The Effect of Visualising NPC Pathfinding on Player Exploration

Madeleine Kay
Games Academy
Falmouth University
United Kingdom
mk186442@falmouth.ac.uk

Edward J. Powley
Games Academy
Falmouth University
United Kingdom
edward.powley@falmouth.ac.uk

ABSTRACT

Foregrounding AI behaviour through visualisation is an under-explored but potentially powerful technique for influencing player behaviour, particularly in terms of sparking the player's curiosity and encouraging them to spend more time exploring the game environment. This paper looks at visualising various methods of enemy NPC pathfinding in a 3D Metroidvania game. We demonstrate a number of cases where player exploration and player deaths are affected by either a change in pathfinding method or by visualising the pathfinding process. This suggests that the choice of pathfinding algorithm and the option for this to affect the visuals of the game are useful considerations for game designers.

ACM Reference format:
Madeleine Kay and Edward J. Powley. 2018. The Effect of Visualising NPC Pathfinding on Player Exploration. In *Proceedings of FDG, Malmö, Sweden, August 7-10, 2018 (FDG'18),* 6 pages.
DOI: 10.1145/3235765.3235824

1 INTRODUCTION

Curiosity in games can take many forms, but one of the most obvious forms of curiosity is exploration. Rather than trying to reach the goal as quickly as possible, one expects a curious player to explore off the beaten track. Thus a game designer might consider how best to encourage this behaviour, to provide a richer, longer and more varied gameplay experience.

In most games, AI systems work in the background to produce the desired experience; the player is generally not aware of them. An interesting idea is to turn this on its head, pushing the AI into the foreground so that the player must engage with it (as a system and not just as a simulated intelligent agent) as a core part of the game. This paper looks at whether visualising AI pathfinding can encourage players to become more curious and take a more exploratory approach to traversing the game environment.

This paper looks at visualising the pathfinding of an enemy *Non Player Character* (NPC) using *Rapidly-exploring Random Tree* (RRT) pathfinding and Unity's *Navigation Meshes* (NavMeshes), and the effect this has on player exploration. This was tested in a 3D Metroidvania game, which collected data on the amount of time the players spent in a level and what percentage of the level they explored. Analysis of this data found a number of cases where the different pathfinding methods and visualisations affected the percent of the level explored and the number of player deaths but no cases where the time spent in the level was affected to a statistically significant level. These are analysed in more detail in Section 5.

As Section 2 shows, previous papers explore visualising and foregrounding AI and pathfinding. However, there is little on using this in digital games beyond game design. The potential impact of the results of this study could be in game design as the use of different pathfinding techniques here have an effect on how players explore. Game designers could take this into account when designing games as they could design environments to promote exploration and curiosity or to funnel players down a specific path. The research question addressed in this paper is: how does visualising pathfinding in an NPC affect how a player explores a game level?

The remainder of this paper is structured as follows. Section 2 is a review of the literature on existing relevant work. Section 3 details the methodology used in the experiments in this study. Section 4 details the data collected and how it was filtered and Section 5 looks at the analysis of this data and what the results imply. Section 6 details the potential issues and ways to address them in future work. Section 7 looks at future research that could be done on this subject. Finally Section 8 provides some concluding remarks.

2 RELATED WORK

2.1 A* Pathfinding

Hart *et al* [1] first proposed A* pathfinding in 1968 as an improvement on Dijkstra's algorithm. A* aims to expand the fewest nodes possible to minimise the cost of the path where the cost is the distance between the start and goal nodes. Algfoor *et al* [2] survey numerous papers on pathfinding. Their focus is on the use of different grid shapes in pathfinding and the numerous algorithms available. They state that A* is the most popular pathfinding algorithm in digital games, and is also widely used in robotics.

Nash *et al* [3] say that A* cannot always find the true shortest path in 2D or 3D space as it is limited to a grid. The shortest path can however be found using A* with post-smoothing paths or by using A* variants such as Theta* [3, 4]. Theta* expands on A* as it allows for all edges and angles in the grid to be used. Firmansyah *et al* [4] compare A* with Theta*. They found that they performed similarly time wise. However, A* produces a path with fewer nodes expanded and Theta* produces a shorter path.

