Heatmap Weighted A* Algorithm for NPC Pathfinding and Graph Switching

Paul Williamson
School of Computing and Mathematics
University of South Wales
CF37 1DL, Pontypridd
e-mail: paul.williamson@southwales.ac.uk

Christopher Tubb
School of Computing and Mathematics
University of South Wales
CF37 1DL, Pontypridd
e-mail: christopher.tubb@southwales.ac.uk

***Abstract*— Non-Player Characters are characters within a video game, which are not controlled by a human participant. While they are mainly used to fulfil a role not designated for a human player, there are occasions when an NPC needs to play in a human role, and therefore needs to imitate appropriate gameplay behaviours, in such a way that it is not easily distinguished from a human player. Navigation is a fundamental gameplay behaviour, focused on how a player traverses the environment when undertaking objectives. This paper explores the possibility of modelling human navigation by modifying A* algorithm with a heatmap derived from human-based data. This is achieved by having participants complete a search and collect experiment. The data is saved for analysis and to develop a navigation model. NPCs using the model undertake the same experiment, but with a heatmap weighted A* graph. The experiment explores adjusting the weight of the heatmap so its influence on the pathfinding varies and a comparison can be made to see which weight better reflects the human results. This paper also investigates switching between the heatmap weighted graph and a standard A* graph, depending on the task being undertaken. This graph switching was used in an experiment to evaluate if the model has an impact on navigation perception when subjects were tasked with fighting against NPCs.**

Keywords—NPC; Player Modelling; Pathfinding; Gameplay; A* Algorithm; Perception.

I. INTRODUCTION

This paper explores using human player data to influence the way NPCs navigate the environment by manipulating a graph-based search algorithm [1]. In the context of an NPC, pathfinding is the mechanism used to find a suitable route between two points on a map. The type of game genre and size of the map can influence which technique is more practical because some solutions are only viable under predefined constraints [2]. In First Person Shooter (FPS) games, a common technique used is A* algorithm, or some variant of this method, where a 2D grid is superimposed over the map, then using cost and heuristic the algorithm calculates a shortest cost path.

This paper expands on the A* algorithm. It focuses on adjusting the weight cost of nodes in accordance with a heatmap. For the purposes of this experiment, the heatmap is generated from data captured from human players roaming the environment, undertaking an experiment in which they need to find and collect eight coins. This data enabled a model to be developed, which captures not only the general areas of navigation, but also intricate behaviours associated with the act of roaming, and the influence they have on pathfinding.

Secondly, this research uses a tagged environment to help determine pathing based decisions. This limits the distance of the routing decisions to only what is within view from the perspective of the NPC. This technique drastically reduces performance cost because, despite the size of the A* graph, the distance between the NPC and destination node is always relatively short.

In A* algorithm, graph switching refers to the process of changing from one graph representation to another during the search for a path. This can be useful in certain situations, such as when the search space becomes too large to be represented efficiently in a single graph, or when the structure of the graph changes dynamically during the search. By switching to a different graph representation, the algorithm can continue to search for a path while avoiding the limitations of the previous representation. For this research, graph switching will be used to switch between two graphs:

- Roaming: This will be the lowest priority objective; it is activated when no other tasks are being undertaken.
- Standard: This graph will be used when the NPC knows the location of its objective. This is essentially all the behaviours except roaming.

While all behaviours could be modelled and a specific graph created for each, it was decided that introducing too many graphs at a time could complicate the analysis process when determining if behavioural based graph switching is a viable technique.

In Section II, this paper discusses the motivation behind a player driven A* algorithm solution. It states why it is important for NPCs to use the same navigational behaviours as human players and why heatmaps generate a useful tool for this purpose. Section III examines research in the field of improving the usefulness of A* algorithm for pathfinding solutions, and some implications of the limitation of these methods. In Section IV, the method for the data capture experiment is explained, which involves human subjects roaming the map to collect several coins. Section V uses the same experiment as Section IV, except in this experiment, NPCs with the roaming model are used and a thorough examination of the results is conducted to determine the applicability of the model and how it compares to the human subject results. The model evaluation in real-time is conducted in Section VI, which details how an experiment was undertaken in the form of a deathmatch between NPCs and human subjects, and then discussing the results from the perspective of how the NPCs navigation was perceived. Finally, Section VII concludes the paper.

II. MOTIVATION

While roaming may appear a random action, it is often a more strategic behaviour where a player tries to maximise scanning efficiency by positioning their character to visually cover as much of the map as possible. This increases the likelihood they will spot their objective and reduce the chance of checking already checked areas of the map [3].

The motivation for this paper is to address how NPCs can roam the environment and increase the likelihood it will interact with a human player. This is important in both single and multiplayer games. In single player games, the game should revolve around the player, so ensuring regular engagement from NPCs is crucial. Regarding multiplayer games or roles generally reserved for a human player, it is important that NPCs can imitate the general behaviours seen in a human player, which include using roaming in a way which is consistent with the routes a player might take. For example, during a death match scenario, players roam the map in search for opponents to eliminate. When NPCs pathfinding is not modelled to reflect the same generalised routes as a human player, it can cause them to patrol areas rarely visited by players.

