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

1507866

Abstract—**The abstract goes here.**

I. INTRODUCTION

THE research question proposed in this project is: 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 pathfinding using RRT path-finding.

Figure 1 shows an example of RRT path-finding. Here the RRT explores the area and then draws a path along the tree [1]. RRT involves nodes being randomly placed. Then it chooses a parent node based on whichever pre-existing node it closest to it.

This project will look at visualising the tree produced by RRT and the process taken to produce it. The visualisation will be around an enemy NPCs. This will allow the participants to see where the enemy NPC is going and what their line of sight is.

Implementation of the visualisation will be in a level of a 3D game made in Unity 5.6 [2]. Logging tools in the play testing software will record the amount of time spent in a level and what percent of the level they explore. Analysis of this data will then determine whether the visualisation has any affect on participant exploration.

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 affect on the percent of the level the participant explores.
Hypothesis 1: Visualising RRT significantly affects the percent of the level the participant explores.
Hypothesis 2: Visualising RRT significantly affects the length of time the participant spends in the level.
Hypothesis 3: Visualising Unity navmeshes affects t the percent of the level the participant explores compared to RRT.

II. RELATED WORK

A. Foregrounding and Visualising AI

Most modern digital game make use of AI. However, it is rarely foregrounded or visualised in those games. Treanor *et al* say that often the design of AI in games is to fit the game and complement game play. These AI are supporting the game play rather than being central to it [3] [4].

Treanor *et al* surveyed 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 project are "AI as a Villain" and "AI is Visualized" . They describe the first pattern as having the AI not try to outright defeat the player. Instead it's designed to create an experience like in the game Alien Isolation [5], [3]. In Alien Isolation the AI hunts the player. This is foregrounding as the player has to observe the AI and learn how to avoid. There is also some visualisation as the player has a scanner that will inform them of the AI's position.

This paper will use "AI as a villain" pattern 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 decision making process they do visualise the possible decisions available to the player 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. Their study did not come to any definite conclusions. However, their results suggested that the trees aided players. 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 later levels. This makes the usefulness of the tree being questionable in this example.

A potential issue with this study is that Haworth *et al* only tested the tree on a simple 2D game for school children. Of these school children only a few had experienced playing digital games before. This may mean that this data may not be relevant for 3D games or games available to buy on the market. In contrast, Isla's visualised Occupancy Maps are in the game Third Eye Crime which is available on Steam, IOS and Android [6] [7].

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).

Another use of visualisation is in game design. Often to check player behaviour in player testing or to aid the design of levels [10] [9] [11] [12].

B. Pathfinding

In digital games the A* path-finding algorithm appears to be the most used [13]. Algfoor *et al* surveyed numerous papers on path finding.

Their focus was on the use of different grids in path-finding and the numerous algorithms available [13]. The most popular being the A* algorithm for use in both digital games and robotics. RRT path-finding was not mentioned here. 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 the Unity engine, the engine used in this project [14]. While their implementation is in an older version Unity the implementation in Unity 5.6 should still be similar.

The previous section mentioned Third Eye Crime for it's foregrounding of AI. 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], [15]. Isla used Occupancy maps to show where the enemy AI thinks the player currently is. The enemy then moves to investigate that area reducing the probability of the player being there to 0. Similarly, Miles and Loius used also used influence maps. While their example is specific to *Real Time Strategy* (RTS) games, like Isla they used Occupancy maps. They used them as a base to for A* path-finding instead of A* using the map itself for path-finding [15].

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 [16].

Mendona *et al* look at path-finding both in robotics and digital games [17]. 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 [18] [17]. Instead they try to find the most stealthy path. They generate custom navigation meshes (navmeshes). Then they assign a weight to each polygon in the navmesh depending on how close it is to being behind cover.

As Mendona *et al* 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 AI agent being in cover. Therefore, a requirement might not always be the optimal path.

This project will favour interesting visualisations over the optimal path in the variants that use RRT.

