VR Stealth Action Video Game in Unity

Peter Do
Stanford University
peterhdo@stanford.edu

Brennan Shacklett
Stanford University
bps@stanford.edu

Abstract

We created a first person VR stealth game, along with an environment designed to create a captivating experience while allowing us to demonstrate our stealth mechanics. Our goal was to create as enjoyable an experience as possible both in terms of comfort and immersion. To accomplish this, we experimented with the tradeoffs inherent in several types of locomotion and environment designs, and our results provide insight into designing a coherent VR experience.

1. Introduction and Motivation

Stealth games have traditionally been a somewhat niche genre, but recent high profile successes such as Metal Gear Solid V show that the genre provides a unique opportunity for gameplay that meshes tightly with an intricate environment. Additionally, there is a very basic set of core mechanics necessary to make a stealth game: a cover system, and a type of enemy AI to evade. In turn, these mechanics lead to a game loop between two states: sneaking and fleeing. While the mechanics are relatively simple at first glance, there can be infinite variations and possibilities within them, so we decided this would be a good place to experiment with VR game design. Additionally, we had previous experience with developing within the stealth genre as we developed a third person total stealth game as our final project in CS248. We thought that the challenge of taking a stealth game to virtual reality would be interesting and allow us to explore different angles that our previous game would not.

Starting with the lessons learned from our third person stealth game, we decided to work from the ground up on rethinking all the mechanics for VR. In particular, we wanted to be sure we had a game that was clearly designed for VR, rather than simply being a port of a prior game, which would have resulted in a much less interesting and a worse performing experience overall. One of the key motivations for doing another stealth game was the possibilities offered by the position and orientation tracking capabilities of modern head mounted displays when combined with the traditional stealth concept of cover. In 3rd person stealth games for conventional screens, the cover system is often very complicated and hard to implement well, due to issues such as detecting when a player character is near cover and then using inverse kinematics to properly position the player. This typically means that even well implemented cover systems occasionally have moments where they become clunky or frustrating. Conversely, the positional tracking capabilities of our HMD allow for more natural cover based mechanics with a player's crouching movements in the real world directly translating to ducking behind cover in the game. This leads to an overall more satisfying cover mechanic that gives players a lot of control over their own movement while behind cover.

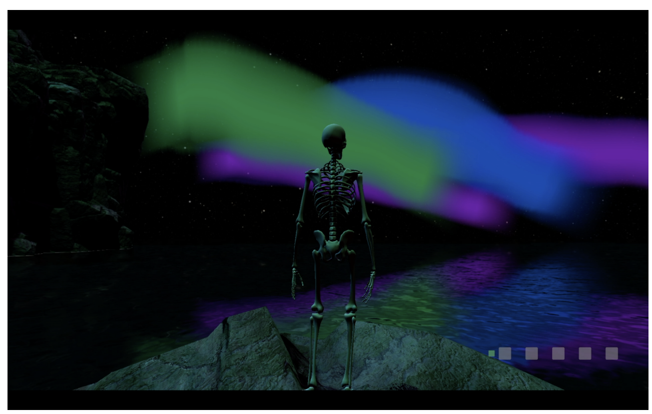

Figure 1. A view of the beginning of the environment from the previous game. This is the type of atmosphere we planned to recreate.

We had to deal with some limitations of VR in doing so, however. We developed the environment around movement, which we knew had to be revamped from the original game. We wanted our environment to be similar in aesthetic to the previous game, which would allow for a more immersive experience. The virtual environment, however, had to take into account the different perspective shift and VR limitations. In addition, we had to keep in mind performance limitations as we wanted the Oculus Rift to run at a reason-

Figure 2. The UI of the previous game included ”traces” of the player character to make it easier to determine where states were stored.

able frame rate. Movement presented its own fair share of issues that we will describe in detail in later sections, but the level of freedom that existed in the third person stealth game we developed would not be compatible in VR without more sensors than we had and a lot more room. We decided that a large environment would provide more interesting gameplay opportunities and implemented a warping mechanism to allow players to traverse the level, primarily from cover to cover. Traditional stealth works well on its own, but we decided to incorporate a time mechanic that allowed players to save state at their current location to return to later. This works identically to how it would work in a third person game, but players have to be more cognizant of their position in space as first person does not allow for as much of the level to be in view.