Heat maps offer a good overview of which parts of the map contain the most interactions. Utilising this information can help develop NPCs that are not hard coded to patrol a certain route, a technique which is commonly used, which is predictable and often recognised by a player. Instead, providing the NPC with human player acquired data so they can undertake roaming with a more human-like characteristic. This should enable naturally occurring interactions, rather than forced encounters where the NPC can appear omniscient.

The perception of omniscience is a common issue with NPCs, which is often caused by making decisions and/or performing actions with information that it should not have. For instance, in some cases an NPC will shoot at a wall with a player on the other side. They should not know the player is there. However, they are provided with an extra layer of information, which can influence actions. An important part of making NPCs appear more human-like is therefore removing this perception. The development of a new model of navigation, as discussed here, is intended to do this. To achieve this goal NPCs can only make decisions based on what they can ‘see’ and internal parameters such as health or ammunition count.

Further motivation for this research is using and modifying solutions for other pathfinding problems, specifically regarding A* algorithm. Video games handle very large A* algorithm graphics by using a variety of techniques to optimize performance. One technique is to use a grid-based system to divide the game world into smaller sections, and only perform the A* algorithm on the section of the grid that the player is currently in. This reduces the number of nodes that need to be searched and allows the algorithm to run more quickly [4]. A similar technique is to use a hierarchical version of the A* algorithm, where the game first performs a rough search of the entire game world and then zooms in to perform a more detailed search in the area of interest. This research aims to increase the practicality of A* pathfinding, by incorporating specific use graphs, which could also work with other techniques, such as hierarchical A* grid-based systems [5].

III. BACKGROUND AND RELATED RESEARCH

Pathfinding is a crucial aspect of an NPCs core mechanics. Some form of navigation is essential in games where the NPC is required to move. The complexity of the pathfinding has increased as games have become more intricate. A* algorithm has remained an important technique in modern games [3].

In FPS games, A* is popular because of its graph-based nature. It can find an optimal route between two points. However, this can lead to predictable routes, which can be exploited. Furthermore, an exponential performance cost can occur when increasing the size of the map, as it increases the size of the graph [6], thus, adding more nodes that could be checked when forming a route.

Comparison analysis was conducted by Permana et al. [7], in which they looked at A*, Dijkstra and Breadth First Search (BFS) in a maze runner genre. They focused primarily on the performance impact of each technique, as well as the efficiency in context of functionality. The results suggest that all methods are capable of pathfinding, however, A* was more efficient computationally.

There has been substantial research to modify A* so it can excel at certain tasks. Sazaki et al. [8] showed that some of the limitations of A* can be overcome by developing a model, which was used in a car racing scenario. This model focused on assisting A* with a Dynamic Pathing Algorithm (DPA). The results demonstrated that it could avoid moving obstacles. This addresses one of the problems with A*; the need to continually update the graph if the map is not static. This suggests that combining pathfinding models and techniques can yield positive results and shows that the effectiveness of A* can be enhanced when aided with other techniques.

Makarov et al. [9] used Voronoi-based pathfinding that has been developed with obstacle avoidance and tactical elements to reduce the probability NPCs will traverse the dangerous areas of the map. They showed that including what NPCs can visually ‘see’, it was able to make tactical and logistical decisions. When incorporating internal information, such as previous enemy encounters, the NPC uses all the data to make decisions, including navigation. This indicates that when making navigation decisions, providing the NPC with more specific data about itself can lead to an adaptable NPC, which could appear more human-like.

Like the forementioned work, the research in this paper uses NPCs vision to make decisions on navigation. NPCs can only move to a location it can visually 'see'. This significantly reduces the size of the pathing and low computational overhead. When using a non-static map, the A* graph needs to be updated on a regular basis, so NPCs do not attempt to traverse non-walkable areas. This can have a negative impact on performance. While significant research has been undertaken to address this issue, it is still a problem that needs to be considered when using A*. The approach proposed in this research could be useful as the NPC could update the graph based on what is in its view.

Research undertaken by Sturtevant et al. [10] has shown that dynamically adjusting the cost of A* nodes based on the terrain they occupy can yield useful results. They used this technique by creating an abstraction layer which deals with terrain cost and dynamic terrain. They determined that from a performance perspective, when used with several different terrain types, the solution can be up to ten times faster in finding a suitable path, while remaining 2-6% optimal. This is important to this research because it shows that weighted environments can be used with other techniques to positively impact the overall pathfinding. This is supported by Pan [11] who proposes a multi-technique approach. They used a bootstrap Jump Point Space (JPS) technique when there are no threats present, then switch to a waypoint-based solution when the NPC detects a threat. This is an interesting approach to a dynamic pathfinding system which responds to the current circumstance of the NPC. When combined with a weighted A* graph, this could help develop a more realistic navigation system because the pathfinding technique will change to reflect the behaviour expected to be displayed.