FDG'18, Malmö, Sweden

978-1-4503-6571-0/18/08…$15.00
DOI: 10.1145/3235765.3235824

2.2 RRT and Pathfinding

Rapidly Exploring Random Trees (RRT) is commonly used in robotics [5, 6]. LaValle [5] first proposed RRT in 1998, with the intent to produce a random algorithm more efficient than the other search algorithms available at the time. The process for RRT involves the random placement of nodes. A parent is then selected by finding the closest pre-existing node [6].

The goal of RRT is to find a path between two points with no collisions, however the path found may not be optimal [6, 7]. Karaman and Sertac [8] say that the chance of RRT finding an optimal path is very unlikely [8, 9]. Whereas A* is guaranteed to find the shortest path as seen in Section 2.1, RRT is unlikely to find the shortest path and may not even find a path at all. However RRT is better suited to unknown environments, whereas A* assumes the environment is known in advance.

2.3 Foregrounding and Visualising AI

Most modern digital games make use of AI. As this paper looks at visualising NPC AI existing methods for foregrounding and visualising AI were looked at. Treanor *et al* [10] say that often the design of AI in games is to fit the game and complement gameplay. These AI are supporting the gameplay rather than being central to it. Few games *foreground* the AI: generally the AI works in the background to produce the desired experience, without the player being aware.

Treanor *et al* [10] survey many games that foreground or visualise AI in different ways. From this, they propose a series of design patterns for foregrounding AI in digital games. The two design patterns relevant to this paper are "AI as a Villain" and "AI is Visualised". They describe the first pattern as having the AI try to not outright defeat the player, but to create an experience. An example of this pattern is Alien Isolation [10, 11], a game in which an enemy AI hunts the player. This is foregrounding as the player must observe the AI and learn how to avoid it. There is also some visualisation as the player has a scanner that informs them of the enemy's position.

This paper uses the "AI as a villain" pattern as each enemy NPC has their pathfinding visualised around them. The players can consider this when exploring a level so they do not get attacked by the enemy. The use of this pattern also aims to have an NPC that creates an experience rather than one that always finds the player. While the player likely does not want to be caught by the enemy NPC they may want it to chase them so they can learn its patterns or lead it away from other enemies to make it easier to attack.

The second relevant design pattern is "AI is Visualised". This is where there is a visual representation of the AI's state or decision making in the game [10]. Most games hide this from the player but this design pattern visualises it making it a mechanic. The example given by Treanor *et al* [10] is the game Third Eye Crime [12, 13]. The game uses probabilistic object tracking through Occupancy Maps. As the enemy moves around the map it removes areas where the player is not from the Occupancy Map [12]. Generally, stealth games involve avoiding enemies. This design encourages the player to trigger the mechanic, allowing them to use the visualisation to mislead and avoid the enemy [12, 13]. This pattern is relevant to this paper as the visualisation of pathfinding allows the player to see where the enemy is searching and react accordingly.

Haworth *et al* [14] visualise the possible decisions available to the player in a game on a tree structure. They visualise decision trees in a game to see what effect it has on gameplay and the analytical reasoning of children. Their results suggest that the trees aided players as in the later levels of the game the children without the visualised decision tree struggled to beat the game. However, they noted this could also be due to unbalanced difficulty in the later levels.

2.4 Applications of Pathfinding

Pathfinding is most commonly used in games for NPC pathfinding. However it can also be a foregrounded aspect of the game, like in this paper, or it can be used to aid level design. Bauer and Popovic [15] use RRT for level design to calculate the possible routes the player could take and visualising the data to aid level designers or to analyse procedurally generated levels. The output of the tool may be difficult to for designers to interpret, so they use van Dongen's method [16] for graph clustering to make the output more legible [15].

Mendona *et al* [17] look at pathfinding both in robotics and digital games. Their focus is on stealth pathfinding in games and applying that to robotics. Like RRT, the methods they propose do not necessarily find the shortest path [8, 17]. Instead, they try to find the path where the agent spends most of the time in cover. They generate custom NavMeshes and assign a weight to each polygon in the NavMesh depending on how close it is to being behind cover.

Tremblay *et al* [18], like Bauer and Popovic [15], also use RRT visualisations to aid level design and clustering to make the results less cumbersome to the user [18]. Their system allows level designers to see where players are likely to go and adjust the level design accordingly [18]. The use of RRT here is because it is flexible and inexpensive. Also, its random nature allows for the mimicking of a wider range of player behaviours [18]. However the chances of RRT finding a path decrease as the grid size increases, and also as the number of attempts decreases. This suggests that a potential issue with RRT is that it may not always find a path.

