

Evaluating Player Experience in Stealth Games: Dynamic Guard Patrol Behavior Study

Wael Al Enezi, Clark Verbrugge

School of Computer Science, McGill University, Montréal, Canada
wael.alenezi@mail.mcgill.ca, clump@cs.mcgill.ca

Abstract

In stealth games, guard patrol behavior constitutes one of the primary challenges players encounter. While most stealth games employ hard-coded guard behaviors, the same approach is not feasible for procedurally generated environments. Previous research has introduced various dynamic guard patrol behaviors; however, there needs to be more play-testing to quantitatively measure their impact on players.

This research paper presents a user study to evaluate players' experiences in terms of enjoyment and difficulty when playing against several dynamic patrol behaviors in a stealth game prototype. The study aimed to determine whether players could differentiate between different guard behaviors and assess their impact on player experience.

We found that players were generally capable of distinguishing between the various dynamic guard patrol behaviors in terms of difficulty and enjoyment when competing against them. The study sheds light on the nuances of player perception and experience with different guard behaviors, providing valuable insights for game developers seeking to create engaging and challenging stealth gameplay.

Introduction

Guard patrol behavior is one of the essential design aspects in stealth games. These behaviors are typically manually crafted by game designers to provide a suitable level of challenge for players. However, this approach relies on static game levels, making it impractical for procedurally generated environments. Previous research has proposed dynamic guard patrol behaviors that can adapt to procedural game levels. Despite these advancements, the impact of such behaviors on player experience has not been thoroughly evaluated through play-testing.

Play-testing plays a crucial role in game design as it allows designers to assess how various game components influence players and make iterative improvements to enhance the overall player experience. This research paper aims to present a user study to evaluate a set of dynamically generated patrol behaviors in a prototype of a stealth game. We collect gameplay data and survey responses from human players to gauge their experiences.

This study holds significant relevance as it provides insights into the enjoyment and difficulty levels associated with these dynamically generated patrol behaviors in stealth scenarios. Additionally, the study examines the traits that players assign to these behaviors and how these traits impact their perceived enjoyment and difficulty levels. The contributions of this chapter are as follows:

- We re-design a prototype to allow human players to play against a game prototype. The prototype is a stealth game¹. A Unity-built top-down simulation allows for testing and analysis of various stealth behaviors.
- We conducted a user study to evaluate the effect of different behaviors on players' experience. The user study was based on playing with the prototype and survey data. We consider three forms of patrol behavior; two methods are more heuristically complex, and the third is a simple baseline method.

Related Work

Emergent Guard Patrol & Search Behavior

Multiple studies have delved into the potential for generating emergent patrol and search behaviors. In the context of game development, the commercial game *Third Eye Crime* presented a captivating search behavior that drew inspiration from occupancy maps, a technique commonly employed in robotics for exploration and mapping purposes (Moravec 1989). It incorporated occupancy maps to improve agents' knowledge representation to create a more realistic pursuit and search behavior in stealth scenarios (Isla 2013, 2005).

Additionally, occupancy maps have been employed to generate dynamic and exploratory behavior for NPCs in various games. In the turn-based rogue-like game *NetHack*, for instance, occupancy maps were utilized to enable an NPC to exhibit exploratory behavior (Campbell and Verbrugge 2019). Similarly, in open real-time strategy (RTS) games where fog of war plays a role, another grid-based approach known as *potential fields* was utilized to direct an NPC's navigation within the game space (Hagelback and Johansson 2008). This technique relied on generating potential fields that influenced the NPC's movements, considering factors

¹The source code can be found at <https://github.com/wralenezi/Stealth-Simulator.git>

such as visibility and unexplored regions, thereby facilitating strategic exploration.

Work by Xu, Tremblay, and Verbrugge aimed at guard movement and patrol patterns, using a generated road map in the game level and adding a grammar-based route and behavior construction 2014. Another study used the same presentation to create a multi-agent search behavior for an adversary (AI Enezi and Verbrugge 2021).

Recent work introduced a method for an observer to explore and patrol a game level by decomposing the undiscovered space and allocating a numerical value to each convex unit in the undiscovered area. Then the observer prioritizes the undiscovered convex regions (AI Enezi and Verbrugge 2020).

Researchers use the percentage of the area covered to evaluate exploration performance in robotics (Paull et al. 2018). Regarding patrol, previous work in robotics studied patrol performance and used the uniform coverage of the area as a heuristic, assuming that the more evenly robots surveyed the area, the more effective their patrol behavior is (Jana, Vachhani, and Sinha 2022). We follow the same method to assess the dynamic patrol performance quantitatively.

Player Perception of NPC Behavior

Research has indicated that players generally find NPCs that exhibit more human-like behavior to be more enjoyable to play against (Soni and Hingston 2008). Also, various studies have sought to directly address player enjoyment, exploring the possibility of enhancing games by establishing mathematical models that dictate game progression. The objective is to define a game’s progress model in order to adjust player enjoyment through the manipulation of “acceleration” within the game’s progress model, particularly in sports and board games (Sutiono, Purwarianti, and Iida 2014; Sutiono et al. 2015; Iida, Takeshita, and Yoshimura 2003). Moreover, player enjoyment has been examined in the context of NPC AI across multiple game genres. This includes turn-based strategy games (Wetzel and Anderson 2017), board games (Iida, Takeshita, and Yoshimura 2003), and first-person shooters (Soni and Hingston 2008; Hingston 2009). These studies have aimed to better understand and improve player satisfaction by refining the behavior and interactions of NPCs within these specific game types.