2. Related Work

2.1. World Design

There has been much work in the field of virtual reality, particularly in the realm of video games. We found that papers discussing VR motion sickness, locomotion techniques, VR level design techniques, and VR sound design were insightful and assisted in the design of our game. As a large environment with plenty of cover served as the main backdrop for the game, we decided that our world design had to be different from our prior design, based on the work described in ”Designing Intentional Impossible Spaces in Virtual Reality Narratives: A Case Study.” In their paper, the authors explain that due to space limitations in the real world, ”impossible spaces can be utilized to enhance the aesthetics of [...] an interactive narrative” [2]. However, instead of interlacing different rooms together we decided that players needed to be able to traverse a more open level more quickly and so we introduced the warp mechanic into our stealth game.

2.2. Traversal

One of the major issues we came across when we began working on the game was figuring out how movement would work in VR. There are several existing solutions that allow player movement through positional tracking but those generally restrict the player to what the sensor can view. Mahdi Nabiyouni and Doug A. Bowman emphasize in their paper ”A Taxonomy for Designing Walking-based Locomotion Techniques for Virtual Reality” that ”Designers have yet to find a fully general and effective solution to solve the problem of walking in large or unlimited virtual environments” [4]. The paper also emphasized the fact that natural locomotion allows for a more immersive experience, decreases motion sickness, and contributes to a better user experience [4]. Despite this, devices that allow for natural traversal are not practical at this time, which limited our options to what we could work with: a traditional controller and positional head tracking.

2.3. Sound Design

Sound design was also an important aspect of the VR experience and we discovered that players could feel far more immersed in an environment with accurate spatial sound [5]. The enemies from our original game were audible and player movement was also emphasized to give feedback on the state of the world. We wanted to maintain the level of awareness provided by the audio cues given in the game. A study has demonstrated that using spatial-sound techniques in a virtual environment contributes to ”higher levels of presence experienced, and does so with a medium to large effect” [5]. In their study, a virtual scene of a forest was shown with or without sound to participants and the questionnaires indicated that they felt more immersed with spatial-sound present. With this in mind, we worked to incorporate spatial sound to bring players more fully into the scene.

2.4. Motion Sickness

One issue we wanted to avoid when developing the stealth game was motion sickness. There has been a large amount of work done in the space of virtual reality to combat this issue and we found that with a headset that has a good field of view and resolution like the Oculus Rift, motion sickness can be avoided with good game design. A study has shown that field of view restrictors can be helpful to users for adjusting to a virtual reality environment, but that many subjects didn’t notice the difference [1]. However, we felt that having a field of view that attempts to be as immersive as possible adds the experience and didn’t generally lead to motion sickness. Regardless, we used the default field of view in the Oculus as it seemed satisfactory. Much of the Oculus and Unity documentation indicated that movement with a gamepad in virtual reality would lead to motion sickness so we avoided that. However, in a stealth game, it is important to be able to move to look over and around cover to make a plan to tackle the situation at hand. A study titled ”Lean into it: Exploring Leaning-Based Motion Cueing Interfaces for Virtual Reality Movement” indicated that having leaning movements done with a gamepad would not lead to any motion sickness [3]. Other results within the paper showed that usability, longevity of use, enjoyment, and presence were all not affected by the type of controller. Because of this, we decided that keeping leaning attached to the positional tracking in the Oculus Rift would work well.