In a subsequent study, Tremblay *et al* [9] compare several variants of both A* and RRT, as well as hybrids of both. While A* is consistently fast with high success rates, RRT has varying results. Both of Tremblay *et al*'s [9, 18] papers show that a potential issue with the use of RRT is that it may not find a path, even when run offline thousands of times. In contrast A* is theoretically guaranteed to find a shortest path if one exists. However RRT explores the search space more widely, and produces a partial tree even if a path is not found, so may produce more pleasing visualisations which may still influence players.

2.5 Exploring Game Environments

This paper is researching how player exploration is effected by a change in pathfinding method and visualisation. Therefore, how players explore and how that can be influenced is looked at in this section One method of guiding players through games is to use *wayfinding*. Wayfinding in games is often architectural differences or visual cues in the environment that guide the player to an area of interest [19, 20]. Wayfinding cues are often subtle cues in the environment such as ivy growing up a wall to suggest the player can climb there.

The intention of visualising pathfinding in this paper is not to guide the players. However, like Si *et al* [19] it observes how players navigate and explore levels and whether, like the presence of wayfinding cues, it affects player behaviour. Moura and Bartram [21] investigate the effects of different wayfinding cues on players. They looked at methods used in AAA games and mimicked them in their own game. Their results show that the absence of wayfinding cues was obvious to players. In contrast, the version with wayfinding cues does not have enough cues to sufficiently guide the player. These results suggest that wayfinding cues alone may not be enough to guide the player. They concluded that there is a need for more research as the results were inconclusive.

While this paper focuses on enemy NPC pathfinding this could be another interesting application of pathfinding and visualisation. The pathfinding could remain hidden from the player, as it normally is in digital games, but wayfinding cues could be placed based on the path. This could then subtly guide the player through the game.

Si *et al* [19] investigated how players explore virtual environments. While their experiments were specific to Real Time Strategy (RTS) games the results may apply to other game types. Si *et al* [19] say that three common types of spatial exploration are; environment mapping, bonus item collecting and location/landmark discovery. The relevant exploration type for this paper is spatial mapping. Firstly, as it logs what the enemy NPC is doing. Secondly, as it is the player behaviour that is being measured by the logging tool in the software.

3 METHODOLOGY

An a-priori power analysis showed that 52 participants would be sufficient to demonstrate an effect size of 40% for any of the quantities measured in this study. As not all datasets were complete further participants were tested resulting in 62 sets of results. The playtesting was completed by participants both online and in person to reach the required sample size.

Of the playtests done in person, a large majority of the players were students studying game development; the potential issues relating to this are discussed in Section 6. The rest of the playtesters were recruited online through Reddit. Many of the subreddits it was posted on were related to gaining participants and playtesters. Where the in-person playtesting has the issue of the players all being students, the online players could have a similar problem as the players are all users of the subreddits listed above that are looking for surveys to complete. Some of the subreddits listed were also related to game development and therefore the feedback given was related to the state of the game, not the pathfinding.

While 21 playtests were completed online 10 of them have incomplete or invalid data. Therefore, those participants' results were filtered out of the final dataset and more tests were done in person to reach the required sample size. The filtration of the data is discussed in more detail in Section 4.

The game the pathfinding variations have been tested in is 'Gates of Amenti', a 3D Metroidvania game which has been designed with a focus on exploration. A Metroidvania game is an action-adventure game with a focus on exploration to discover new areas and power-ups [22]. 'Gates of Amenti' was developed as a final-year project at Falmouth University by a team of game development students including the present first author. There are four variations of the game, shown in Table 1 and Figure 1. Variations 1 and 2 use a custom implementation of RRT for enemy pathfinding, whereas Variations 3 and 4 use the Unity engine's built-in implementation of A* over navigation meshes. Variations 1 and 3 overlay the game with a visualisation of the RRT tree and the navigation mesh respectively.

A random number generator was used to randomly select a game version for each participant. The players were asked to play that version for as long as they wanted. While playing, the software logged the time the player spent in the level per life and what percent of the level they explored. The percent explored is calculated by recording how many doors the player walks through as the game requires players to unlock doors to progress. The data were collected and analysed, using T-Tests and correlations. Independent sample T-Tests were used as the independent variable is always true or false.