Anderson [12] introduced a new heuristic method, that can be used to find the best path on a four-connected grid-world. This additive heuristic is very powerful, significantly reducing the number of nodes that need to be searched. However, it does require the use of two abstract graphs in order to calculate the heuristic. The heuristic can be calculated efficiently using on-demand techniques, with instance-dependent pattern database (IDPDB) being the most effective method. This method reduces the total number of nodes searched by a factor of five on room graphs, and a factor of two on video game graphs. On maze graphs, the efficiency of the heuristic is not as high, reducing the total number of nodes searched by a factor of 1.9. Additionally, the execution times are even more noteworthy than the number of nodes expanded. Using the additive heuristic on room graphs resulted in a 29-fold reduction in execution time compared to using the Manhattan Distance heuristic, and on maze graphs it resulted in a 7.6-fold reduction. Using the additive heuristic on video game graphs resulted in a 2.5-fold reduction in execution time. While this is not the same as graph switching, it does show that manipulating graphs can yield significantly improved results when performing specific tasks.

IV. HUMAN ROAMING MODELLING

The modelling phase involved having human subjects undertake a roaming experiment. So, generalised behaviours can be identified and incorporated into a model, which will aim to imitate an average human player roaming characteristics.

A. *Data Capture Experiment*

The data capture experiment was conducted by having subjects roam the map in search of eight coins. A heatmap was generated by adding a standard A* graph, each node was given a collider detection and when a subject intersected with the collider, a counter specific to that node was incremented by one. Constraints were added to prolong the overall length of the experiment, so a more accurate model of roaming could be achieved. Only one coin is present on the map at any given time. This was to prevent chaining where the subject spots a coin as they are moving to collect another coin. When a new coin spawns, it can spawn anywhere on the map, but not in view of the subject current position and cannot collide with terrain. This was to prevent the chance of coins repeatedly spawning close to subjects.

The purpose of this method was due to the separation of the navigation model and A* graph. Wherein, the NPCs uses the graph to plot a route, but it is not part of the overall navigation model.

Figure 1 shows an overview of the map, each number represents a room, additionally, there is a large open foyer area in the centre of the map.

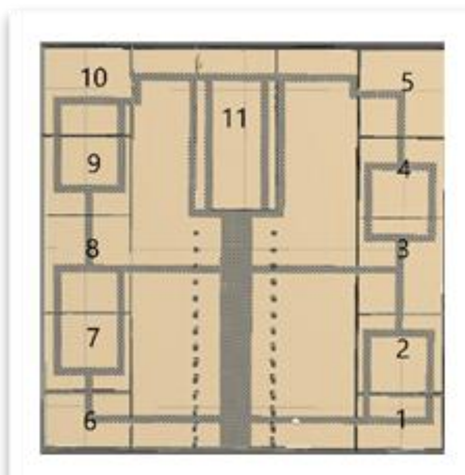

Figure 1. Map Overview

The map was specifically designed this way because it represents the type of map occasionally seen in a video game. This is where there are a number of small, confined areas and then one open space from which the smaller areas are connected.

Non-model related data was captured so a comparison can be made as to the efficiency of model in relation to the overall performance of the roaming behaviour. This was to conclude if the act of roaming is random, or if there was a more significant strategy as to why subjects used certain doorways and routes. Therefore, the position and rotation of the subject was logged every 0.5 seconds, which can be input back into the experiment for behaviour observation by a researcher.

B. *Results and Analysis*

A total of 30 subjects took part in the experiment. Figure 2 shows the combined heatmap of all subjects. The result shows an interesting trend where subjects were more likely to traverse the outer edge of the map, which influenced which doorways were likely to be used.

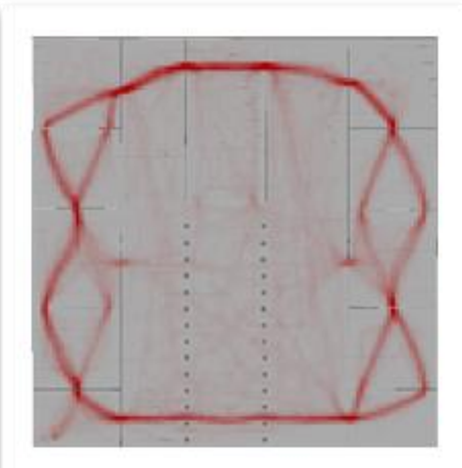

Figure 2. Heatmap and Room Numbers.

This demonstrates that roaming is very strategic, and subjects increase their likelihood of finding a coin by increasing their viewing coverage. It also highlights that roaming routes are funnelled via doorways, and it is likely that subjects' navigation decision-making was primarily limited to the space between doorways. The experiment confirmed initial results, which showed that human players had preferred routes through the map [3]. While this is not surprising as subjects need to use the doorways to traverse the environment, as the map resembles a typical office, it indicates the importance of map design and the strategic value of funnel points. Even in open world maps, generally there are points of interest, with routes, such as roads, leading directly to these areas. There is some heat in the centre of the foyer, this was caused by coins appearing in random locations including the middle of the foyer. This is evident by observing the heat patterns around the foyer doorways, which show a significant change in direction from the initial roaming path towards the location of the coin, indicating that as subjects are entering the foyer, they spot a coin, and then readjust pathing to collect it.

As roaming is strategic, map coverage is therefore an important objective. Figure 3 shows an example of the amount of map uncovered by a randomly selected subject. It shows that at the end of the experiment >95% of the map has been revealed, with a small area in the corner of room 4, which was not uncovered.