3. Methods

3.1. Movement

As mentioned earlier, the first key design decision we made was how to allow movement through the VR world. Initially (and in our proposal), we had planned to use simple first person movement with a traditional gamepad, but we quickly discovered several issues with this approach. In our 3rd person stealth game, we allowed the player to use the left stick to control the direction and speed at which the player was walking, meaning pushing farther forward on the stick caused the player to go from a slow careful walk to running. This level of control was necessary for a stealth game to avoid being detected by enemies, but when translated to VR we found that the acceleration inherent in this movement system resulted in almost immediate nausea due to the lack of correspondence between the behavior of the camera and the stationary player in the VR headset. We also experimented with being able to toggle on and off constant forward movement, but found that this was very frustrating to use correctly and still resulted in some nausea. Ultimately we settled with a warp point mechanism, where the screen very briefly fades to black and then fades the player back in at their new location. This was inspired by the Oculus and Unity documentation and recommendations, which primarily recommended either fade to black or ”blinking” for warp locomotion in order to avoid any perceived movement by the player.

While many VR games which use warp based locomotion allow the player to warp nearly anywhere in the level, we decided to restrict warping to predetermined warp points which we place throughout the level. This rationale behind this will be discussed more in the final section, but this decision allowed us to very easily optimize the player's experience in the environment, because there are only a finite number of positions the player can experience the level from.

While this simplified movement greatly, it left the problem of how to select which warp point to travel to. Although the option of allowing the player to select from the warp points currently in their field of view using the d-pad or one of the controller sticks was briefly a possibility, we quickly found that this system introduced numerous edge cases, because there is a disconnect between the relatively simple input of pressing left or right on the controller and which of possibly numerous points with different depth the player actually wanted to switch focus to. To simplify the design and make it more consistent, we implemented a system where we consider all the warp points within a relatively small cone in front of the player, and select the closest point. We experimented with the size of the cone, and ultimately settled on only considering points 20 degrees off of the direction the player is currently looking. This ensures that points to the edge of the screen are ignored, while also ensuring the system isn't too particular about needing to look directly at a point in order to warp there, which would likely be a frustrating and not particularly intuitive experience. We also limited the points considered by their distance: if a point is too close, it is likely the point the player was already located at, so it should be ignored, and if the point is too far, we do not want to allow the player to warp across the entire level. In addition, a short cool down was introduced after a warp to prevent players from quickly warping from spot to spot without properly planning their actions.

The visual elements of the warp points underwent several iterations. Originally the selected warp point was simply marked by a blue spotlight on the ground, which was clear and easy to see on the white plane we used for testing. Unfortunately, once our scene became more complicated, with different ground textures and even grass, the spotlight quickly broke down and became vague and hard to see. To combat this issue we added a glowing blue orb slightly above the spot where the player would warp to. This made the current warp point choice easy to immediately recognize, and worked over any type or color of terrain.

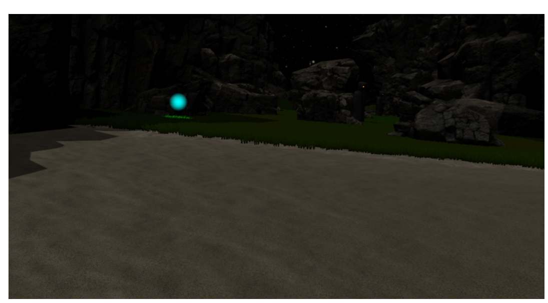

Figure 3. Warp points are highlighted with a point light and spot light to give players a point of reference as to where they will transport in a given direction.

3.2. Time Mechanic

Closely related to the visuals of the warp points is our time rewind system. In the original 3rd person stealth game, the player had a save slot system, where they could save up to 5 previous positions, and then rewind to previous positions or fast forward to future positions. This was indicated with a minimalist UI that tracked how many save slots had been used, and a ”trace” (ghost like copy) of the 3rd person character was left at the rewind point. Since overlay style UI simply does not work naturally in VR, our minimalist interface could not be used, and our system of traces simply did not hold up in VR. Therefore we used a similar glowing orb to the warp points, with a green orb marking the fast forward position, and a red orb marking the rewind position. The rewind / fast forward orbs are also elevated considerably higher than the warp point indicators, so the player can easily see the general position of their rewind position from over cover.