Table 1: Playtest Variations

Playtest Variations	Pathfinding Method	Pathfinding Visualised
Variation 1	RRT	Yes
Variation 2	RRT	No
Variation 3	NavMesh	Yes
Variation 4	NavMesh	No

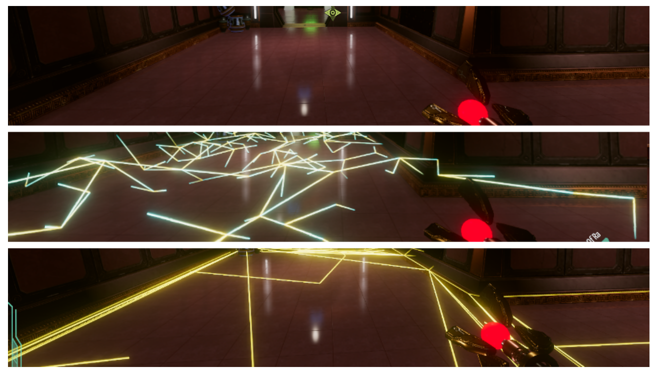

Figure 1: Screenshots of the playtesting game with no visuals, RRT visualised and NavMeshes visualised.

4 DATA COLLECTION AND FILTERING

There are 107 datasets collected from the 62 participants. However, as not all participants gave complete datasets 10 datasets have been removed leaving 97 datasets from 52 participants.

When the player dies in-game the game resets, this leads to a number of results having 0 percent explored and low play times. As these were not proper playtests the 11 instances of this were removed from the dataset leaving 86 datasets. Many players also died within the first few minutes of playing, giving a playtest with a low time and exploration amount, the 34 instances of this were also removed from the dataset. Removing these left 52 datasets; one per player. There is a potential issue here where the player may not explore and area again if they have already explored it before dying. However, data from play-throughs where the player died in under 5 minutes would likely have had a more significant effect.

5 ANALYSIS AND INTERPRETATION

Table 2 shows the correlations and P-Values from the T-Test results calculated using the statistical software R. We consider a P-Value of 0.05 or less (i.e. 95% confidence) to be statistically significant; these results are highlighted in the table.

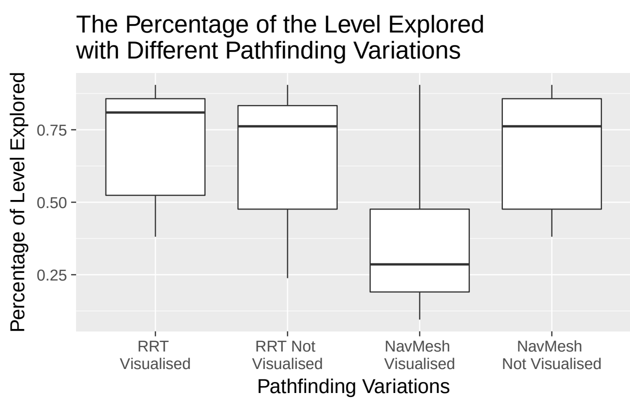

Figure 2: Box and Whisker plot of the percent of the level explored with the 4 game variations.

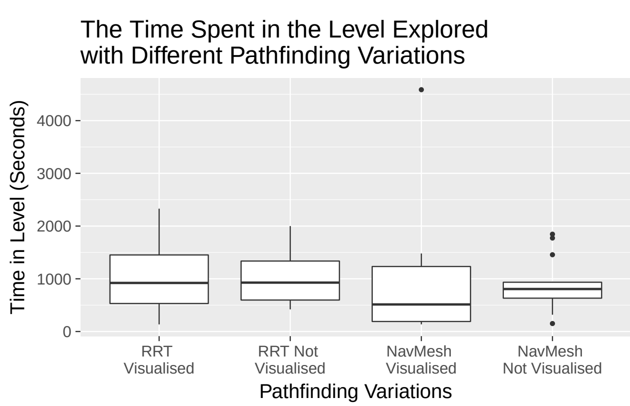

Figure 3: Box and Whisker plot of the time spent in the level with the 4 game variations.

5.1 RRT and the Percent of the Level Explored

The first significant result is a small positive correlation of 0.2807 between the percent of the level explored and the use of RRT pathfinding. This suggests that players explore more in the RRT variations.

The graph in Figure 2 shows that the percent explored between the two RRT variations are both close, with the 'not visualised' variation being slightly lower.

