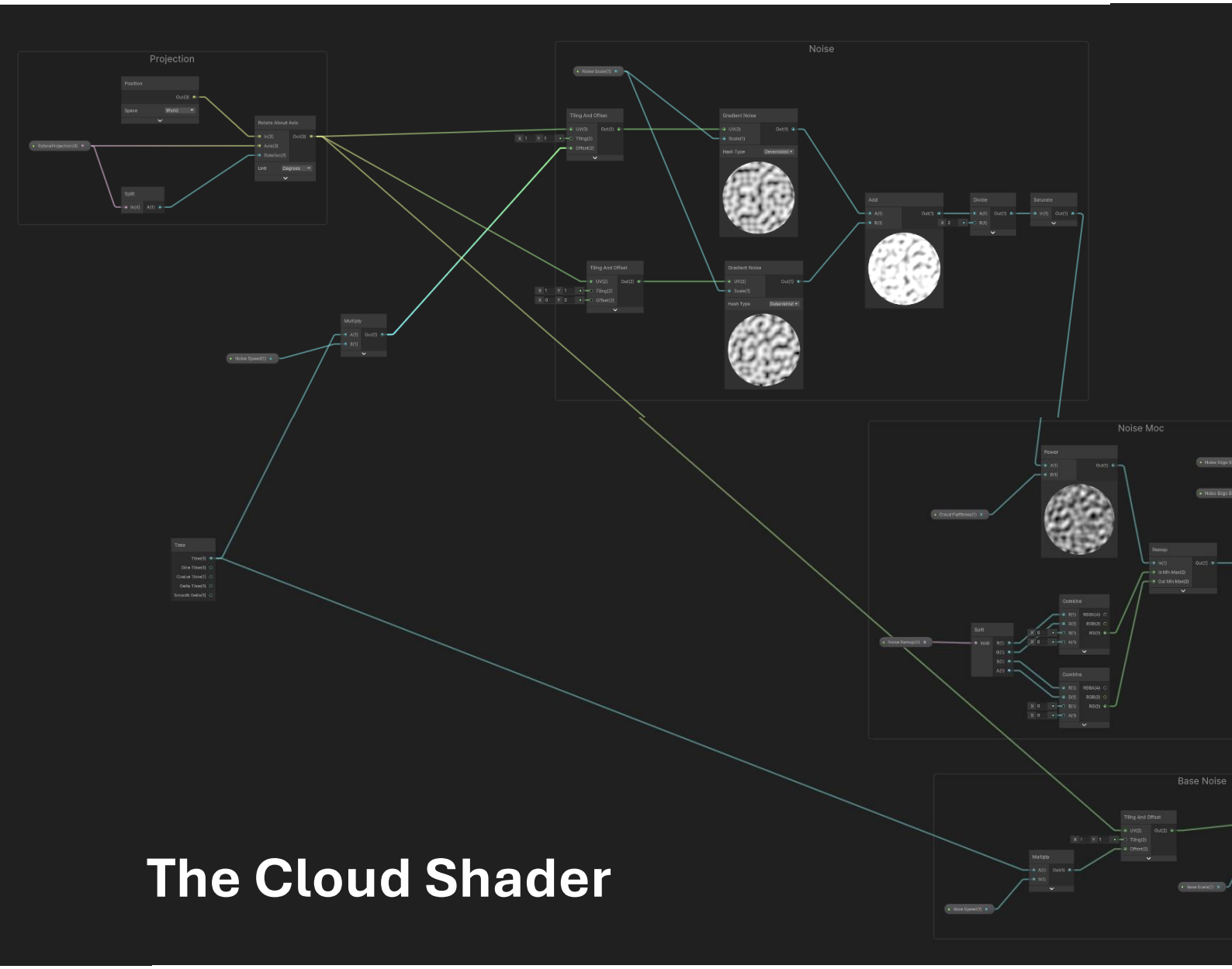


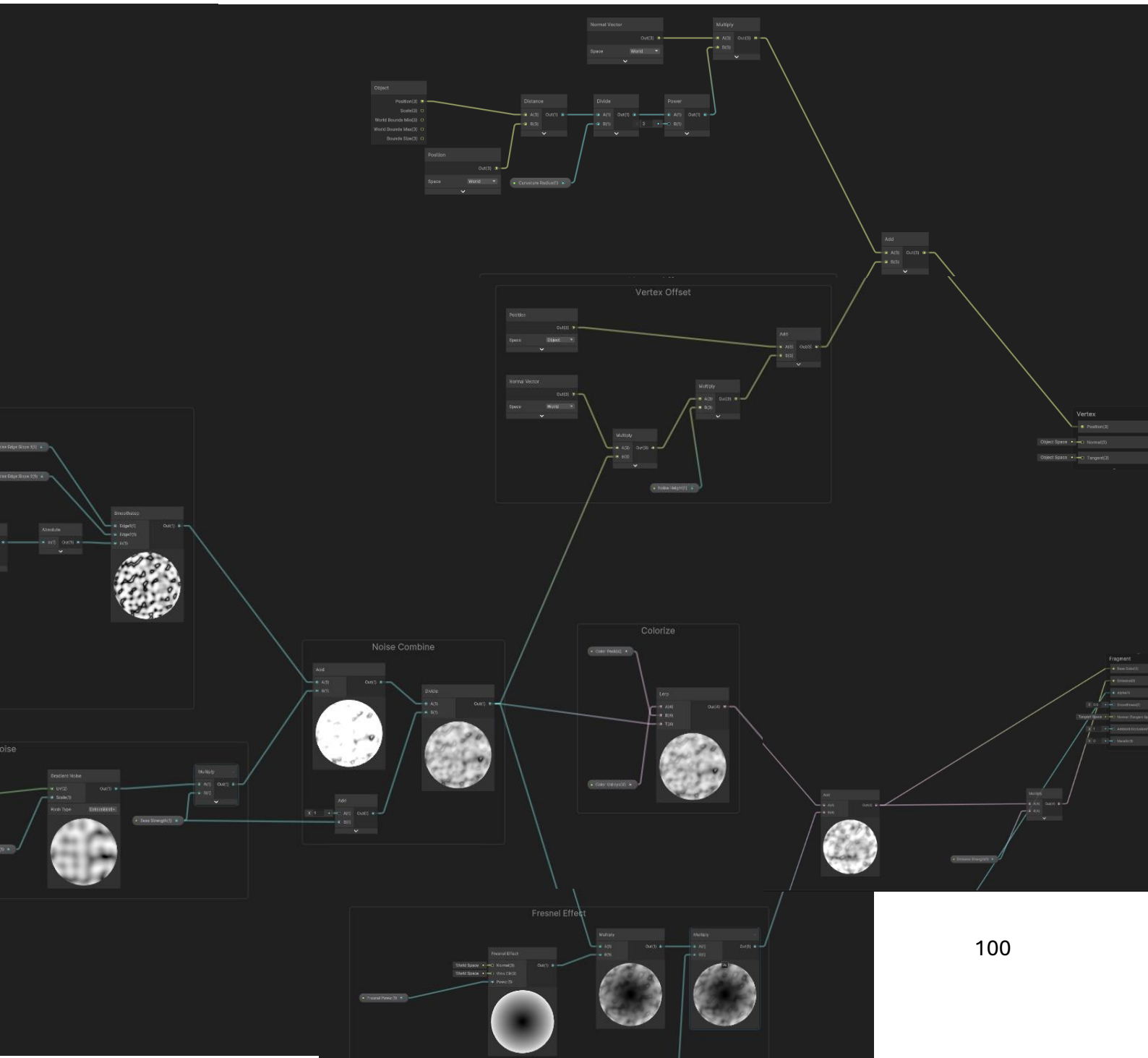
Slangen and Ladders can also resemble a decision tree? Can slangen and ladders make it more engaging?

Aquaplay is similar to a marble track. Does the diverging and converging path resembles logic gates?

[58 – 62]

11.2A2: Realization




Vertex Offset
Noise Combine
Colorize
Fresnel Effect
Vertex

The code of the entire visualization can be found in the UTwente Archive.

Or contact c.s.tenholder@student.utwente.nl for more information.