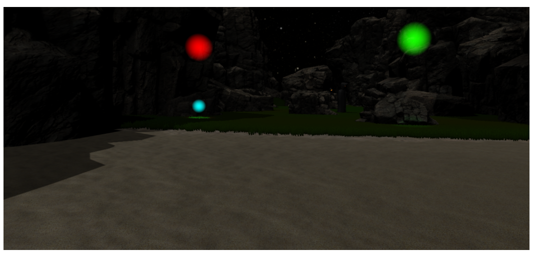

Figure 4. The time mechanic includes a red point light to indicate a backwards time travel point and a green light to indicate a forwards time travel point.

3.3. Sound Design

Another major design element is the game’s sound design. While in the 3rd person stealth game there was an ambient sound of water and wind to heighten the immersion of the player, in VR we found that this ambient noise was simply too distracting, because it only detracted from the ability of the players to tell where the enemies were located. In order to ensure that an observant player could tell the position of nearby enemies from sound alone (in case the player was hiding behind cover for example), every enemy has an ambient sound, which grows as the player is closer to them using Unity’s built in spatial sound systems. Additionally the enemies give a very distinctive screech when they see the player, in order to ensure that the player is immediately alerted to the fact that they need to change their position as soon as possible.

3.4. Enemy AI

The enemy AI is designed to be somewhat unpredictable, but also to allow for as fair of a user experience as possible. The enemy AI will only detect the player if they can directly see the player’s current head position. While this is obviously inaccurate since a stray arm or shoulder could potentially give away a player’s position, without a player model for the user to reference, we decided that any other approach would be unfair.

Another key element of the AI is its interaction with player sounds. Warping and rewinding both make noise, which enemies will hear, and then investigate. To add realism, sufficiently large cover will block sound from travelling to enemies, so this mostly prevents the player from warping directly next to an enemy with its back turned. Additionally, we combined the Oculus’ positional tracking capabilities with player sound, so every movement by the player in the real world translates to some sound enemies can detect in the virtual world. This is implemented by measuring the change in position of the player every frame, and creating a larger sound for larger movements. The intent of this feature is to increase immersion, as the player must remain physically still when near an enemy in order to avoid being detected. It was fairly evident early on that the enemies could not exist in the same manner they did in the third person stealth game as they moved too quickly and aggressively for players to react properly in VR. Some tweaking of values such as the distance they could view players from was required to get the game into a playable state.

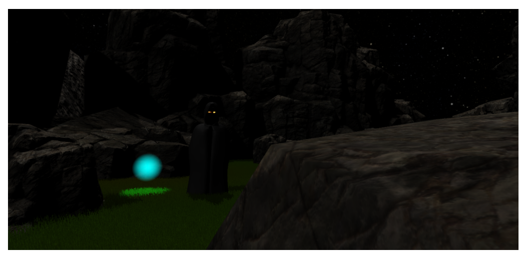

Figure 5. The cloaked enemies emit sound and have glowing eyes so players can discern their location. Players need to use cover and the warp and time mechanics to avoid them.

3.5. Player Character

One issue unique to VR was the virtual presence of the player character within the environment. In our 3rd person stealth game the player was a skeleton, and originally we planned to have the skeleton’s body in the first person VR perspective, so you could look down within the virtual environment and tell you were a skeleton. Unfortunately matching the skeleton’s model to the positional tracking of the camera turned out to be essentially impossible given Unity’s current tools. While Unity includes support for using inverse kinematics (IK) to position a model, it does not provide IK support for setting the position of the head. This meant we had to manually control the amount the skeleton model crouched, based on the positional tracking information and our estimate of the user's height. This wound up working decently, but the real problem was rotating the skeleton model correctly. The naive approach is to simply rotate the skeleton to match the current looking direction, but this is clearly not how people look around in the real world. Instead people rotate their head at the neck and then only rotate their body when necessary to look behind them. Unfortunately given the Oculus with only one positional tracker there is simply no way to track these complexities in Unity, and we found the disconnect between your movement's and the movements of the skeleton were very disconcerting and broke immersion. Therefore we simply removed the player character model.