There is no significant correlation between the RRT being visualised and the percentage which would suggest that visualising the RRT does not affect the percentage of the level explored. However, when just looking at visualised pathfinding there is another significant correlation between the type of visualised pathfinding used and the percent of the level explored. There is a strong positive correlation of 0.55 which again suggests that players explore more when RRT is used in comparison to NavMeshes. Another RRT related significant correlation is between non-visualised RRT and the percent explored which has a positive correlation of 0.281.

There is no significant correlation between RRT use and the time spent in the level. This suggests while players explore more when the RRT is used they do not spend more time in the level.

Some participants commented that they thought the RRT visualisation was leading somewhere and followed it through the level or went to it when they saw it. However, this only accounts for when the RRT is visualised. Another possibility is that the RRT was spawned in a square and often did not fill a room which may encourage players to move around the edge of the room rather than down the middle. This may have made it easier to find hidden doors increasing their exploration percentage.

Another reason for this could be that in the playtesting software each RRT was limited to a single room as enemies could not follow players out of rooms this may have allowed them to avoid combat, or make combat faster, allowing them to explore more.

5.2 NavMesh Visualisation and the Percent of the Level Explored

In contrast to RRT having a positive correlation with the percent of the level explored, the visualisation of NavMeshes has a negative correlation of -0.4894 with the percent explored. This suggests that players explored less when the NavMesh was visualised. As many playtesters were game development students using Unity they may have known that the visualisation was of the NavMesh. This may have helped them find a quicker route through the level or lead them to avoid rooms with enemies in. Another reason could be that seeing the pathfinding showed them the possible pathways aiding them in finding a more direct path through the level.

5.3 Visualised Path-Finding and Player Deaths

Another significant correlation is the positive correlation between the visualised pathfinding and the number of player deaths. This means that players with the variations with visualised pathfinding died more than players without it. One reason for this could be the player being distracted by the visualisation and therefore did not focus on enemies.

In the RRT variation, the tree visuals are only in areas where there are enemies. As some players reported moving towards the RRT whenever they saw it, they could have been more likely to find enemies than players without the visualisation.

Of the 52 playtesters, 21 did not die and a further 21 died once. The one death could also be due to the player not knowing how

Table 2: T-Test and Correlation Results

Independent Variable	Dependent Variable	Correlation	P Value	T Value	Degrees of Freedom
Visualising Pathfinding	Percent Explored	-0.1766742	0.2009	1.2963	49.335
Visualised Pathfinding Method	Percent Explored	0.5500408	0.003752	-3.2666	20.61
RRT Used	Percent Explored	0.2807048	0.04425	-2.068	46.315
Non-Visualised RRT Use	Percent Explored	0.2807048	0.04425	-2.068	46.315
Visualised RRT	Percent Explored	0.1697141	0.4229	-0.81894	19.146
Visualised NavMesh	Percent Explored	-0.4893995	0.01176	2.7493	21.813
Visualised Pathfinding	Time	0.0002296794	0.9987	-0.0016952	42.752
RRT Used	Time	0.06559258	0.6445	-0.46481	42.18
Visualised RRT	Time	-0.0153404	0.9384	0.078154	23.869
Visualised NavMesh	Time	0.001491802	0.9943	-0.0073083	16.307
RRT Used	Deaths	-0.2656382	0.05917	1.9483	36.148
Visualised Pathfinding	Deaths	0.318452	0.01833	-2.4424	47.973

to play the game which may have skewed the results. A smaller number of players (8 and 2 respectively) died 2 or 4 times.

Figure 3 shows that the time spent in the level for all four variations were close. However, unlike figure 2 there are a number of outliers on the NavMesh variations. Firstly, on visualised NavMeshes, there is an outlier with over 4000 seconds spent in the level where a majority of the other playtests for all four variations were under 2000. Finding this outlier in the playtest data showed that this player has a very high play time with a low exploration percent. As this was a remote player this suggests that the player just launched the level and left the game running without playing, although the player did die so they may have made some attempt to play.

6 DISCUSSION

While the methodology of the study is appropriate in this case, there were many areas that could be improved in future work to improve the quality of the data collected. This section identifies a number of weaknesses in this study and how they could be addressed in future work.