Rapidly-Exploring Random Trees (RRT) are a search method used more in robotics than digital games [20] [1]. Kuffner and LaValle first proposed RRT. 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. Path Planner is a variant of RRT which has the intended use of finding paths from a generated tree [1].
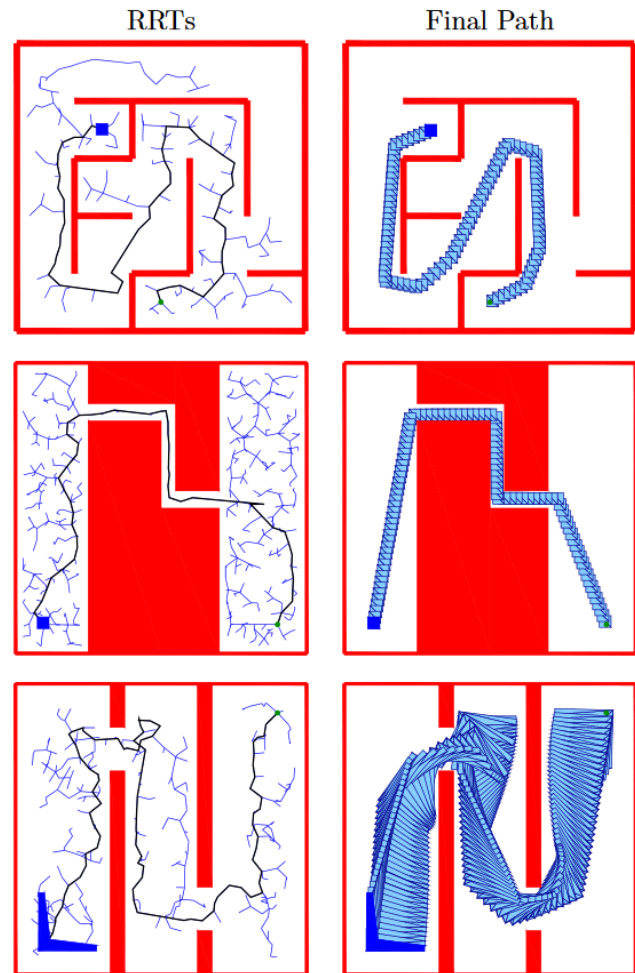

Fig. 1. Kuffner and LaValle's [1].

RRT's goal is to find a path between two point with no collisions. The path found may not be the optimal path though [1], [21]. Karaman and Sertac say that the chance of RRT finding an optimal path is very unlikely [18] [12]. Karaman *et al* propose a variant of RRT called RRT*. RRT* starts the same as RRT. However, when a new node has to choose a parent node instead of choosing the nearest node it evaluates the cost of the nodes in regards to reaching it's goal. Each iteration involves re-evaluating parent nodes to reduce

the cost. Rewiring of the tree happens when a lower cost path is located.

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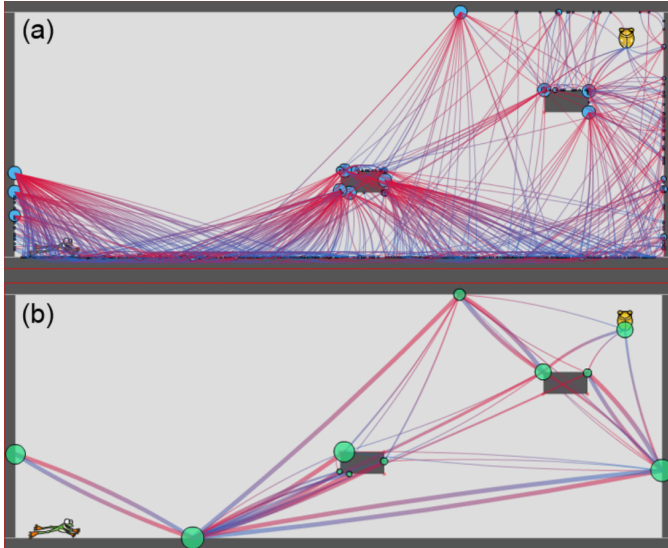

Fig. 2. Bauer *et al*'s graph-based representation of RRT with and without clustering [9].