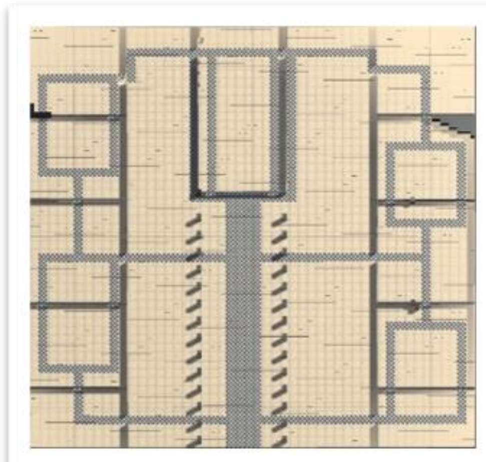

Figure 3. Map Coverage.

There were three key generalised characteristics:

- Player Positioning: Subjects were more likely to traverse the outer edge of the map, thus, increasing viewing angle to cover more map.
- Peeking: Subjects occasionally 'peeked' into the foyer area, this involved moving to the doorway connecting to the foyer for a quick look, before continuing their intended route.
- Rapid Room Scanning: Upon entering a room, subjects were likely to quickly scan the room as they continued to move towards the next doorway.

These behaviours were consistent across most subjects and emphasise that there is a clear logic behind roaming that is not a random undertaking. It is an organised activity where the objective is maximising the efficiency of map coverage.

A critical behaviour that emerged was the speed in which subjects' navigation behaviours changed when new information was presented. While the roaming was methodical, when subjects identify a coin, the behaviour shifts immediately to acquiring the coin. The behaviour changes from looking around the map, to a focused behaviour where the subject remained fixated on the coin and moved directly to retrieve it. Figure 4 shows the results from one subject. The circles show where the subject spotted a coin and immediately breaks away from the roaming route, then after collection they resume on the same roaming route. In one instance, the subject can be seen to traverse the width of the map in a near straight line when spotting a coin.

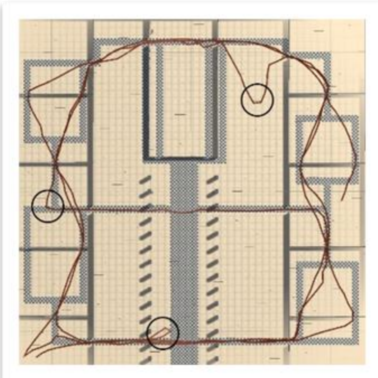

Figure 4. Beeline Behaviour.

This suggests that the navigation model could require a subset of models for the various behaviours associated with moving throughout the map. This could lead to establishing that navigation is more than point to point pathing, but an expression of behaviours related to fulfilling specific objectives. This could explain why subjects were scanning the surroundings when roaming and why they were fixated on their target when collecting coins.

When discussing the act of 'peeking', it is a common pattern among subjects. The data shows that most subjects would look towards doorways they are not moving towards. This gives them the opportunity to see the area behind the doorway, thus, allowing them to scan more of the map. However, at times subjects would actively move to a doorway, scan the area before moving back to their original route (Figure 5). To calculate the parts of the map the observed, a black overlay was added which is removed if inside the subject camera field of view. Ensuring this did not interfere with the experiment, the black overlay was culled from the camera, therefore, would not be rendered during the experiment.

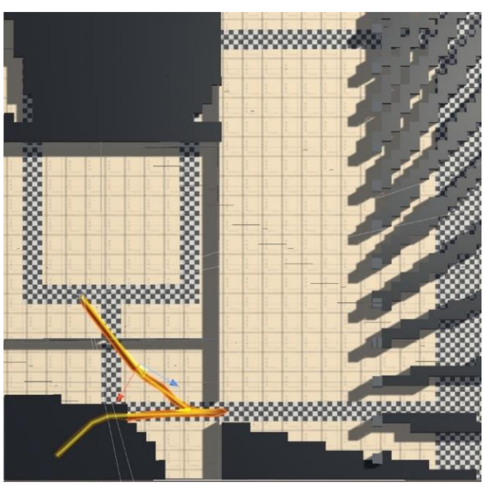

Figure 5. Peeking into Foyer

In this example the subject started in the bottom left and the yellow/red line shows the path they took. They move to peek into the foyer and visually scan a large area, after that they turn around and move towards the doorway adjoining the next room. The importance of this mean they were able to check a significant part of the foyer and a room with very little time lost.

V. ROAMING MODEL ANALYSIS

A roaming model analysis experiment was conducted to determine if the roaming model developed represented the characteristics of an average human subject. The objective was to compare the human data and NPCs directly, to establish the accuracy of the model and determine if there were any negative consequences from using a heatmap.

A. Pathfinding and Roaming Model

There is a distinction between pathfinding and roaming. Pathfinding uses the heatmap weighted A* algorithm to plot a route between two points on the map. Whereas the roaming model controls the navigation decision-making and behaviour of the NPC as it moves between these two points.

