How Does Visualising Path-finding in an NPC Effect How Participants Explore a Level?

1507866

***Abstract*—The abstract goes here.**

I. Introduction

THE research questions proposed in this project are: how does visualising *Rapidly Exploring Random Trees* (RRT) path finding in a *Non Player Character* (NPC) affect how a player explores a game level?

This project will look at visualising an enemy NPC's path-finding using RRT based path-finding. Figure 1 shows an example of RRT path-finding where the RRT explores the area and then a path is drawn along the tree [1]. The visualisation will occur around an enemy NPC allowing the participant to see where the enemy is going in a level of a game made in Unity [2]. Logging tools in the play testing software will record how long a player spends in a level and what percent of the level they explore. This data will be used to see if the visualisation has any effect on how the participant explores the level of the given game.

Previous papers have researched visualising *Artifical Intelligence* (AI) and foregrounding AI. However, there is little on on what effect this has on how the participants play the game.

A. Hypothesis:

Null Hypothesis: Visualising RRT path-finding has no effect on the percent of the level the participant explores.
Hypothesis 1: Visualising RRT significantly effects the percent of the level the participant explores.
Hypothesis 2: Decreasing the size of rooms decreases the percent of the level explored.
Hypothesis 3: Varying size of RRT visualisation effects? Percent of level explored and/or time spent in level??

II. Related Work

A. Foregrounding and Visualising AI

AI is used in most modern digital games. However, it is rarely foregrounded or visualized in those games. Treanor *et al* say that the AI in games if often designed to fit the game. Therefore, this AI is supporting the game play rather than being central to it [3], [4].

Treanor *et al* surveyed many games that foreground AI in different ways. From this they proposed a series of design patterns for foregrounding AI in digital games. The two design patterns relevant to this project are firstly "AI as a Villain". They describe this pattern as having the AI not try to outright defeat the player. Instead it's designed to create an experience like Alien Isolation [5], [3]. In Alien Isolation the AI hunts the player. This foregrounded appears here as the player has to observe the AI and learn how to avoid.

This paper will also use AI as a villain as enemy NPC's will have their path-finding visualized around them. The participants will have to observe this to learn how to not get caught by the enemy NPC.

The second relevant design pattern is "AI is Visualized". This is where there is a visual representation of the AI's state or decision making in the game.

Most games hide this from the player but this design pattern visualizes it making it mechanic. The example given by Treanor *et al* is the game Third Eye Crime. Third Eye Crime is a game that followed the "AI is Visualised" design pattern [6], [7]. The game uses probabilistic object tracking through Occupancy maps. The game uses Occupancy Maps to display where the enemy thinks the player could be in the map, as the enemy moves around the map it removes areas where the player is not from the Occupancy Map [6]. 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 [6] [7].

This pattern is relevant to this project as the enemy NPC will have RRT path-finding visualized around it. Allowing the player to see where the enemy is going and decide how to overcome or outsmart it.

While Haworth *et al* do not visualise an AI process they do visualise the possible decision in a game on a tree structure [8]. They research visualising decision trees in a game to see what effect it had on children's analytical reasoning and game play. While they did not come to any definite conclusions their results suggested that data aided players in playing the game as in later level the children struggled to beat the game without the visualised tree. However, an issue they noted was that the game could be unbalanced at the end making the usefulness of the tree being questionable.

A further issue is that Haworth *et al* only tested the tree in a relatively simple 2D game that was tested on children. This does not give any data on 3D games on the market??? In contrast, Isla's visualised path-finding in Third Eye Crime is on sale?? (Word it better) [6].

Like Haworth *et al*, Bauer *et al* also research visualising tree structures [9]. However, they did use an AI technique, they used Rapidly-Exploring Random Trees (RRT).

B. Pathfinding

In digital games the A* path finding algorithm appears to be the most widely used [10]. Algfoor *et al* surveyed numerous papers on path finding. The focus appeared to be on the type of grids used in path-finding and then numerous algorithms that can be used [10]. The most popular being the A* algorithm for use in digital games and robotics. RRT path-finding was not mentioned. They surveyed many grid types and gave the advantages of each. However, RRT does not use grids it instead uses nodes making the grid type irrelevant.