One issue with the playtesting software is that the game is not specifically designed for use in this paper. While the game is a Metroidvania game with a focus on exploration a few participants noted the enemies in the game were very aggressive and a majority of players died at least once which could have affected their desire to explore. In a future study, a game would either be designed specifically for the playtesting or altered more to suit the study.

Another issue with using the game 'Gates of Amenti' is that many of the playtesters were game development students who may have playtested other levels of the game or seen the game trailer during presentations in class. This means that they could have been looking for things they had previously seen in the trailer as opposed to exploring or being affected by the visualisation. As a majority of them were students studying a game development course they playtested the game looking for bugs and issues instead of just playing the game which again may have affected the results. Future work would use a wider sample of demographics, probably still focussing on gamers but less so on those familiar with game development.

The RRT variations of the game were more resource intensive to run than the NavMesh versions. This lead to a frame rate drop at the start of the game in the RRT variations. Many of the playtests used university computers which were capable of running both versions well. However, some playtests were done remotely and therefore the computer specification and performance of the game are unknown. In future studies, this would be addressed by either doing all the playtests in one location ensuring everyone uses similar computers or to expand on the game's data collection to include the frame rate and/or the player's computer specification.

The visualisation itself is also potentially an issue as many players did not know what it was or they assumed it was part of the game's aesthetic and ignored it. (One participant described the visualisation of the RRT pathfinding as "sci-fi brambles".) However, the visualisation could still affect the player even if they do not understand it. Many participants suggested they do not understand the RRT visualisation but the results still showed their exploration was affected. Adding some explanation of the visualisation may also have an effect.

While the RRT visualisation worked as intended, in future work an area to explore would be increasing the RRT size. In this experiment, the RRT was resource intensive so in future work the RRT could be optimised more and then the RRT could be increased in size and clustering could be used to filter the tree.

The data collected from the game was usable but it could be improved to get more data that would allow for more in-depth analysis. For example, currently the game only records how many doors the player walks through, which is a fairly coarse-grained measure of exploration. An improvement on this would be to record what doors they walked through to create heat maps and perform more statistical analysis to see whether players explore differently in the different game variations or if the players died in similar locations.

There is a potential issue of reproducability, in the sense that two players are very unlikely to see exactly the same visualised paths. RRT randomly explores the environment and so is nondeterministic by nature, and the results of both pathfinding algorithms can be sensitive to initial conditions such as precise initial placement of player and NPCs, and the movement of physics-based obstacles. However as both versions are affected by this we do not see it as a major factor in this study.

7 FUTURE WORK

The most unexpected significant result found in Section 5 is the NavMesh visualisation decreasing the percent of the level explored. Future research on this subject could research that in more detail to find a more definite reason for why that happened.

Also, as Section 6 mentions, there were issues with the game used in this study, as future research would require a different or more complete game it could also give interesting data to look at pathfinding in more complex environments and other game genres such as stealth games.

Another area that could be explored in future work is the RRT visualisation and variations of it. As the use of RRT in this study had numerous small RRTs with less than 500 nodes each, another area to research could be comparing having an RRT per room to an RRT per level or varying the number of nodes in the RRT. Also for larger RRT variations clustering methods, as discussed in Section 2.4, could be used to provide clearer and more legible visualisation.

8 CONCLUSION

While the use of RRT is more prevalent in robotics than digital games its use was feasible in this paper. Its past use in game design tools and the results gathered here show that it can be used in games. Previous studies have looked at wayfinding and player exploration and suggest that environmental factors in digital environments do have some effect on player exploration. Similarly, this study found player exploration changed between games variations.

This study found a number of significant results which suggest that the pathfinding method and visualisation do affect player exploration in a game level. Like many of the papers reviewed in Section 2 this could be applied in game design. Level designers could take it into account what type of pathfinding will give the result they want as different results may be wanted for different genres or even different sections of the game.

We have shown that visualisation of NPC pathfinding has potentially significant, though not always intuitive, effects on player behaviour. Even when not visualised, a decision such as which pathfinding algorithm to use can still change the gameplay experience. This suggests that these considerations are worth bearing in mind for game designers, particularly when seeking to encourage the player to explore the game environment.

ACKNOWLEDGEMENTS

The first author would like to thank Bears are OP for allowing the use of their game in playtesting, and all the playtesters who took part in the study. The authors would also like to thank the third year BSc Computing for Games students at Falmouth University, whose peer reviews and support were greatly appreciated throughout this project.