The technique uses the heatmap to adjust individual node cost in the A* graph. This was achieved by adding all the specific node counters from the data capture experiment, then subtracting this weight from individual nodes when creating the A* graph. Figure 6 is a pseudo code example of the method used to create the A* graph with heatmap.

```
Loop X grid size
  Loop Y grid size
    create node world position
    check if node is walkable
      int movement penalty = 100
      walkable = True
      movement penalty -= heat weight value
    Else
      walkable = False
    Add node to array
```

Figure 6. Pseudo Code for Heatmap A* Graph.

As the distribution of cost between neighbouring nodes can vary significantly. It was decided that a smoothing technique was required to blur the differences. This was also required to help prevent NPCs occasionally traversing very close to walls as nodes neighbouring non-walkable nodes, such as walls, had their cost increased.

The smoothing technique used a box blur algorithm to normalise the cost of a node. A compromise was made where the box blur was set to 3x3, because when testing 2x2 the blur was not enough and when using 4x4 and 5x5, the smoothing was so significant that the heatmap had no effect. Figure 7 displays the box blur equation. Each number represents the weight cost of a node, the centre number is the node being blurred by adding all weights then dividing by the number of neighbouring nodes.

$$\frac{1}{9}\begin{bmatrix}1,1,1\\1,1,1\\1,1,1\end{bmatrix}$$

Figure 7. Box Blur Equation

Subjects showed that doorways provided pivotal and strategic points on the map, as they are funnel points and are the only means of traversing between rooms. Therefore, as NPCs were restricted to information only in view, doorways became a central point to the model. Each NPC stored personal data about doorways and assigned a dynamic weight cost to each doorway, which reflected the heat observed from the subject experiment. The NPC will attempt to prioritise the doorway in view with the highest weight value. When successfully using the door, it will temporarily decrease the weight to prevent room cycling. Additionally, when NPCs have selected a doorway, the destination point is not a fixed point, but a random location slightly beyond the destination. The purpose of this was to reduce patterns forming when reaching the destination and calculating a new path. This is a common problem with some NPCs as they will perform the same action, at the same time, which can become noticeable by an external observer.

Unpredictability is an element of human players gameplay in an FPS game. There is a probability of performing a certain action in a scenario, but it is never certain. This was reflected in the roaming model, which aims to reduce predictability, but remain logical and consistent with human behaviour. This was achieved by implementing a random number generator of between one and ten, which represented the probabilistic outcome.

When entering a room, NPCs had an 80% chance to scan the room as they moved to the next location, as well as apply special attention to looking at other doorways. Peeking had a special importance when roaming because subjects used this technique to tactically scan open spaces without entering the area.

Lastly, when traversing open spaces, human subjects showed an awareness of their surrounds and took advantage by occasionally looking towards the open spaces, while still moving towards their intended location. This gave the appearance of the subject strafing as they were not moving in a forward-facing direction. This was modelled by enabling the NPC to have awareness of the distance between itself and open spaces to their left and right. Using this distance and a probabilistic algorithm, the model decided whether the NPC should scan the left or right side. After the NPC has successfully scanned the environment, a timer is started to ensure that the NPC does not keep repeating this action in a short space of time, a behaviour not seen in human subjects.

B. *Experimental Protocol*

The purpose of this experiment was to directly compare human subjects and NPCs. It was decided that having NPCs undertake the same experiment as the data capture experiment would provide a good basis to compare the results.

To remain consistent, NPCs run the experiment several times, varying the weight impact of the heatmap, so it could be determined which weight better represented the characteristics of the average subject. There were four different weight profiles. The heavier the weight profile, the higher the base node cost on the graph, which is represented by the darker the colour (Figure 8).

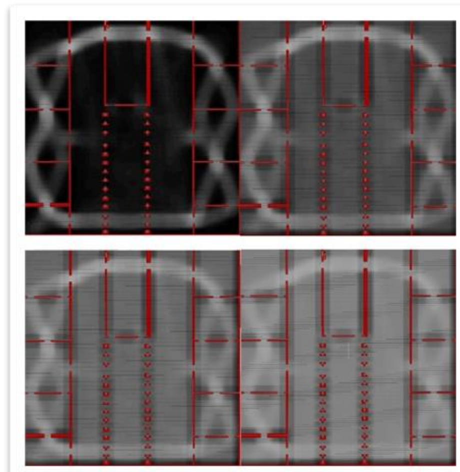

Figure 8. Heatmaps Comparison.

It shows the significant difference between weight costs of the different profiles. The top left profile aims to aggressively influence the NPC to adhere to the heatmap. While the bottom right profile was aimed at being more of a light influence on the pathfinding. This is a promising sign because it means the heatmap is working as intended and the degree of change between the profiles demonstrates that the model should be flexible in its application. This presents a novel approach to pathfinding as the heatmap is not strictly limited to player data. It could be used to prevent NPCs roaming the same areas by increasing the weight cost of nodes based on its own heatmap which is calculated over a set length of time. This technique could also make use of multiple graphs, each with a unique heatmap. It would enable developers to have more control over the navigation behaviours of NPC's, based on the task they are undertaking.

Figure 9 shows an example of the A* graph without the heatmap. While this profile was only used once in this experiment, it is a good comparison to show the influence the heatmap has on the A* graph.