11.3 A3: Evaluation

Hey [participant],
Thank you for being here and testing before. My graduation is about attack trees and I'm not going what it means, but I made something related to it. I've made an interactive visualization where [the participant] play as a burglar, trying to rob this diamond. You can use the arrow keys and mouse to move and click. The goal is for you to reach the diamond by selecting as little keys shown at the bottom as possible. I'm going to ask you later questions about my design and how much you understood it. So feel free to explore the design as much as you want, or quit early. It is totally up to you.

Task 1:
Explore the attack tree

Task 2
Now please, there is a textual explanation about attack trees. Please read this and explore it. This is an image of the same attack tree but shown in a different and static way.

Questions

1. **Knowledge**
   - Could you have an idea what this attack tree is trying to show to you?
   - In this example, there is an element named Packet Sniffing. Do you have an idea what "Packet Sniffing" could mean?
2. **Comprehension**
   There are various symbols in this attack tree.
   - Can you point to me the elements called Basic Attack Steps in this example?

   What do you think this symbol means? What is their task?
   - Sequential AND-gate
   - OR-gate
3. **Application**
   What do you think this leaderboard represent? How do you think this ranking work.
   - Given the following situation in which three BASs are true, is this an attack set?
   - Can you explain to me what the most important minimal attack set would be if you must include the BAS named "Capture Packets"?
   -

Engagement feedback (FFWWD) (examples)
What specific interactions or feedback were frustrating and your favourite about the tool?
What did you want to do, but couldn't?
If you had a magic wand and you could remove, change or add something, what would you do?
Any other tips for the design regarding the usability of the tool?

Workflow Researcher for the user

A brief explanation about Attack trees

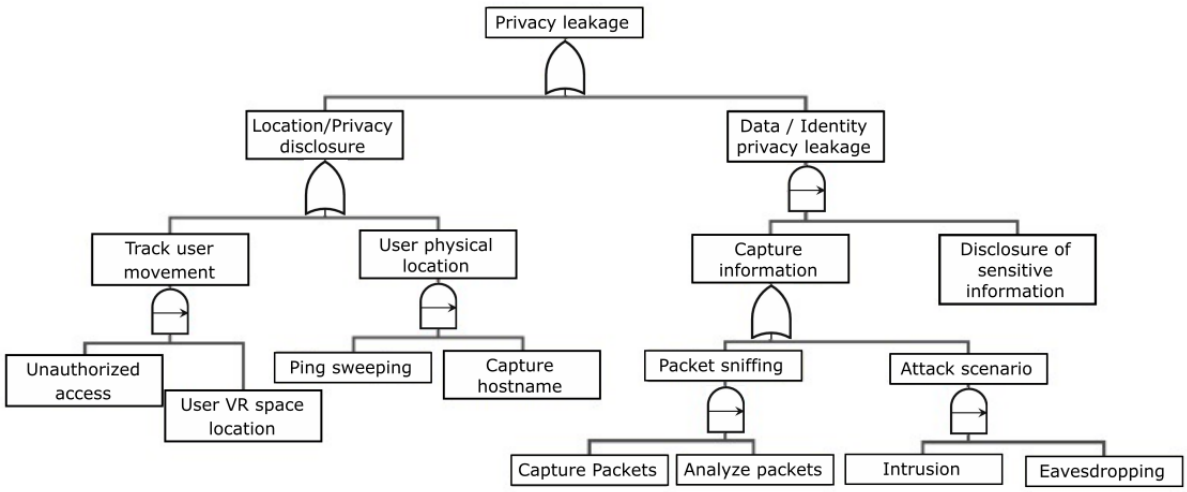
An attack tree is a mathematical model that shows how a target might be attacked. Attack trees can be made for different contexts. It can be displayed as a multi-levelled diagram, a tree, consisting of a root, child nodes and leaves. An attack is considered successful when a complete path can be made from one or multiple leaves, thought he child nodes' conditions, up to the root. The leaves of the tree are also called Basic Attack Steps (BAS).

The condition of these nodes can vary, and represent specific actions or conditions in the attack tree

- ** SAND gate**: This gate means that all conditions under it must be true in a specific sequence for the attack to proceed.
- **OR gate**: This gate means that any one of the conditions under it must be true for the attack to proceed.