REFERENCES

[1] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, 1968.
[2] Z. A. Algfoor, M. S. Sunar, and H. Kolivand, "A comprehensive study on pathfinding techniques for robotics and video games," *Int. J. Comput. Games Technol.*, vol. 2015, pp. 7:7–7:7, Jan. 2015.
[3] A. Nash, K. Daniel, S. Koenig, and A. Feiner, "Theta*: Any-angle path planning on grids," in *Proceedings of the 22Nd National Conference on Artificial Intelligence - Volume 2*, ser. AAAI'07. AAAI Press, 2007, pp. 1177–1183.
[4] E. R. Firmansyah, S. U. Masruroh, and F. Fahrianto, "Comparative analysis of a* and basic theta* algorithm in android-based pathfinding games," in *2016 6th International Conference on Information and Communication Technology for The Muslim World (ICT4M)*, 2016.
[5] S. M. LaValle, "Rapidly-exploring random trees: A new tool for path planning," Computer Science Dept., Iowa State University, Tech. Rep., 1998.
[6] J. J. Kuffner and S. M. LaValle, "RRT-connect: An efficient approach to single-query path planning," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, vol. 2, 2000.
[7] S. Karaman, M. R. Walter, A. Perez, E. Frazzoli, and S. Teller, "Anytime motion planning using the RRT*," in *2011 IEEE International Conference on Robotics and Automation*, May 2011, pp. 1478–1483.
[8] S. Karaman, "Incremental sampling-based algorithms for optimal motion planning," *Robotics Science and Systems VI*, vol. 104, 2010.
[9] J. Tremblay, A. Borodovski, and C. Verbrugge, "I can jump! exploring search algorithms for simulating platformer players," in *Tenth Artificial Intelligence and Interactive Digital Entertainment Conference*, 2014.
[10] M. Treanor, A. Zook, M. P. Eladhari, J. Togelius, G. Smith, M. Cook, T. Thompson, B. Magerko, J. Levine, and A. Smith, "AI-based game design patterns," in *In Proceedings of the 10 International Conference on Foundations of Digital Games, FDG 2015. Society for the Advancement of the Science of Digital Games*, 2015, pp. 5–6.
[11] "Alien Isolation," Steam, Creative Assembly, 2014.
[12] D. Isla, "Third Eye Crime: building a stealth game around occupancy maps," in *Proceedings of the Ninth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, ser. AIIDE'13. AAAI Press, 2014, pp. 206–206.
[13] "Third Eye Crime," Steam, Moonshot Games, 2014.
[14] R. Haworth, S. S. T. Bostani, and K. Sedig, "Visualizing decision trees in games to support children's analytic reasoning: Any negative effects on gameplay?" *Int. J. Comput. Games Technol.*, vol. 2010, pp. 3:1–3:11, Jan. 2010.
[15] A. W. Bauer and Z. Popovic, "RRT-based game level analysis, visualization, and visual refinement." in *AIIDE*, 2012.
[16] S. M. Van Dongen, "Graph clustering by flow simulation," Ph.D. dissertation, Utrecht University, 2001.
[17] M. R. F. Mendona, H. S. Bernardino, and R. F. Neto, "Stealthy path planning using navigation meshes," in *2015 Brazilian Conference on Intelligent Systems (BRACIS)*, Nov 2015, pp. 31–36.
[18] J. Tremblay, P. A. Torres, N. Rikovitch, and C. Verbrugge, "An exploration tool for predicting stealthy behaviour," *IDP*, vol. 13, 2013.
[19] C. Si, Y. Pisan, C. T. Tan, and S. Shen, "An initial understanding of how game users explore virtual environments," *Entertainment Computing*, vol. 19, pp. 13–27, 2017.
[20] F. Bacim, A. Trombetta, R. Rieder, and M. Pinho, "Poster: Evaluation of wayfinding aid techniques in multi-level virtual environments," in *2008 IEEE Symposium on 3D User Interfaces*, March 2008, pp. 143–144.
[21] D. Moura and L. Bartram, "Investigating players' responses to wayfinding cues in 3d video games," in *Proceedings of the extended abstracts of the 32nd annual ACM conference on Human factors in computing systems*. ACM, 2014, pp. 1513–1518.
[22] C. Nutt, "The undying allure of the metroidvania," [Online]. Available: http://www.gamasutra.com/view/news/236410, [Accessed: 01-May-2018].