Their focus is on level design not game-play. They propose a tool that analyses a level generated by PCG or a level designer. They then use RRT to calculate possible routes the player could take when playing. This produces an image that is difficult to read. The use of Dongen's method for graph clustering makes the output more legible [9], [22].

The focus of this project is on a similar type of visualisation but in a game instead of game design. Therefore, there is a need for a similar technique to organise the output of RRT. This should make the visualisation more useful as it should aid 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 [11] [9].

Like Mendona *et al*, they focus on designing stealth games and finding stealth orientated paths in the game levels [17] [11]. Like Bauer and Popovic, they use RRT to visualise possible moves the player could make. Then they use clustering to make the results less cumbersome to the user [11].

The use of RRT in this case is because it is flexible and inexpensive. Also it's random nature mirrors a wide range of player behaviours [11]. This project will use RRT but not for reflecting player behaviour. Instead it's use is for creating a visualisation that fits with the game and that is interesting for the player to interact with. A path that is interesting to play with is more important in this project than an optimal path. There will then be a comparison between a visualisation of the Unity navmeshes against the RRT visualisation.

While the use of RRT to find a stealthy path is different to it's use 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. It also decreased as the number of attempts decreased as shown in Figure 3.
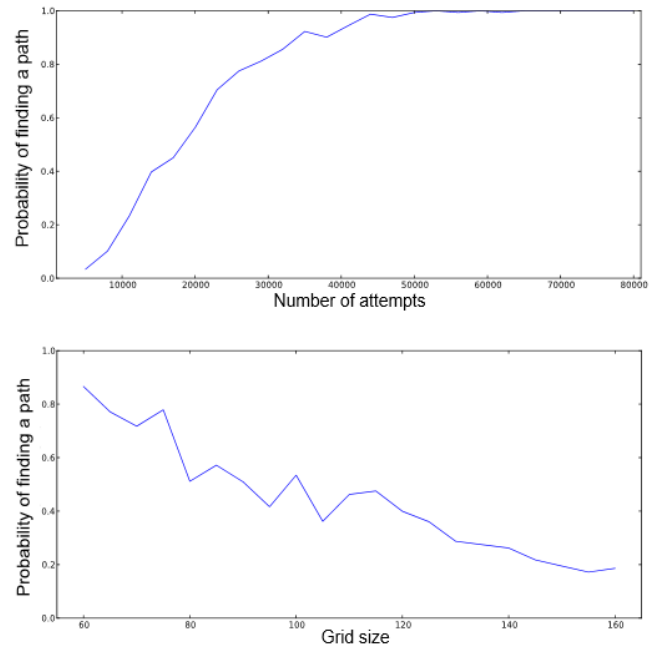

Fig. 3. Performance analysis of Tremblay *et al*'s RRT when running on a Metal Gear Solid level [11] [23].

Tremblay *et al* again look at the use of search algorithms for use in game development. This work builds on the 2013 paper on RRT in stealth games [12]. They look at the use of A*, *Mote Carlo Tree Search* (MCTS) and RRT to visualise player behaviour in platform games similar to Bauer [12] [9] .

RRT is limited to drawing straight lines between the nodes, as straight lines are not always possible in RRT Tremblay *et al* also use either A* or MCTS to find a path to the node and ensure that it is reachable.

C. Exploring Game Environments

One method of guiding players through games is to use way-finding. Way-finding in games is often visual cues in the environment that will guide the player to an area of interest [24], [25].

The intention of visualising RRT path-finding in this project is not to guide the players. However, like Si *et al* it will observe how players navigate and explore levels and whether, like the presence of way-finding cues, it affects player behaviour.