The figure below shows a qualitative attack tree. There are many forms of Attack trees, such as quantitative and cost-damage attack trees. These attack trees will not be addressed in this explanation.

Identification of a list of minimal combinations of BASs that lead to the final objective can reveal where a security system is the most vulnerable. These are also called minimal attack sets. The order or importance of a minimal attack set is defined by the number of BASs that must be true to reach the final goal



Consent Form for *Visualization and gamification of attack trees*

YOU WILL BE GIVEN A COPY OF THIS INFORMED CONSENT FORM

Please tick the appropriate boxes	**Yes**	**No**
Taking part in the study		
I have read and understood the study information dated [18/06/2024], or it has been readto me. I have been able to ask questions about the study and my questions have been answered to my satisfaction.	☒	○
I consent voluntarily to be a participant in this study and understand that I can refuse to answer questions and I can withdraw from the study at any time, without having to give a reason.	☒	○
I understand that taking part in the study involves an interview, and the researcher will take notes. After finalization, the notes will be discarded.	☒	○
Use of the information in the study		
I understand that information I provide will be used for the design of an interactive visualization explaining attack trees together with the bachelor thesis related to this project.	☒	○
I understand that personal information collected about me that can identify me, such as [e.g. my name or where I live], will not be shared beyond the study team.	☒	○

Signature

Name of participant	Signature	Date

I have accurately read out the information sheet to the potential participant and, to the bestof my ability, ensured that the participant understands to what they are freely consenting.

Casper ten Holder

Researcher name	Signature	Date

Study contact details for further information:
Casper ten Holder, c.s.tenholder@student.utwente.nl

Contact Information for Questions about Your Rights as a Research Participant

If you have questions about your rights as a research participant, or wish to obtain information, ask questions, or discuss any concerns about this study with someone other thanthe researcher(s), please contact the Secretary of the Ethics Committee Information & Computer Science: ethicscommittee-CIS@utwente.nl

AT: understanding and engagement

* Verplichte vraag

1. I thought the attack tree was easy to understand *

   Markeer slechts één ovaal.

	1	2	3	4	5	
Stro	◯	◯	◯	◯	◯	Strongly agree

2. I found the attack tree unnecessarily complex. *

   Markeer slechts één ovaal.

	1	2	3	4	5	
Stro	◯	◯	◯	◯	◯	Strongly agree

3. I felt very confident understanding the system. *

   Markeer slechts één ovaal.

	1	2	3	4	5	
Stro	◯	◯	◯	◯	◯	Strongly agree

4. I found that the various nodes in this attack tree were well integrated. *

   Markeer slechts één ovaal.

	1	2	3	4	5	
Stro	◯	◯	◯	◯	◯	Strongly agree

Engagement

In this section, two question will be asked about the different factors playing a role in the engagement

5. The given challenge (finding the minimal attack sets) was optimally challenging *

Markeer slechts één ovaal.

- ◯ -2: Too simple
- ◯ -1: Simple
- ◯ 0: Optimally challenging
- ◯ 1: Difficult
- ◯ 2: Too difficult

6. Did you find it easy to navigate through the attack tree? *

Markeer slechts één ovaal.

	1	2	3	4	5	
Stro	◯	◯	◯	◯	◯	Strongly agree

7. The design was visually appealing *

Markeer slechts één ovaal.

	1	2	3	4	5	
Stro	◯	◯	◯	◯	◯	Strongly agree

8. The design is novel and engaging. *

Markeer slechts één ovaal.

	1	2	3	4	5	
Stro	◯	◯	◯	◯	◯	Strongly agree

Demographics

9. How old are you?

_____

10. What is your gender? *

Markeer slechts één ovaal.

- ◯ Male
- ◯ Female
- ◯ Prefer not to say
- ◯ Anders: _____

11. Do you have a technical background? *

Markeer slechts één ovaal.

- ◯ Yes
- ◯ No

12. Have you heard of attack trees before this test? *

Markeer slechts één ovaal.

- ◯ Yes
- ◯ No

Example of notes during an evaluation of a participant.