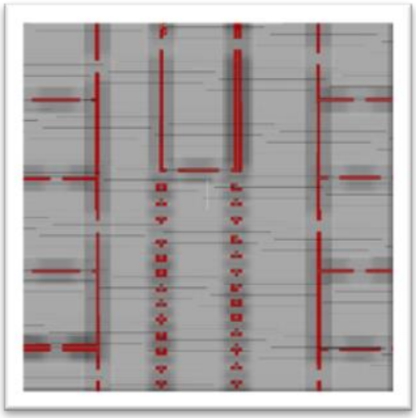

Figure 9. No Heatmap Example.

The node cost smoothing technique was added to all the graphs. This ensured that neighbouring nodes did not have wildly different movement costs. This was essential because in a heatmap, some neighbouring nodes could have significant variation in cost, which would result in the NPC having a jerking motion as they moved.

While some NPCs were required to collect eight coins for a direct comparison with individual human subjects, other NPCs were required to collect forty coins per run, which was the equivalent of five subjects. This number was decided as the purpose of the experiment was to analyse the generalised roaming route of the NPCs. A line renderer was used to track NPCs routing. Therefore, a compromise was required where it would provide enough data to make conclusions, but not too many where the lines become saturated and confusing.

As with the the human subject experiment, the whole map is covered with a black fog, which is instantly removed when entering the view of the NPC. The fog provided a measure of the areas of the map the NPC has scanned and allowed comparison with observations of the human subjects.

C. *Results and Analysis*

When directly comparing the four weight profiles, the results show that amplifying the significance of the heatmap on the cost of the A* nodes, NPCs pathfinding was noticeably affected. Generally, the model has a positive effect on the navigation and each of the heatmaps accurately reflects the roaming patterns observed in the human subject experiment (Figure 10). However, it also had an adverse effect when not roaming. NPCs were taking very inefficient routes to reach a specified location, such as moving to a coin location. On some occasions, NPCs were not making a beeline behaviour after identifying a coin. They would lose sight of the coin and move through multiple rooms, before finally acquiring the coin. Although this behaviour is clearly at odds with that exhibited by the human subjects it does comply with the lowest cost path calculated by the modified A* Algorithm.

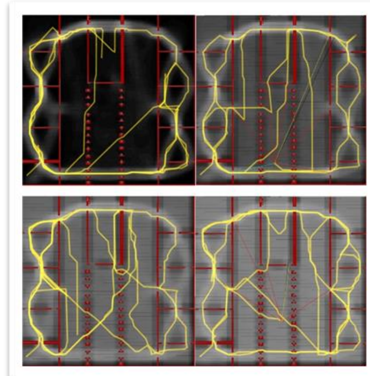

Figure 10. Heatmaps Analysis.

An important observation is in the centre of the map because there is a correlation between the base node cost and likelihood of cutting across the map to reach a destination when roaming. This opens the pathfinding to a degree of flexibility because multiple graphs could be created and the model decides which to use, or if the graph is regularly updated, it can decide how aggressive the roaming should be in relation to the heatmap. This could be useful in a scenario where NPCs are tasked with tracking players and the developer does not want to use scripting to force interactions. Similar research has been undertaken where NPCs are influenced by pheromones, which are generated by other game agents with positive results [13]-[15]. While these examples are generally focused on real-time strategy games, and are intended to explore swarm intelligence, commonality can be derived with the technique presented in this paper. Subject to further investigation, research could be undertaken where the players emit pheromones that temporarily decreases the cost of nodes within the vicinity.

Analysing the model when the coin count was set to forty, the results remain consistent with what was observed with eight coins (Figure 11). In this example, a moderately aggressive base node cost was chosen to help prevent NPCs using the middle of the map to roam, but not too costly that NPCs would take inefficient and irregular route when moving to a coin. The results show the heatmap has a very strong influence on roaming, and when moving to a coin, the NPC would use the middle of the map.

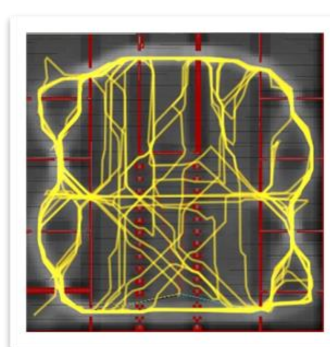

Figure 11. Forty Coin Heatmap Analysis.

When comparing the heatmap against a normal A* graph, the results appear somewhat similar, however, when scrutinising the straightness of the paths, it shows a degree of difference. Figure 12 shows the heatmap A* (left) and the standard A* (right). The heatmap lines show they are not straight, but instead have a slight meandering characteristic.

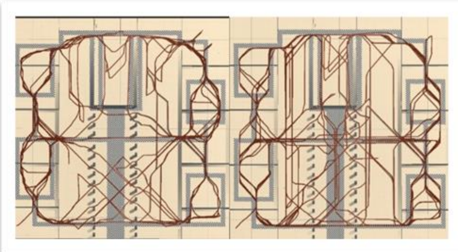

Figure 12. Heatmap A* vs Standard A* Comparison.