Moura and Bartram investigate the effects of different way-finding cues on players [26]. They looked at methods used in triple A games and mimicked them in their own game. Their results show that the absence of way-finding cues was obvious to players. In contrast, the version with way-finding cues did not have enough cues to sufficiently guide the player. These results suggest that way-finding cues alone may not be enough
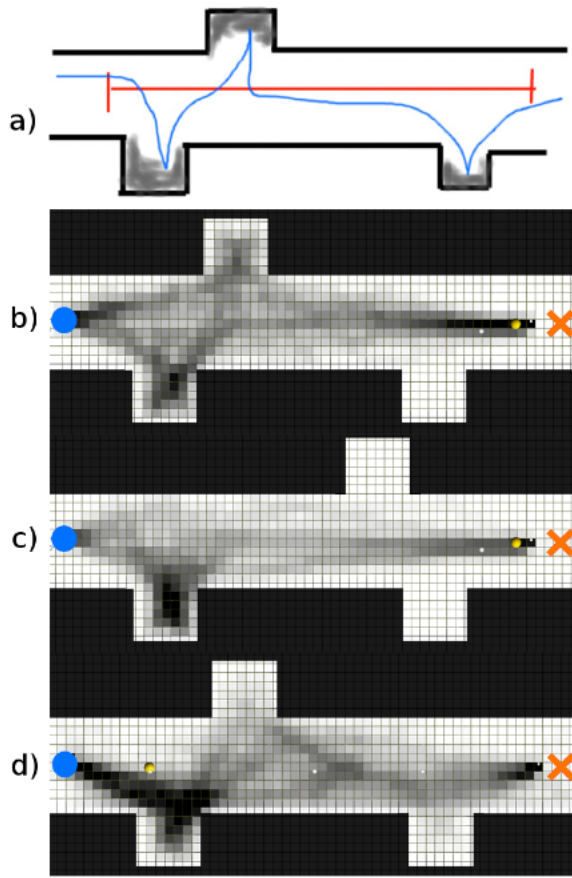
Fig. 4. Heat maps of a single level where the design has been changed each time [11].
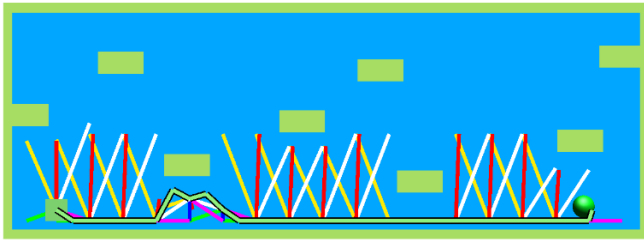

Fig. 5. Tremblay *et al*'s visualisation of an RRT tree [12].

to guide the player. While the use of visualisation of RRT path-finding in this project will be as an enemy another application could be in way-finding. Either as an enemy or friendly NPC the visualisation of path-finding could have a use a an aid to the player. However, Moura and Bartram concluded there is a need for more research as the results were inconclusive.

Si *et al* investigated how players explore virtual environments [24]. While their experiments were specific to Real Time Strategy (RTS) games the results may apply to other game types. Si *et al* say that three common types of spatial exploration are; environment mapping, bonus item collecting and location/landmark discovery. The relevant exploration type for this project is spatial mapping. Firstly as it is what the enemy NPC will be doing. Secondly as it is the player

behaviour that is being measured by the logging tool in the software.

Haworth *et al* did not look at player exploration but the game they used in their study involved a map where participants had to explore a maze. Participants which had the decision tree visualisation found it easier to navigate the maze. This suggests that visualisation could aid the exploration process [8].

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 the enemy NPC will be tested in is a 3D metroidvania game which has a focus on exploration. Players will be given one of the game variations to play and then asked to complete a questionnaire on their experience. The game will also log the players position at regular intervals to calculate what percent of the level they explore. How long a participant spends exploring the level will also be logged.

A. Playtest Variations

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

1) Control: Unity Navmesh: The first variation will be the control version of the game. This version will have no visualisation on the enemy NPC's path-finding. The enemies in the game will use the default Unity navmeshes.