3.6. Environment

The environment served as the playground for the stealth action and we spent a significant amount of time deciding the trade offs between a more immersive environment and an environment that was more about the gameplay. Having a simple environment would allow for players to easily scan the environment and understand the layout, but might not have as much visual variety. In addition, creating assets for virtual reality can be difficult compared to a regular third person game as players will be able to see the environment at different angles and are sometimes able to see textures mere inches from their face in virtual space. To try and make the game as playable as possible, we kept the environment fairly simple while also giving players a lot of options to go through the level as they wished. The style of the environment is fairly similar to that of the previous game as it takes place at night with a shoreline and the ocean not far off. The traversible land has some vegetation and large rocks on either side of a valley, indicating that the player should progress down that direction. In addition, warp points provide plenty of opportunities for players to lure enemies and use the time mechanics to maneuver around the environment to reach the end. We decided to include the vegetation as ground textures in VR can easily look unconvincing if the ground is flat and seams between the ground and objects are easily visible.

We had to take several items into account when creating the environment including lighting, scale, placement of enemies, and opportunities. As the scene took place at night, we wanted it to be dark enough to fit the time of day but bright enough to make the scene visible and playable. The fact that warp points and time states were highlighted with lights helped significantly in drawing the player's attention where it needed to be however and the enemies' glowing eyes made them stand out as well. Scale was somewhat of an issue as many of the assets we had originally didn't work in their current state - cover would be too tall and rocks too large to maneuver around, or the grass wouldn't look right at certain heights. We had to adjust the scale of all of these objects to make a consistent environment that was immersive. We also had to take into account where enemies would be placed and that they needed to be in areas that would encourage the player to use mechanics or look for optimal routes to avoid losing their progress. This ties into the opportunities created by the environment as the abundance of cover locations allows players to approach from a variety of angles.

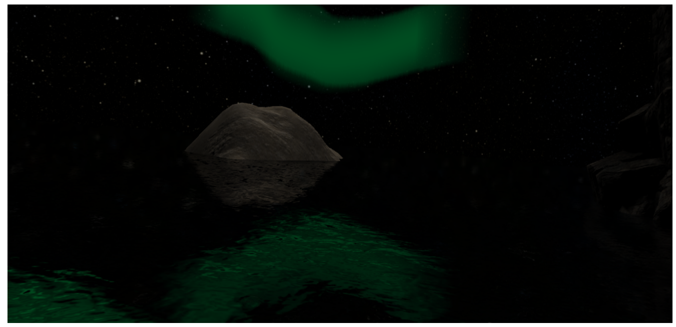

Figure 6. We tried to keep the ambiance of the environment dark and somewhat foreboding.

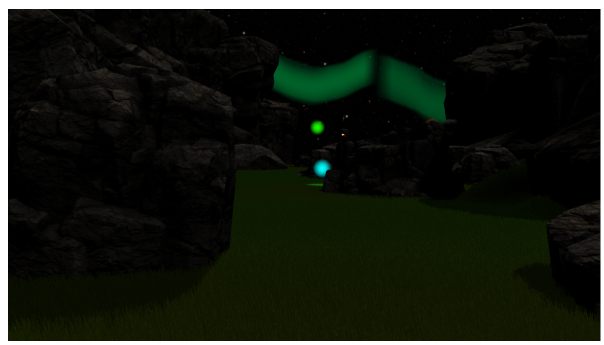

Figure 7. There are multiple paths one can take to complete the level.