Hu *et al* propose an implementation of A* path-finding in Unity, the engine being used by this project [11]. While their implementation is in an older version Unity the implementation in Unity 5.6 should still be similar.

Third Eye Crime was previously mentioned for it's foregrounding of AI. However, it also uses visualisation as an important mechanic [6] [7]. Isla uses Occupancy Maps to show where the enemy NPC thinks the player could be. Occupancy or Influence maps do not produce a path instead they show the probability of the player being in different locations across the map [6], [12]. Isla used Occupancy maps to show where the enemy AI thinks the player currently is. He then used .... for the NPC to navigate to the most likely location of the player. Similarily, Miles and Loius used also used influence maps. While their example is specific to *Real Time Stratedgy* (RTS) games like Isla they used Occupancy maps as a base to for A* path-finding instead of A* using the map itself for path-finding [12].

A further paper on path finding is Wang and Lu's paper which looks at path finding in a 3D environment. While again they were using A* they look at using A* in 3D and suggest using nodes instead of a grid?? [13].

Mendona *et al* look at path-finding both in robotics and digital games [14]. Their focus is on stealth path-finding in games and applying that to robotics. Like RRT, the methods they propose does not necessarily find the optimal path [15] [14]. They tried to find the most stealthy path. They generated custom navigation meshes (navmeshes) and assigned a weight to each polygon in the navmesh depending on how close it is to cover.

As Mendona *et al* as use path-finding to find a stealthy path the path with the optimal distance may not always be the path with the lowest cost in relation to the robot or AI agent staying cover. Therefore, the optimal path may not always be required.

This project will favour interesting visualisations over the optimal path in the variant that uses

'point is optimal not always necessary M

navmeshes [16] [14]

Rapidly-Exploring Random Trees (RRT) are a search method used more widely in robotics than digital games [17] [1]. Kuffner and LaValle first proposed RRT in 2000, they intended to produce a random algorithm more efficient than the other search algorithms available at the time. Figure 1 shows Kuffner and LaValle's RRT Path Planner a variant of RRT that can be used to find paths from the generated tree [1].

RRT's goal is to find a path between two point with no collisions, the path found may not the the optimal path though [1], [18]. Karaman and Sertac say that the chance of

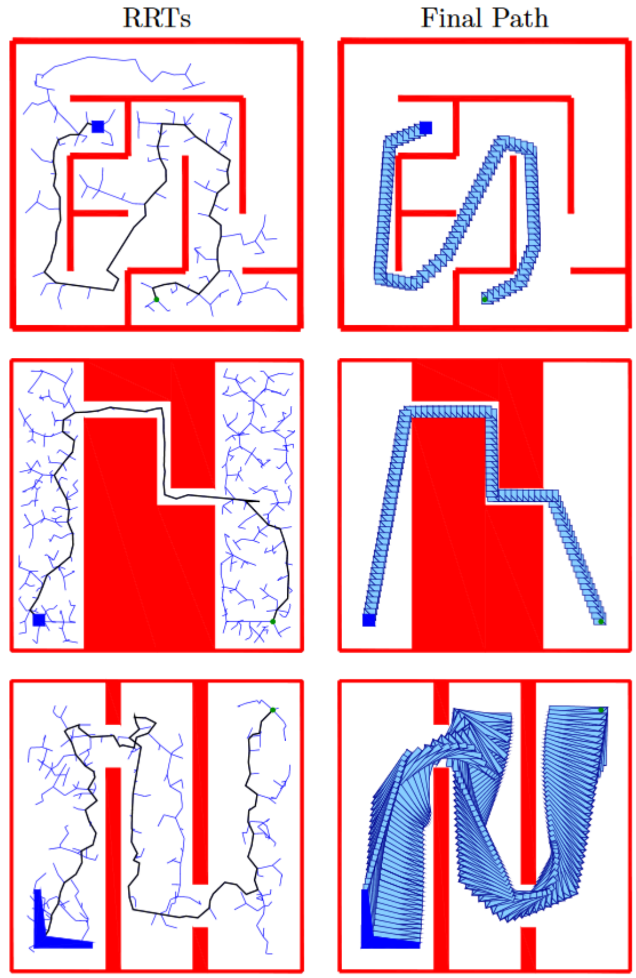

Fig. 1. Kuffner and LaValle's [1].