Zooming in to specific areas further highlights intricate differences between the heatmap and non-heatmap A*. Figure 13 focuses on a single room. The right image shows a uniform pattern, whereas the left image is less structured that is more reminiscent of human subjects. The heatmap NPC (left) shows that by implementing the random destination point, it affects the pathing where the difference is subtle but clearly different.

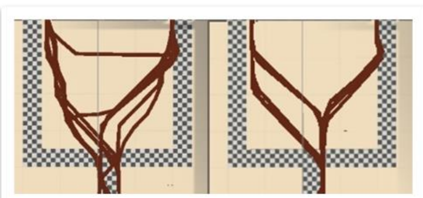

Figure 13. Room Comparison.

When compared to a human subject in a similar room (Figure 14), it shows NPCs using the heatmap is more akin to the subject than the NPC using the standard A* graph. While this is a subtle observation, it is nevertheless important, because when NPCs move in straight lines and display the same movements, these patterns cannot become noticeable when viewed over a length of time.

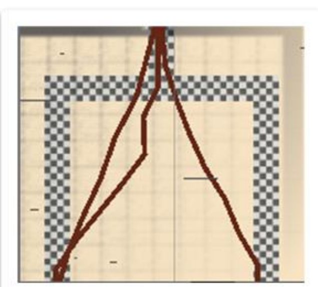

Figure 14. Human Subject Room Analysis.

A key feature of the roaming model was imitating how human subjects entered a room and the time it takes to start moving to a new location. Figure 15 shows the doorway exit and entrance trajectory. The results indicate that the model is working as intended. There are no identical paths, all adhere to a logical tactic, but there was little pathing efficiency cost.

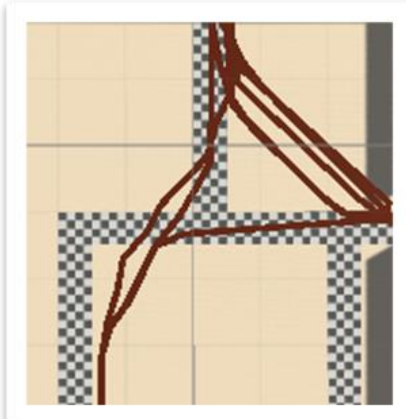

Figure 15. Door Trajectory Analysis.

A key strategy used by human subjects was peeking into large areas. The effect of peeking can be seen most clearly when looking at the fog, the red line indicates the path the NPC followed (Figure 16). It moved into the room, peeked into the foyer area before resuming initial route. Such characteristics are integral to having an accurate imitating roaming model because it projects a degree of intelligence when observed by a player.

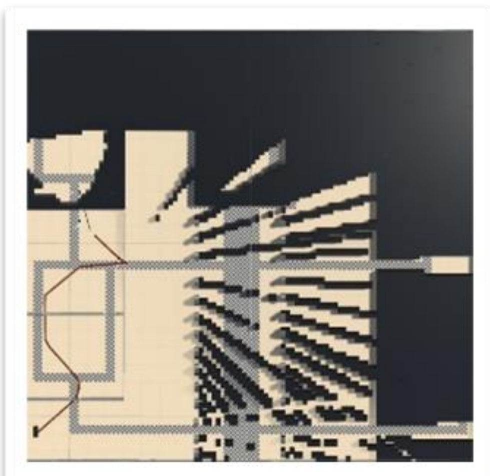

Figure 16. NPC Peeking Identification.

When discussing the map view coverage, with the objective set to collect eight coins. NPCs showed approximately the same level of map coverage as human subjects was achieved (Figure 17). This shows that the roaming model can comfortably seek out and acquire anything within the realms of the A* graph.

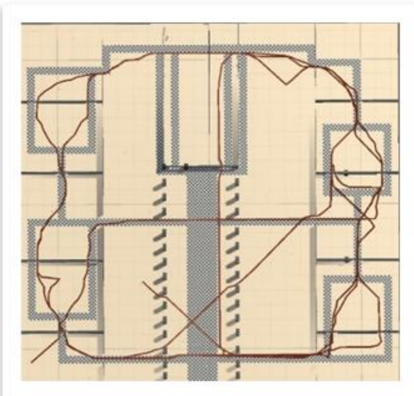

Figure 17. Roaming Model Map Coverage.

Being able to fully scan the environment is a crucial aspect of navigation. When the goal location is not predetermined, if the NPC cannot scan and analyse the entire map, some goals might become impossible to complete.

Finally, when analysing performance, a good measurement is the completion time, as it refers to the efficiency of the navigation. Table 1 shows the time to complete comparison between human subjects and NPCs using various weighted heatmap A*.

Table 1: Completion Time Comparison

Heatmap Weight	Time (Minutes)
0	6.3
25	4.66
50	7.7
100	5.9
Human Subject	6.6

While the results vary slightly, this could be partly due to the randomness of the coin spawns. What the data does indicate is that the model is approximately as efficient as human subjects. This is encouraging as the heatmap is influencing the navigation behaviour of the NPCs and also the efficiency, which are two important areas of gameplay.