Conversely, a survey conducted among professionals in the video game industry revealed concerns regarding players’ ability to comprehend and perceive the increased complexity of NPCs. This phenomenon, often referred to as “the black hole of AI”, underscores the need to understand the limitations associated with it (Johansson, Eladhari, and Verhagen 2012). As a result, it is crucial to evaluate the boundaries of this phenomenon. Several studies have delved into various aspects of NPC behavior in contemporary commercial games, which have the potential to impact a player’s enjoyment. Lankoski & Björk have proposed a range of pertinent design patterns that contribute to the believability of NPCs, one of which focuses on providing evidence of intentionality. This concept was examined through an analysis of a specific character in the game “Oblivion” (Lankoski and

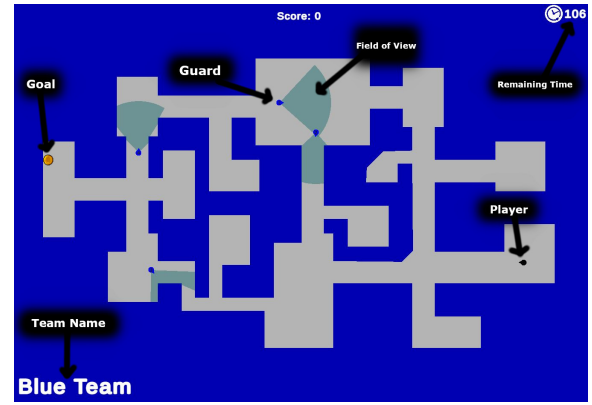


Figure 1: A screenshot of the game’s features presented to the user study participants.

Björk 2007). Such investigations into NPC behavior contribute valuable insights into enhancing the design and development of NPCs within video games.

Furthermore, additional research endeavors have aimed to utilize learning-based NPCs in order to predict player enjoyment. This is achieved by defining an interest value based on a set of metrics, with player enjoyment assessed through post-game questioning after experiencing various NPC behavior variations (Yannakakis and Hallam 2007b,a). While the investigation of player enjoyment in stealth games is relatively scarce, this particular study explored a related problem variation in the form of a predator/prey scenario, which shares similar properties to our own setup.

Prototype Scenario

Each participant engaged in a game comprising three rounds, where they encountered guards with distinct patrol behaviors. Each round lasted 120 seconds, during which the participant controlled a character within the game level. The primary objective was to collect randomly spawned coins to increase their score.

The game level featured four guards, all assigned the same patrol behavior. If any of the guards detected the player’s character, their score would gradually decrease over time. To ensure that participants could solely focus on observing the patrol behavior, the guards were programmed to ignore the player if discovered and continue their predetermined patrol patterns without any interruptions. We allowed for negative scores to maintain a continuous impact on the player’s actions throughout the round, even if they never collected a coin. Collecting each coin had a predetermined fixed amount that would positively contribute to the player’s score, motivating them to strategically navigate to specific locations within the game level. Figure 1 shows an illustration of the game elements.

We presented the game as a top-down real-time game, with the main elements being:

- Game level: The game level consisted of a polygon shape with holes representing the walkable area. The walkable area within the polygon was depicted in grey, while the

areas deemed unwalkable were colored according to the assigned color of the current guard team. To ensure optimal visibility and observation of the guards' behavior, we ensured that all walkable areas were constantly visible to the player throughout the game.

- **Guards:** The game level featured four guards positioned randomly. To ensure a uniform distribution of the guards, the initial guard was randomly placed on a corner of the outer polygon. Following that, each subsequent guard was positioned on a corner representing the farthest path from the previously placed guards. This process was iterated until all four guards were appropriately positioned within the game level.
- **Coin:** Similar to the guards, a coin would randomly appear on one of the corners at the edge of the game level. The coin acted as a goal for players to collect and increase their score. Once the player successfully picked up the coin, it reappeared on the corner farthest from their current location. This placement rule ensured a uniform level of difficulty for all players. Throughout each round, only one coin was available at any given time, maintaining a strategic and competitive gameplay experience.
- **Intruder:** The intruder is the character within the game controlled by the player. The player's main objectives are twofold: firstly, to avoid being detected by the guards to prevent their score from decreasing, and secondly, to enhance their score by collecting coins. To ensure a consistent level of challenge for all participants, the player-controlled character begins the game at the corner farthest from the coin and the guards. This starting position contributes to a balanced difficulty level and provides equal opportunities for all players.
- **Time:** After a time limit of 120 seconds, the round is concluded. The remaining time is continuously displayed on the screen, allowing participants to monitor it throughout the round.

To mitigate any biases, the order of the rounds was randomly assigned. Furthermore, to aid in distinguishing between the guard teams, each team was assigned a random color. The randomization of colors aimed to prevent players from forming associations between the guards' behavior and the specific color assigned to them. In the subsequent section, we will investigate the details of the three patrol behaviors employed in each round.

Methodology

This section defines the dynamic patrol behaviors and the user-study setup for evaluating these behaviors.

Patrol Behaviors

The choice of patrol behavior determines the way guards move in space. Our choices were limited to behaviors that result in dynamic patrol without hard-coding the patrol patterns; thus, we chose three behaviors for this study:

A. VisMesh This method was first introduced in (Al Enezi and Verbrugge 2020). This method relies on distinguishing

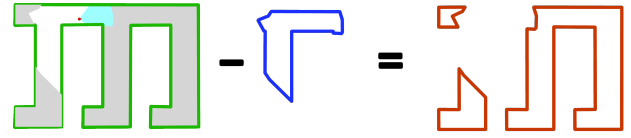


Figure 2: The covered area (blue) is subtracted from the overall space geometry (green) and results in the uncovered areas (red). Taken from (Al Enezi and Verbrugge 2020).

the surveyed area by a guard so that it will prioritize surveying the unsurveyed places. This method only handled scenarios of a single guard. For this paper, we extend it to support multi-guard scenarios.