RRT finding an optimal path is very unlikely [15]. Karaman *et al* propose another version of RRT called RRT* this version

Bauer and Popovic used RRT for level design in games [9]. Like Haworth *et al* they visualised the data to aid users [9] [8]. However unlike Haworth *et al* the visualisation is for game developers not the players.

Their focus is on level design not game-play. They designed a tool that could analyse a level generated by PCG or a level designer. They then use RRT to calculate possible routes the player could take when playing. This produced an image that was difficult to read so Dongen's method is used for graph clustering to make the output more legible [9], [19]. This project is focused on using the visualisation in the game to the game design. However, a similar technique will be required to organise the RRT output to make it understandable to the player in way that they can look at the visualisation and interpret what the NPC is going to do.

Tremblay *et al*, like Bauer and Popovic, also use RRT visualisations to aid level design [20] [9]. Like Mendona *et al*, they focused on designing stealth games and finding stealth orientated paths in the game levels [14]. Similar to Bauer and Popovic, they used RRT to visulaise possible moves the player could make, then they used clustering to make the results less

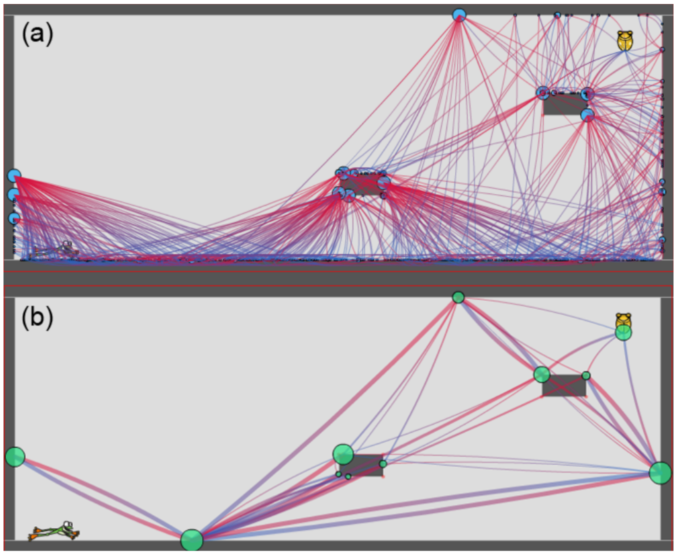

Fig. 2. Bauer *et al*'s graph-based representation of RRT with and without clustering [9].

cumbersome [20].

RRT was used in this case as it's flexible, inexpensive and it's random nature mirrored a wider variety of player behaviours [20]. Similarly this project will use RRT not for reflecting player behaviour but instead to try and create a visualisation that fits with the game its put and and that is interesting for the player to interact with. A path that is interesting to play with is more important in this case than an optimal path. To see how this influences the player's experience it will also be compared to a visualisation of the built in Unity navmesh system.

While the use if RRT to find a stealthy path is different to RRT's application in this project a potential problem is that Tremblay *et al*'s results showed that the chances of their RRT implementation finding a path decreased as the grid size increased and as the number of attempts decreased as can be seen in Figure 3.

Tremblay *et al* again look at the use of search algorithms for use in game development [21]. They looked at the use of A*, *Mote Carlo Tree Search* (MCTS) and RRT to visualise player behaviour in platform games. Like

A* and optimal paths vs RRT better visualisation

C. Exploring Game Environments

One method of guiding players through games is to use wayfinding. Wayfinding in games is often visual cues in the environment that will guide the player to an area of interest [22], [23]. While the intention of visualising the RRT path-finding is not to guide players through a level but one of the behaviours being observed in participants is whether they navigate or explore differently with the presence of visible path-finding.