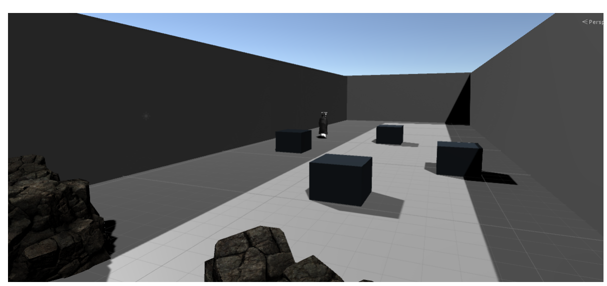

Figure 8. An early experimental level where we developed the mechanics. A cleaner level made for a tighter experience but did not provide the theme we wanted to convey.

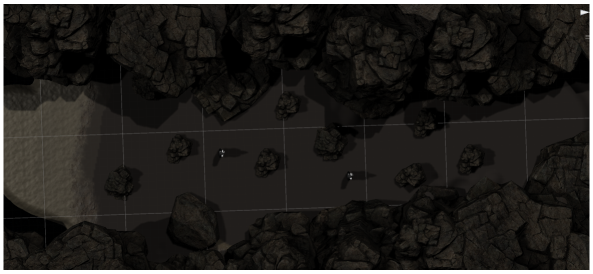

Figure 9. An aerial view of the layout of the level.

4. Evaluation

4.1. Comfort

We found our warp point locomotion system to be essentially devoid of motion sickness, which makes the game feel very comfortable to play overall. Unfortunately, there are some inherent limitations to the hardware we had which can make the experience uncomfortable in certain edge cases. Most significantly, since we only have a single Oculus sensor, rotating around in a full 360 degree turn is not always completely smooth. In particular, once the user's head is between the sensor and the HMD, the Oculus tends to skip around significantly. This usually results in a noticeable lurch that can feel uncomfortable when the user looks behind them. To combat this we made the level relatively linear moving forward, in order to minimize the amount that the player has to look backwards. Unfortunately looking backwards to check on an enemy that was just evaded is a fairly common action, so this happens relatively often.

A similarly uncomfortable issue arises when the player attempts to move their head inside an object within the environment. Unity's default response to this is to zoom the scene outwards rather than let a player go within an object within the world. This is very confusing to experience at first, as it causes the illusion of the virtual environment to rapidly deteriorate. One alternative would to simply allow the camera to clip through the object, or to fade out objects which are immediately in front of the camera, but both these solutions would likely also be quite disconcerting at first. Additionally we deciding that fading out objects would be hard to balance within the stealth game. At the simplest level, this would allow players to easily look across the level without needing to stand up and take the risk of peeking over cover, which is one of the core interactive gameplay components. Additionally, the enemy AI currently would not be able to detect a player who's head was inside a piece of cover within the environment. Therefore while this issue is somewhat immersion breaking, we do not see an easy to way to work around it without fundamentally altering the gameplay.

4.2. Performance

In general we avoided using any particularly demanding assets, and the scene was able to run smoothly within the headset without any noticeable juttering or framerate drops. One of the most potentially expensive assets in the scene is the water behind the player, because the water must rerender the scene twice to handle reflections and refractions. We were able to optimize this by simply only reflecting and refracting the very small number of objects near the water, which made the performance impact of the water negligible. We also made sure to keep the sight lines within the environment relatively small so occlusion culling could save us from rendering expensive parts of the scene (such as the auroras or the water) at all times.

5. Discussion

5.1. Visuals

Designing the environment was ultimately a difficult compromise between variety within the level and how crisp the level appeared within the HMD. Ultimately our lack of resources for creating high resolution assets for VR meant that their were some visual compromises we would rather not have made. Rather than limiting ourselves to simple assets where errors would be less apparent, but which would create a fairly dull environment, we decided to use the assets we could find and accept that not everything would look amazing up close or from every angle. This allowed us to create a scene with a lot of interesting sight lines and a consistent theme. Realistic-looking virtual reality environments can be difficult to create with the current state of HMDs and the power of consumer graphics cards. If future work were to be completed on this project, we would likely consider a stylistic change to allow for a more aesthetically pleasing environment and avoid the issues of creating high-fidelity virtual reality assets.