The Vismesh method generally works by modeling the walkable space by a mesh of convex polygons, similar to the navigation mesh. This mesh is constantly updated every fixed time step to reflect how often the area is surveyed. The main idea is to split the area into covered and uncovered areas.

After that, we partition the covered and uncovered areas into a mesh of convex polygons. We will have convex polygons in the seen area, called seen polygons, and the rest are in the unseen area, called unseen polygons. These polygons form the Vismesh.

Each polygon in the Vismesh is associated with a numerical value, referred to as staleness. It indicates how long time has passed since the corresponding area is surveyed. So in seen polygons, the staleness will be 0. As for the unseen polygons, the staleness is a weighted average of the staleness of the intersecting polygons from the previous Vismesh.

The original method was tested on one guard. Every time the guard requested a new decision to be made, it chose one of the unseen polygons and found the shortest path toward the centroid of it. However, to extend this method to multi-guard scenarios, we defined new heuristics that determine which unseen polygon each guard will choose.

At every time, the guard chooses an unseen convex polygon to cover. To explore the multiple factors that can affect the patrol behavior, the selection is made by evaluating the fitness value for all polygons and choosing the one with the highest fitness value. In our extension, guards share the same Vismesh, and they decide their next move based on a weighted sum of the following heuristics:

- The normalized area of the polygon. The larger the convex polygon is, the higher this value is. We normalized this value to simplify the fitness calculation.
- The staleness of the polygon. It is a normalized value of how long the guards have not covered a polygon. A value of 1 means that this is the oldest polygon, and 0 means that the polygon has just been covered.
- The normalized distance of the centroid of the polygon from the guard. It gives a sense of how far a polygon is from the guard. So naturally, a guard would prioritize close polygons, for example. The distance is normalized by dividing the map's shortest path distance by the longest path distance.

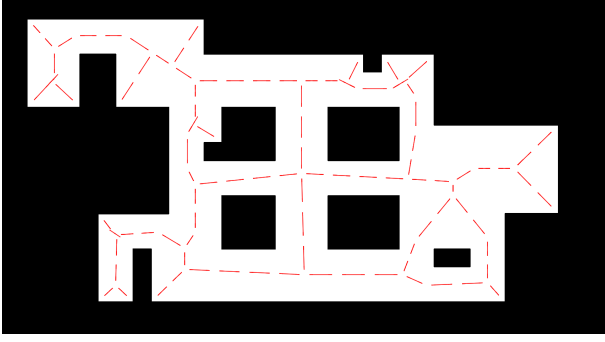


Figure 3: The divided graph of the Docks map of Metal Gear Solid. Taken from (Al Enezi and Verbrugge 2021)

- The normalized distance of the closest guard of the others to the centroid. We included this value to provide a degree of separation between the guards. So a guard may prioritize polygons that are further away from the locations of other guards.

To calculate the fitness of a polygon, we find the weighted average of the properties by using equation 1. We define the values of the weights by exploring the possible combination of weights and choosing the best performance according to the uniform coverage performance.

$$f(n) = v_a * w_a + v_s * w_s + v_d * w_d + v_g * w_g \quad (1)$$

where $f(n)$ is the fitness value of a polygon n in the VisMesh. The variables v_a, v_s, v_d, v_g are the polygon's properties area, staleness, distance, and distance from the closest other guards. Each variable has the corresponding weight value, which for simplicity, we limited their values to between 0 to 1.

B. Skeleton-Based Roadmap This method was designed to search for an adversary after they escaped from the guards' FOV (Al Enezi and Verbrugge 2021). It uses the Scale-axis transform to simplify and capture the environment's topology. This graph serves as a roadmap to simplify the search for the adversary instead of uniformly searching the map. Each edge is discretized into line segments for precision in associating areas with the roadmap. Figure 3 shows an example of the roadmap.

Once the guards lose sight of an opponent, this method propagates the probability of them on a line segment through the roadmap. Then the guards will try to cover the line segments in a certain order to find the intruder.

We modified two main features in this method. First, instead of having the probability propagate from the position the opponent was last seen, we set all segments to have the same probability value. This is because, at the start of a patrol scenario, guards have no knowledge of the presence of an opponent. Secondly, we modified the form of decision-making guards make.

In the original method, guards choose a specific line segment and find the shortest path toward it. Which caused the guards to sometimes overlap, leaving other possible

paths unattended. To mitigate that, we modified the decision-making so that guards plan a full path instead of a destination point. Once a guard determines a path, we update the corresponding line segments so that when other guards plan a path, they choose line segments with no or fewer guards passing through.

First, we get the closest segment in the roadmap to guard g ; It represents the start segment of the potential path to be built. After that, similarly to the Dijkstra algorithm, we build a path by exploring the path with the highest total utility. However, we stopped expanding the search if the total distance reached the defined limit. We expand the search by iterating through the connected segments to the current segment and update the possible highest total utility it can reach along with the total distance to reach it. After the search is over, we backtrack the path from the segment with the highest utility to the start.