Moura and Bartram investigated the effects different wayfinding cues have on players [24]. They looked at methods used in AAA games and mimicked them in their own game. Their results showed that the absence of wayfinding cues was

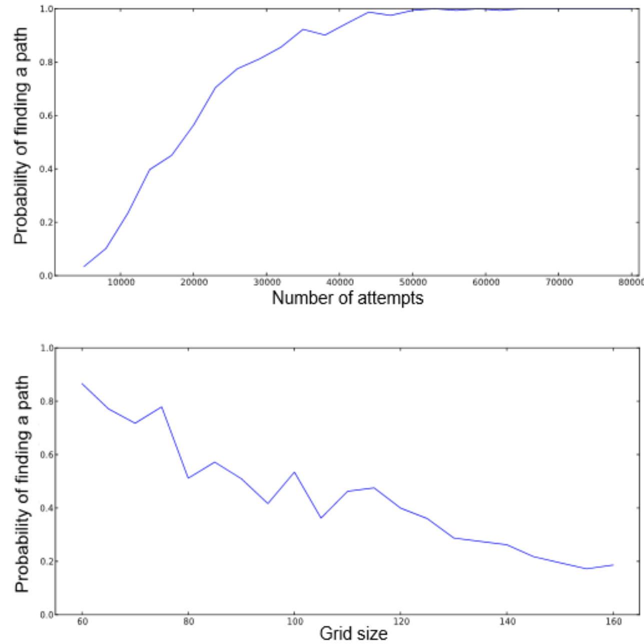

Fig. 3. Performance analysis of Tremblay *et al*'s RRT when running on a Metal Gear Solid level [20].

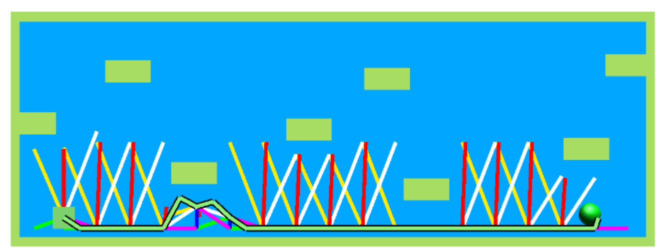

Fig. 4. CAPTION [21].

obvious to players. In contrast the version with wayfinding cues did not have enough cues to sufficiently guide the player. These results suggest that wayfinding cues alone may not be enough to guide the player. While the visualisation of RRT pathfinding in this project will be used as an enemy another application could be in wayfinding. Either as an enemy or friendly NPC the visualisation of path-finding could be used to aid the player. However, Moura and Bartram concluded that results were inconclusive and that more research is required.

Si *et al* investigated how players explore virtual environments [22]. While their experiments were specific to Real Time Strategy (RTS) games the results may apply to other game types.

TODO Link to Haworth visualisation sort of aided exploration of their maze game

Moura *et al* investigate way-finding cues in triple A games [24]. They want to see what effects these cues had on player behaviour and whether that behaviour could be classified. While their results were inconclusive they found that cues alone were not enough to guide the player.

III. Methodology

The methodology that will be used to test the given hypothesis will be play testing and questionnaires. This will require human participants to play the game and fill in the questionnaires.

The game being used us a 3D metroidvania game which has a focus on exploration. Players will be given one variation of the game to play and then asked to complete a questionnaire on their experience. The game will also log the players position ..... the game will also record how long a player spends in a level.

A. Playtest Variations

There will be multiple variations of the game to test the different hypothesis.

The first variation will be the control version. This version of the game will have no visualisation on the enemy NPC's path-finding.

The second variation will use RRT path-finding and will have the tree visualised around it.

The third variation use A* path-finding and be visualised on the level floor.

A-B testing will used on participants. Participants will be assigned different version of the game to play and then the results will be compared.