2) Unity Navmesh Visualised: The second variation use Unity's built in navmeshes and navmesh agents for path-finding. This will be visualised on the floor of the level around the enemy NPC.

3) RRT Path-finding Visualised: The third variation will have visualised path-finding. The path-finding will use RRT and the player will be able to see three around the enemy NPCs.

4) RRT path-finding: The final version will also use RRT path-finding but there will no visualisation around the enemy NPC.

A-B testing will used on participants. A-B testing is where each participant will be assigned a version of the game to play [27]. In this case to prevent them from working out what the goal of the play test is.

The software will log the participant's position in the level throughout their play test. This data will be exported in a .CSV 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.

The exported data will then be analysed using R. Wallner says heat map are useful as they are easy to generate and easy to discern patterns from [28]. Tremblay *et al* also used heat maps on their RRT visualisations. They used heat maps as part of their tool for level designers to use to see where player could go and the effects of them altering the level design which can be seen in Figure 4. Here they will only be used for analysis and to assist in finding noticeable patterns.

Other statistical analysis in R, such as bar charts, will also be generated to support the heat maps.

After completing a play test of the game participants will be asked to complete a questionnaire on their experience.

B. Questionnaire

Alongside play testing participants will also be asked to fill out a questionnaire on the game online using Google Forms. Nordin *et al* say that questionnaires are vital for understanding how players feel when playing digital games [29] [30]. They can also help give uniformity and consistency to the data gathered from participants [30].

Questionnaires are also beneficial as they can prompt players to give answers they may not have given spontaneously. The two options here are to either create a questionnaire specifically for the experiment or use an existing one. There are a wide variety of questionnaires that already exist to measure players experiences in play testing [29] [31]. These come with the benefit of being more likely to be thoroughly tested such as the *Immersive Experience Questionnaire* (IEQ) which uses Likert scales responses [29] [31].

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. 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.

As the quantitative data gained from play testing is more important than qualitative data here the questionnaire will be based on the IEQ with added questions related to the visualisation. The IEQ tests for immersion but has questions related to interacting with the game world and causes of frustration in it that may relate to the visualisation of path finding.

IV. CONCLUSION

In conclusion, while the use of RRT's is more frequent in robotics than digital games it's use appears feasible in this project. It's past use in game design tools shows that clustering can make the output understandable to the user. However, the need of optimisations such as RRT* or the use of A* may arise to produce of more optimal path.

Visualisation and foregrounding of AI has been successfully used before again suggest it can used here. Finally previous studies have looked at way-finding and player exploration. These suggest that environmental factors in digital environments do affect player exploration.

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] M. J. Nelson, "Game metrics without players: Strategies for understanding game artifacts," in *Proceedings of the 19th AIIDE Conference on Artificial Intelligence in the Game Design Process*, ser. AIIDE'11-19. AAAI Press, 2011, pp. 14–18.

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

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

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

[19] 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.

[20] 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].

[21] 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.

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

[23] "Metal Gear Solid," Konami, howpublished =.

[24] 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.

[25] 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.

[26] 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.

[27] P. Hynninen and M. Kauppinen, "A/b testing: A promising tool for customer value evaluation," in *2014 IEEE 1st International Workshop on Requirements Engineering and Testing (RET)*, Aug 2014, pp. 16–17.

[28] 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.

[29] 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.

[30] A. Denisova, A. I. Nordin, and P. Cairns, "The convergence of player experience questionnaires," in *Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play*, ser. CHI PLAY '16. New York, NY, USA: ACM, 2016, pp. 33–37.

[31] C. Jennett, A. L. Cox, P. Cairns, S. Dhoparee, A. Epps, T. Tijs, and A. Walton, "Measuring and defining the experience of immersion in games," *Int. J. Hum.-Comput. Stud.*, vol. 66, no. 9, pp. 641–661, Sep. 2008.