To prioritize the line segments, we distinguish them by the fitness of that segment. To calculate the fitness of a line n , we used the function $GetUtility(n)$ which finds the weighted average of properties assigned to that line segment. They are:

- The staleness of the line. It is a normalized value of how long the guards have not covered the center of the line segment. A value of 1 means that this is the oldest line segment, and 0 means that the line segment has just been covered.
- The number of guards planning to pass through this line segment. It indicates that another guard will cover this line segment. We normalized this value by dividing it by the total number of guards, and to make higher values more desirable, we subtracted the result from 1.

$$GetUtility(n) = v_s * w_s + v_g * w_g \quad (2)$$

The fitness calculation is shown in equation 2, where v_s, v_g are the features mentioned multiplied by the corresponding weights.

C. Random We included this behavior as a baseline for the cheapest behavior to create compared to the previous two. Each guard independently finds the shortest path toward a randomly sampled position on the level. Figure 4 shows the heatmap for this behavior.

Patrol Performance Comparison

The two patrol methods have several parameters that affect patrol behavior. To use them in the user study, we need to define the weights used in calculating the fitness. To do that, we explored the possible combination of these values and chose the best combination that provided the best patrol behavior.

To evaluate patrol behavior, we defined a measure of how well the guards covered the map. To formalize that, we laid a grid of $W \times H$ nodes. Over the patrol shift, we aggregated the time each node was covered by one of the guards' FOV. After the shift, we normalized the coverage time for the nodes, so their values will be from 0, which means the node was covered the least, to 1. A higher mode of coverage time per

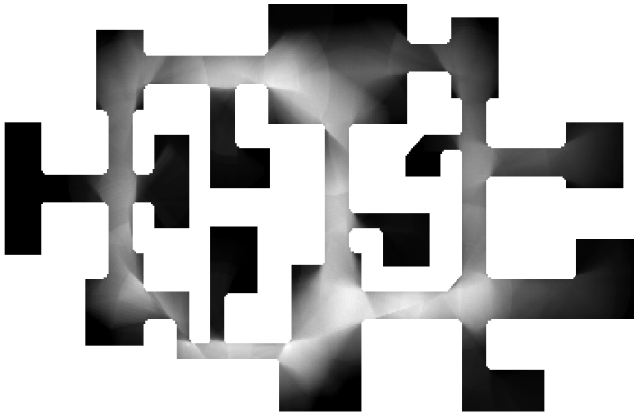


Figure 4: Heat maps of a patrol shift where the guards adapted the random patrol behavior.

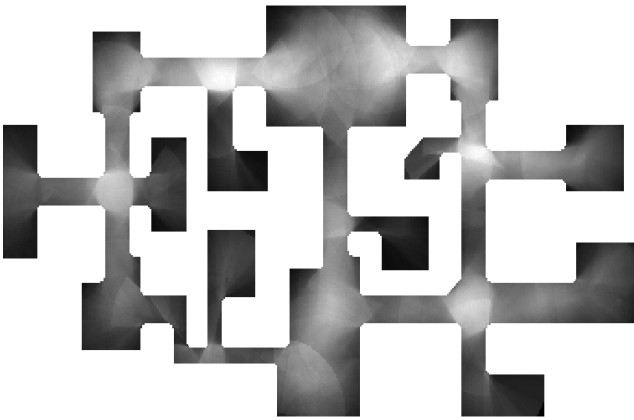


Figure 5: Heat maps of a patrol shift where the guards adapted the VisMesh patrol behavior. The brighter a location is, the more coverage it got.

node on the grid indicates good patrol behavior since the guards covered more areas during the patrol shift. This measure was also used to evaluate robot patrol behavior (Jana, Vachhani, and Sinha 2022).

After testing the combinations of parameters for VisMesh, we found the highest average survey time to be 0.49 for the following values: Area weight: 0, staleness weight: 1, distance weight: 1, and separation weight: 0.5. Figure 5 shows the heatmap for these parameters during a 120 seconds patrol scenario.

As for the RoadMap method, the best parameter combination yielded an average of 0.33 for the following values: Max normalized path length: 0.5, staleness weight: 0.5, passing guard weight: 0, connectivity weight: 0. Figure 6 shows the heatmap for these parameters for a sample game level map.

In order to conduct a quantitative comparison of these patrol behaviors, we examined the overall distribution of coverage for each behavior. Figure 7 depicts the violin plots illustrating the coverage performance of the three patrol methods. Each violin plot represents the distribution of coverage time for the pixels on the game map. A wider upper

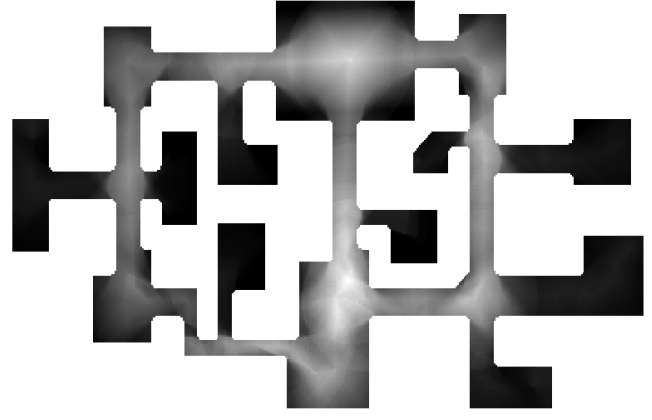


Figure 6: Heat maps of a patrol shift where the guards adapted the Roadmap patrol behavior.