5.2. Movement

With access to another Oculus positional sensor we would have liked to explore how a roomscale VR experience would work with our stealth mechanics. This would have allowed more freedom of movement and reduced the need for the point warping mechanic. For a first step we would explore allowing a player to walk all the way around a piece of cover rather than needing to incrementally warp around it. The greater freedom of movement would likely also give the player the temptation to attempt to physically run away from or evade enemies, and hitting the boundaries of the roomscale environment and needing to warp to go farther could potentially be frustrating when in a stressful situation within the virtual world. One solution to this would be to make sure all cover was sufficiently far apart that the only way to move from cover to cover is warping, to avoid any confusion between when walking around the room versus warping to a point is the correct action for the player. Alternatively, the distance between pieces of cover could potentially be decreased so player could fully walk around a roomscale environment, evade a couple of enemies and then warp to the next room, but this approach could potentially be overly restrictive depending on the maximum size of a roomscale environment. One downside of switching to roomscale VR is that we would significantly restrict the number of people who could play the game, since many people do not have the required two sensors or even enough room to play in a roomscale environment. The current design of our game allows players with any amount of space to play the game, as you only need to be able to crouch, stand up and lean in order to fully experience it.

Regardless, the single sensor warp point system we have still causes some complications. One subtle element of using warp points for movement was deciding where exactly the warp points should go. In anything but a completely regularly shaped environment, which would be potentially boring from a visual and gameplay perspective, there is not a clear correspondence between the piece of cover and where exactly the player wants to warp to. This means we needed to choose between having a lot of warp points so the player could go essentially anywhere, and only a few warp points so the player gets a more controlled experience.

Although fewer warp points could potentially feel restrictive, we found that the player's experience could become confusing when too many possible warp points existed at any one time. If too many warp points sat in a relatively straight line together from a certain position, this situation could also make the user interface frustrating if the player couldn't warp to the point they wanted to go towards. This tended to be a problem in the middle of our environment, where the player could fully turn around and see 5 or more warp points, without a clear indication of which were better or worse than others. Ultimately we mostly addressed this through streamlining level design, rather than modifying the core UI or mechanics.

6. Conclusion

The overall experience provided by our demo creates a convincing case for the potential of a stealth game in VR. Although VR currently has several limitations that make a fully traversable open level unrealistic, we found a suitable set of mechanics that allow the player to interact in the environment in a meaningful way, while still maintaining an comfortable, and most importantly, fun experience.

References

[1] Ajoy S Fernandes and Steven K Feiner. "Combating vr sickness through subtle dynamic field-of-view modification". In: *3D User Interfaces (3DUI), 2016 IEEE Symposium on*. IEEE. 2016, pp. 201–210.

[2] Joshua A Fisher et al. "Designing intentional impossible spaces in virtual reality narratives: A case study". In: *Virtual Reality (VR), 2017 IEEE*. IEEE. 2017, pp. 379–380.

[3] Alexandra Kitson et al. "Lean into it: Exploring leaning-based motion cueing interfaces for virtual reality movement". In: *Virtual Reality (VR), 2017 IEEE*. IEEE. 2017, pp. 215–216.

[4] Mahdi Nabiyouni and Doug A Bowman. "A Taxonomy for Designing Walking-based Locomotion Techniques for Virtual Reality". In: *Proceedings of the 2016 ACM Companion on Interactive Surfaces and Spaces*. ACM. 2016, pp. 115–121.

[5] Sandra Poeschl, Konstantin Wall, and Nicola Doering. "Integration of spatial sound in immersive virtual environments an experimental study on effects of spatial sound on presence". In: *Virtual Reality (VR), 2013 IEEE*. IEEE. 2013, pp. 129–130.