While the participant plays their current location will be exported to a .CSV file every second. Varying the export rate is unlikely to provide data of interest as the focus is on time spent in a level and how much was explored.

R will then be used to analyse the exported data. Firstly to generate heat maps to see if there are any noticeable patterns as Wallner says heat map are useful as they are easy to generate and easy to discern patterns from [25]. Other statistical analysis such as bar charts will also be generated to support the heat maps.

There will also be a questionnaire for the participants to fill out after completing the play test.

TODO Add refs

B. Questionnaire

The questionnaire will be completed using an online questionnaire such as Google Forms. The participants will have to answers questions using Likert scales.

Alongside play testing participants will also be asked to fill out a questionnaire on the game. Nordin *et al* say that questionnaires are vital for understanding how players feel when playing digital games [26]. Questionnaires are also beneficial as they can prompt players to give answers they may not have given spontaneously. There are a wide variety of questionnaires that exist to measure players experiences in play testing. However, Nordin *et al* say there are many issues with using existing questionnaires. Firstly, many of these questionnaires are not readily available. Another issue is that not all questionnaires have been thoroughly checked for validity. Finally, there is the potential issue of a researcher not fully understanding the questions put forward by another researcher or may not understand what data that question is intended to get.

TODO Add refs + more explanation

IV. Conclusion

The conclusion goes here.

References

[1] 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, pp. 995–1001 vol.2.

[2] Unity Technologies, "Unity," [Online]. Available: https://unity3d.com/, [Accessed: 11-Nov-2017].

[3] 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," 2015.

[4] M. P. Eladhari, A. Sullivan, G. Smith, and J. McCoy, "Ai-based game design: Enabling new playable experiences," 2011.

[5] "Alien Isolation," Steam, Creative Assembly, 2014.

[6] 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.

[7] "Third Eye Crime," Steam, Moonshot Games, 2014.

[8] 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.

[9] A. W. Bauer and Z. Popovic, "RRT-based game level analysis, visualization, and visual refinement." in *AIIDE*, 2012.

[10] 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.

[11] J. Hu, W. gen Wan, and X. Yu, "A pathfinding algorithm in real-time strategy game based on unity3d," in *2012 International Conference on Audio, Language and Image Processing*, July 2012, pp. 1159–1162.

[12] C. Miles and S. J. Louis, "Towards the co-evolution of influence map tree based strategy game players," in *2006 IEEE Symposium on Computational Intelligence and Games*, May 2006, pp. 75–82.

[13] M. Wang and H. Lu, "Research on algorithm of intelligent 3d path finding in game development," in *Industrial Control and Electronics Engineering (ICICEE), 2012 International Conference on*. IEEE, 2012, pp. 1738–1742.

[14] 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.

[15] S. Karaman, "Incremental sampling-based algorithms for optimal motion planning," *Robotics Science and Systems VI*, vol. 104.

[16] D. H. Hale, G. M. Youngblood, and P. N. Dixit, "Automatically-generated convex region decomposition for real-time spatial agent navigation in virtual worlds," in *Proceedings of the Fourth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, ser. AIIDE'08, 2008, pp. 173–178.

[17] S. M. LaValle, "Rapidly-exploring random trees: A new tool for path planning," [Online]. Available: http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.35.1853, note = [Accessed: 1-November-2017].

[18] 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.

[19] S. M. Van Dongen, "Graph clustering by flow simulation," Ph.D. dissertation, 2001.

[20] J. Tremblay, P. A. Torres, N. Rikovitch, and C. Verbrugge, "An exploration tool for predicting stealthy behaviour," *IDP*, vol. 13, 2013.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] G. Wallner and S. Kriglstein, "Game research methods," P. Lankoski and S. Björk, Eds. Pittsburgh, PA, USA: ETC Press, 2015, ch. An Introduction to Gameplay Data Visualization, pp. 231–250.

[26] A. I. Nordin, A. Denisova, and P. Cairns, "Too many questionnaires: measuring player experience whilst playing digital games," in *Seventh York Doctoral Symposium on Computer Science & Electronics*, vol. 69, 2014.