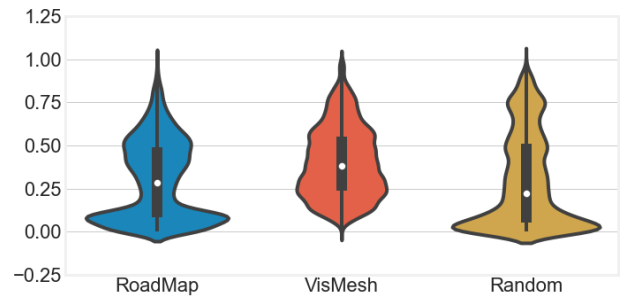


Figure 7: Violin charts at the end of 120 seconds patrol round. Each violin represents distribution of the heat values, so a better uniform coverage results in more pixels with heat values close to 1.

area in the violin plot indicates a greater number of pixels with higher coverage values, indicating better overall coverage in the game level. The results indicate that the Random method exhibited the least uniform coverage, the roadmap method performed better but still fell short, while the Vismesh method offered the most comprehensive coverage.

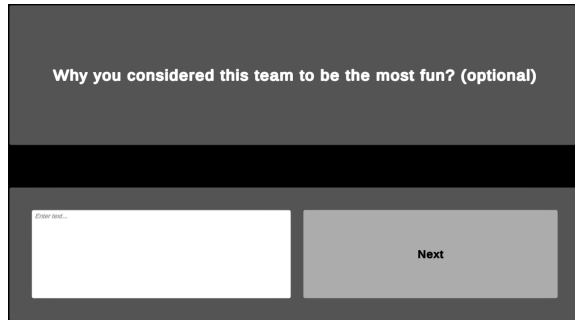
Experimental Setup

To recruit participants for our study, we employed email communication to contact undergraduate and graduate students. The email provided a detailed overview of the study along with a link to an online portal hosting a web-based version of the game. Participants could access the game through the web portal at their convenience. We ensured the anonymity of the participants by collecting their gameplay data and survey responses without any personally identifiable information. On average, the study was completed within approximately 15 minutes.

At the start of the game, all participants were provided with an introduction to the study's objective and given instructions on how to engage in the game. Following this, they had the opportunity to engage in a tutorial level to fa-



(a) Most enjoyable team



(b) Justification

Figure 8: Screenshots of the two questions in the game that pertain to identifying the most entertaining team. We followed them with a similar inquiry about difficulty and effectiveness.

miliarize themselves with the game’s mechanics. Once the tutorial level ended, participants were given a choice to replay the tutorial or commence the actual game when they felt prepared.

Throughout the main game, participants actively participated in three rounds, with each round showcasing guards exhibiting different patrol behaviors. Each team was assigned a distinct color to aid players in distinguishing between the guard teams. The pairing and order of the guard teams and colors were randomized across study sessions to eliminate any potential biases. After completing all three rounds, participants were prompted to indicate the most enjoyable, challenging, and effective teams. Moreover, we allowed them to explain their selections for each aspect via text input, as illustrated in Figure 8. The study concluded at this point, and participants were free to replay the game.

Game Level

For the tutorial round, we utilized a Metal Gear Solid map as depicted in figure 9. We selected this map due to its relatively straightforward design, which aimed to acquaint players with the game mechanics. In the main game, considering the expected limited sample size for this user study, we employed a single fixed map to increase the likelihood of obtaining statistically significant results. This map was intentionally crafted to resemble the “Skeld” map from the game “Among Us”, providing a moderate level of challenge

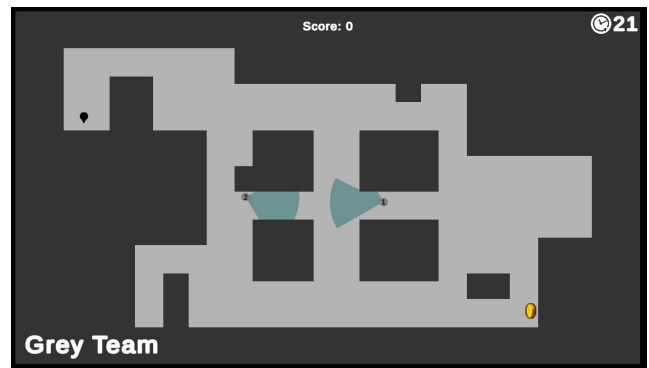


Figure 9: The Docks map from the commercial game Metal Gear Solid.

with multiple cycles and diverse enclosed spaces that facilitated player hiding, thereby adding an intriguing aspect to the map. The map’s layout is illustrated in figure 1.

Guard Teams

For the tutorial round, we limited the number of guards to two to assist participants in understanding guard movement within the game space and to prevent overcrowding due to the relatively smaller level. In the main game, we conducted multiple rounds of testing different guard team formations and ultimately settled on populating the game level with four guards. Regarding the assigned behaviors, we selected the tutorial team with the Random patrol behavior, while the subsequent three rounds featured the three behavior patterns we described in a randomized order. In the following section, we present the outcomes of this study.

Results

In this section, we describe the number of participants in our study. After that, we report the participants’ performance based on their gaming experience and against the patrol behaviors. Then, we analyze and report the possible features contributing to players’ enjoyment and perceived difficulty in patrol behavior.

Participation

We recruited a sample size of 115 that completed the game. We asked the players for their self-perceived familiarity with video games. Figure 10 shows a bar chart of the distribution of participants according to their experience with video games. We found most participants classified themselves as Advanced or Intermediate experience with games. The ratio makes us believe we generalize our findings to video game players.

To confirm if our prototype was easy for the different participant experience levels to learn and play, we compared the average scores each group achieved through the three rounds they played. We found that advanced and intermediate players, which made up the majority of participants, had consistent scores through the rounds.

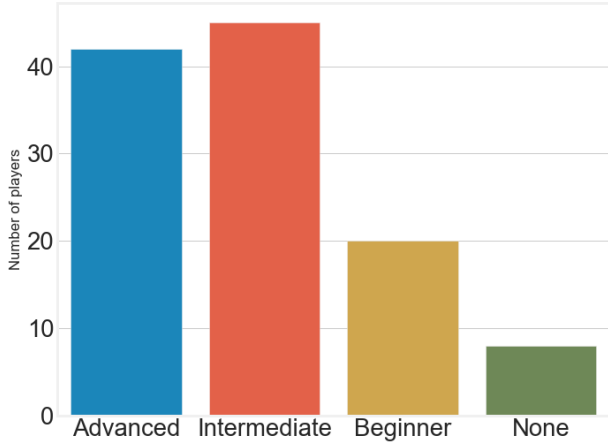


Figure 10: The numbers of players who participated in the study grouped by their respective experience in video games.

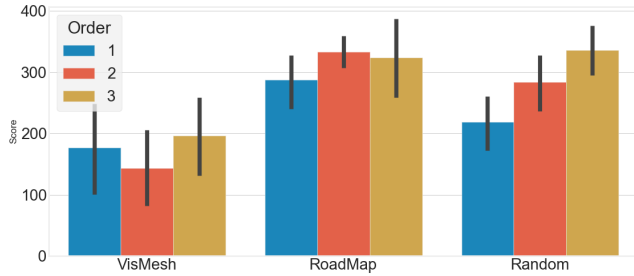


Figure 11: Players' scores against the patrol method grouped by the round where they encountered each behavior. The error bars represent 95% confidence intervals.

Performance

To confirm if each guard team impacted participants' performance differently, we compared their scores according to the round and the type of behavior they played against. Figure 11 shows a bar chart of players' scores against each guard team grouped by the round they encountered the corresponding team. Generally, participants scored the most against the RoadMap team, followed by the Random team, and finally, they significantly scored the lowest against the VisMesh team. As for the effect of the order of the round on the score, participants seemed to score slightly better if they encountered the Random team in a later round. This could be a result of the players becoming more familiar as they played the game; however, the same pattern cannot be seen in the Roadmap and Vismesh. Roadmap could have been easy enough to allow players to score better regardless of the order, and the Vismesh was consistently harder to beat.

Enjoyment

After players played against all teams, we asked them to choose the most enjoyable team and the option to insert a

	χ^2	p
Enjoyment	5.46	0.06
Difficulty	55.76	<0.001

Table 1: The Chi-square goodness-of-fit test results of the players' most enjoyable behaviors. For $\alpha = 0.05$ and degrees of freedom = 2, the critical value is approximately 5.991.

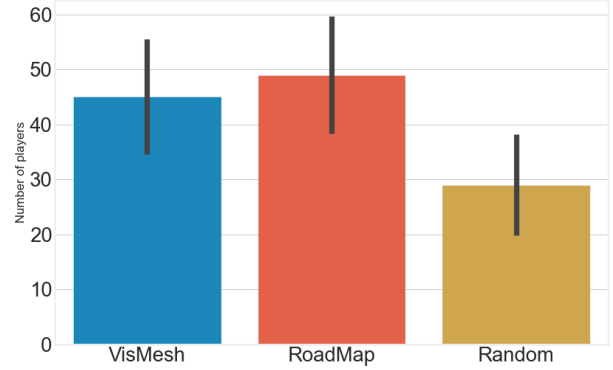


Figure 12: Players' rating of fun for different teams. The error bars represent 95% confidence intervals.

justification for their choice in free text. Figure 12 shows bar charts of players rating of teams in terms of fun. We observe that players found the Roadmap and the Vismesh to be the most enjoyable, and fewer players chose random behavior. Table 1 shows that we cannot reject the null hypothesis; thus, there is no statistical significance of participants enjoying one behavior over the other. Despite this, we believe a larger sample may provide better insight.

To understand how participants enjoy playing against a specific behavior, we investigate several factors that might have affected this aspect of the player's experience. First, we found no significant impact the team's color had on player enjoyment. Therefore, the other possible factors include:

Order The order participants played against a specific behavior might affect their enjoyment of that team. For example, playing against a difficult team, followed by an easier team, might affect the player's enjoyability differently than if it was the reverse order. First, we confirmed that the teams had a uniform chance of being ordered in a particular order for the study, so no specific ordering of teams was significantly repeated more than the others. Figure 13 shows a bar chart of the patrol behavior participants chose as most enjoyable, grouped by the order they appeared in. As we mentioned before, participants reported enjoying the random the least among the other teams; however, the later they played against the Random behavior, the more likely they considered it as the most enjoyable. Justifying this answer would require further testing; however, it is possible that since players scored more against Random behavior in later rounds, they found it to be more enjoyable since they

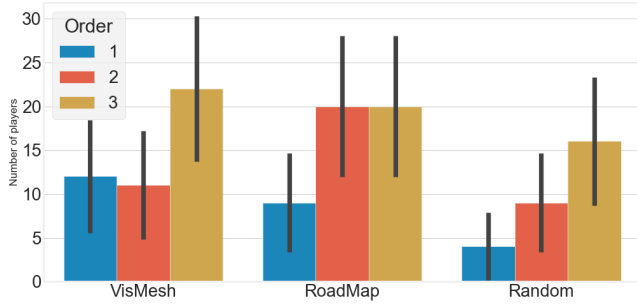


Figure 13: The number of players rating patrol behaviors as most enjoyable grouped by the order the behavior appeared in. The error bars represent 95% confidence intervals.

learned the game from playing against the previous teams. As for the other two behaviors, participants enjoyed them the most when they played with them in the last episode. We believe that it is less likely due to participants' scores against these two behaviors; however, it could be because they were the most recent team thus or because as players got more familiar with the game, they got better at observing guards' behavior and thus enjoyed the complexity of these two methods. Additionally, RoadMap behavior seemed to be more enjoyable starting from the second round, meaning that participants remembered it the most, which the high score they achieved against RoadMap behavior might affect it. However, this is not the case for VisMesh since many players enjoyed it regardless of the lower average score they achieved against it. Participants' preference for VisMesh despite the lower average score suggested the existence of "challenge" as a factor of enjoyment.

Challenge After we reviewed the participants' justification for their choice of the most enjoyable team, we found the primary criterion for many players to be the amount of challenge the guard team demonstrated. Players ranged in their preference from easy guards that allowed players to score better to guards that were the most challenging as they were motivating players to improve in the game. Further, we explored if the players enjoyed the team that allowed them to score the most. We found that 36% of the players chose the behavior that allowed them to score the best. Some players enjoyed the behavior because it was easy. However, an almost equal proportion of players, around 40%, preferred other more challenging behaviors and specifically mentioned their preference for such teams because of their challenging nature.

Predictability Several players considered less predictable guards to be more enjoyable to face. Players seemed to enjoy competing with an AI that showed more emergent behavior, given that this behavior demonstrated a certain level of effectiveness. For example, 20% of the players mentioned they preferred a team because they were unpredictable, and most of them designated VisMesh guards as such.

Effectiveness A team that seemed more effective at their task was more fun to face; This could be because players

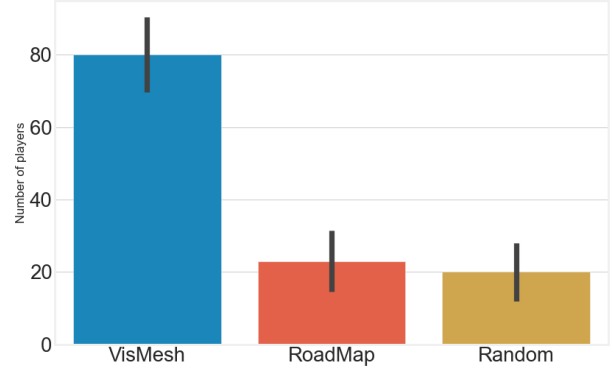


Figure 14: The number of players rating patrol behaviors as most challenging. The error bars represent 95% confidence intervals.

felt more achieved when they competed with a competent AI. In other words, to challenge players while demonstrating that the AI is showing reasonably effective behavior. As part of this study, we asked players to rate which was effective, so we checked if players chose the same team as effective and fun. We found that 37% of players considered teams they classified as effective to be fun as well. In addition, 20% of the players mentioned the team to be well-spread, 5% as meticulous or natural. Examples of the comments we received: "It checked corners of the room it entered, so if it trapped you, you suffered and couldn't just escape unnoticed by hiding in a corner", "This team provides a significant challenge by covering a large amount of ground and move in a more natural pattern, scoping the area more thoroughly with movements that seem instinctive."

Difficulty

As for the difficulty, players distinctively chose a specific team as the most difficult. Figure 14 shows a bar chart of the results of their responses. We found that participants chose the Vismesh as the most difficult, while the same number of players split between choosing Roadmap and Random as the most difficult. The Chi-square goodness-of-fit in Table 1 shows that there is a statistically significant number of participants who considered the VisMesh to be the most difficult.

Figure 15 shows a bar chart of participants' choice of the most challenging team grouped by the order the team showed up in the study. We found no significant impact on their choice by the order; however, the Vismesh seemed to have an incremental pattern, so the later they faced it, the more they considered it the most difficult. This result is insignificant and would require more testing to confirm.

As for the justification for their choice of the most difficult team, we found the main traits participants mentioned were:

Meticulousness 50% of players mentioned how they perceived a team to be difficult when checking rooms and the corners of these rooms. Also, to be well-spread, so they had

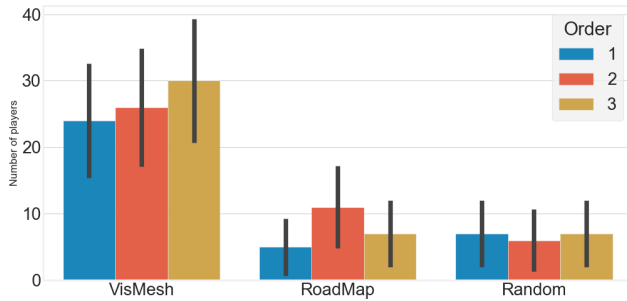


Figure 15: The number of players rating patrol behaviors as most challenging grouped by the order the behavior appeared in. The error bars represent 95% confidence intervals.

less overlap and covered more space. We found this trait, and the previous can describe an effective patrol behavior and confirm the possibility of a relation between players' perception of a challenging and effective AI behavior in this study. After comparing the correlation between players' scores against the teams, they classified as challenging and effective; we found a strong linear relationship (Pearson = 0.83, p -value < .05) between these properties.

Unpredictability Unpredictability was standard trait players mentioned in a difficult AI. Interestingly, many players who mentioned this observation also chose the same team as the most enjoyable team, which made us believe that players do enjoy a certain level of unpredictability.

Limitations

This study has several limitations. Firstly, the sample size was relatively small and confined to university students. Despite the results showing clear trends in player enjoyment, we believe that a larger sample size is important. Moreover, relying on self-reported measures for data collection may have introduced response bias and recall errors, which could affect the data's accuracy and reliability. The short study duration limited the exploration effect of player adaptability to the guard behaviors over time. Lastly, the investigation centered on a top-down game perspective, potentially overlooking the diverse effects elicited by different perspectives. Despite these limitations, the study provides valuable insights into dynamic patrol behavior, prompting future research to address these shortcomings and offer a more comprehensive understanding of its impact on player enjoyment.

Conclusion & Future Work

Developing sophisticated NPC behavior does not always contribute to improving the players' experience. Our evaluation showed that players have different preferences for enemy AI. In our study, some players preferred the difficult NPCs, while others chose the easier ones. Roadmap and Vismesh behaviors were rated as the most enjoyable, while the Random behavior was the least preferred. Factors like challenge, unpredictability, and effectiveness influenced enjoyment. The Vismesh behavior was perceived as the most

challenging due to meticulousness and collaboration.

In future work, it would be beneficial to expand the participant pool to include a diverse range of players and explore the impact of factors such as game genre and level design on enjoyment and performance. Additionally, investigating hybrid patrol behaviors that combine different methods and identifying strategies for enhancing player satisfaction would contribute to a deeper understanding of NPC behaviors and the development of more immersive gaming experiences.

Acknowledgments

This work supported by the COHESA project, through NSERC Strategic Networks grant NETGP485577-15.

References

- Al Enezi, W.; and Verbrugge, C. 2020. Dynamic guard patrol in stealth games. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 16, 160–166.
- Al Enezi, W.; and Verbrugge, C. 2021. Skeleton-based multi-agent opponent search. In *2021 IEEE Conference on Games (CoG)*, 1–8. IEEE.
- Campbell, J.; and Verbrugge, C. 2019. Exploration in NetHack With Secret Discovery. *IEEE Transactions on Games*, 11(4): 363–373.
- Hagelback, J.; and Johansson, S. J. 2008. Dealing with fog of war in a real time strategy game environment. In *2008 IEEE Symposium On Computational Intelligence and Games*, 55–62.
- Hingston, P. 2009. A Turing test for computer game bots. *IEEE Transactions on Computational Intelligence and AI in Games*, 1(3): 169–186.
- Iida, H.; Takeshita, N.; and Yoshimura, J. 2003. A metric for entertainment of boardgames: its implication for evolution of chess variants. In *Entertainment Computing*, 65–72. Springer.
- Isla, D. 2005. Probabilistic Target-Tracking and Search Using Occupancy Maps. In *AI Game Programming Wisdom 3*. Charles River Media.
- Isla, D. 2013. Third Eye Crime: Building a stealth game around occupancy maps. In *Ninth Artificial Intelligence and Interactive Digital Entertainment Conference*.
- Jana, M.; Vachhani, L.; and Sinha, A. 2022. A deep reinforcement learning approach for multi-agent mobile robot patrolling. *International Journal of Intelligent Robotics and Applications*, 6(4): 724–745.
- Johansson, M.; Eladhari, M. P.; and Verhagen, H. 2012. Complexity at the cost of control in game design. In *Proceedings of the 5th Annual International Conference on Computer Games and Allied Technology (CGAT 2012)*. Global Science & Technology Forum, 22–29. Citeseer.
- Lankoski, P.; and Björk, S. 2007. Gameplay Design Patterns for Believable Non-Player Characters. In *DiGRA Conference*, 416–423.

- Moravec, H. P. 1989. Sensor fusion in certainty grids for mobile robots. In *Sensor devices and systems for robotics*, 253–276. Springer.
- Paull, L.; Seto, M.; Leonard, J. J.; and Li, H. 2018. Probabilistic cooperative mobile robot area coverage and its application to autonomous seabed mapping. *The International Journal of Robotics Research*, 37(1): 21–45.
- Soni, B.; and Hingston, P. 2008. Bots trained to play like a human are more fun. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, 363–369.
- Sutiono, A. P.; Purwarianti, A.; and Iida, H. 2014. A mathematical model of game refinement. In *International Conference on Intelligent Technologies for Interactive Entertainment*, 148–151. Springer.
- Sutiono, A. P.; Ramadan, R.; Jarukasetporn, P.; Takeuchi, J.; Purwarianti, A.; and Iida, H. 2015. A Mathematical Model of Game Refinement and Its Applications to Sports Games. *EAI Endorsed Transactions on Creative Technologies*, 2(5).
- Wetzel, B.; and Anderson, K. 2017. What You See Is Not What You Get. In *Game AI Pro 3: Collected Wisdom of Game AI Professionals*, 31–47. CRC Press.
- Xu, Q.; Tremblay, J.; and Verbrugge, C. 2014. Generative methods for guard and camera placement in stealth games. In *Tenth Artificial Intelligence and Interactive Digital Entertainment Conference*.
- Yannakakis, G. N.; and Hallam, J. 2007a. Capturing player enjoyment in computer games. In *Advanced Intelligent Paradigms in Computer Games*, 175–201. Springer.
- Yannakakis, G. N.; and Hallam, J. 2007b. Towards optimizing entertainment in computer games. *Applied Artificial Intelligence*, 21(10): 933–971.