VI. Graph Switching

Graph switching is the term being used for using multiple graphs in A* algorithm. Typically, A* has one graph, and based on cost and heuristics, it finds an optimal path between two points. This paper intends to build upon this methodology and introduce multiple graphs, these graphs have a specific purpose and will be used depending on the task being undertaken. This experiment will use two graphs:

- Roaming: This will be the heatmap generated graph and will be used only when roaming.
- Standard: This graph is a standard uniform graph, with a smoothing technique to stop NPCs walking next to walls.

While it is possible to have more than two graphs, and ideally, graphs would be modelled based on the behaviour being displayed. For the purposes of this paper, it was important to directly compare the heatmap graph against a normal graph and not over complicate the research.

A. Graph Switching Analysis Experiment

This experiment involves two deathmatches, in which a subject will be tasked with competing against two NPCs, in an all verses all scenario. The first deathmatch NPCs will use a standard A* graph only, while the second deathmatch, NPCs will be using the graph switching model.

The objective is to achieve eight eliminations before any of the NPCs. All participants, both subject and NPCs, have access to the same weapons and have the same amount of health. The character models are consistent across all characters with no animations. This was decided as poorly implemented animations could affect the perception of the NPCs. All characters use the same movement, Table 2.

Table 2: Character Controls

Movement	Description
Forward	Moves in the direction the player is facing
Back Peddle	Slowly moves in a backwards direction
Strafe Left	Side steps to the left
Strafe Right	Side steps to the right
Look (Aim)	Rotates the camera and weapon, with crosshairs fixed in the center of the screen

The ability to jump was removed as there are no pitfalls. Eliminations are awarded only to the player that got the last hit. Therefore, if two players are attacking the same target, only one will be accredited with the elimination.

The shooting mechanics was the same for NPC and subject. All weapons had the ability to be shot off the hip or aiming down sights. Each weapon had unique characteristics, such as damage, attack speed and bullet spread. There were three weapons:

- Pistol: Slow fire rate, high damage, and high kickback
- Assault Rifle: High fire rate, medium damage, and low kickback
- Shotgun: Slow fire rate, wide bullet spread and high kickback

These weapons were used because they are frequent in modern video games, which should help subjects quickly familiarise with the experiment when they start.

Throughout the map there are six collectable items, three of which were medic packs, which award fifty health points, and three ammo packs that award one clip of ammunition to the currently equipped weapon. The map layout is the same as the previous experiments, this was to help keep consistency in the analysis when comparing data.

Both NPCs will be using a standard combat model, and a finite-state machine (FSM) to transition between gameplay states. The graph switching occurs when the NPC changes states and updates the navigation model to instruct which graph to use should a path be requested. This enables the model to explicitly state the navigational behaviour it needs to project, as the FSM can only occupy one state at a time.

During development a feature was developed to resemble a 'bug' in the code. This provided a small chance that NPCs could become stuck and not move. The purpose for this was to see if subjects noticed the bug, and if so, what affect it had based on the feedback.

As this is a comparison experiment, the data required will be directly from subjects. After the deathmatch concludes, subjects are asked to complete a questionnaire, which is related to their experience in the deathmatch. This paper is a generalised look at player navigation, it was decided that there would be no subject requirement, and anyone that was interested in partaking could complete the experiment once. The experiment files were posted online so the subject could complete it using familiar peripherals and under normal conditions, as though they were playing a video game. The experiment was conducted in English, and subjects remained anonymous, with no personal data required or acquired.

B. *Results*

When comparing and analysing the feedback, the overall results were promising and showed the graph switching working as intended. It also illustrates that subjects identified the irregular behaviour caused by the bug and it had a detrimental impact on how they perceived the NPCs.

The overall impact on perception can be observed when analysing overall navigation feedback because this is when subjects directly interact with the NPC or observes from a distance. Figure 18 displays the specific feedback from the non-modelled NPC (left chart) and graph switching NPCs (right chart).

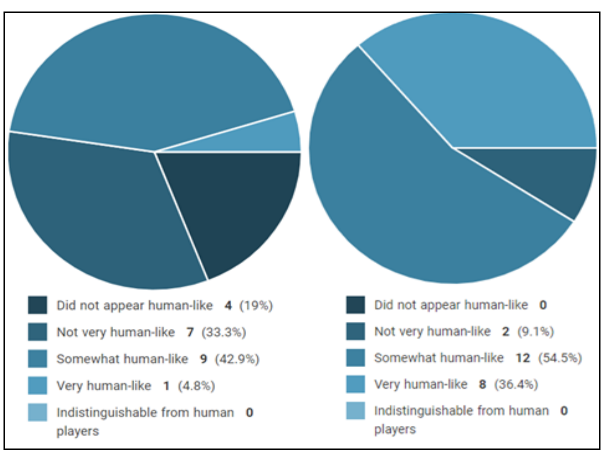

Figure 18. Subject Feedback Comparison

The data indicates that roughly 90% of the subjects considered the NPCs with graph switching to appear fairly human-like. While more than 50% of the standard A* NPCs were poorly received. It can be assumed that this improved perception in the graph switching NPCs is directly related to how they appeared when roaming, because the A* graph for combat was the same as the non-modelled NPCs.

When analysing the data, it is important to focus on areas where the perception was at its weakest, so these areas can be addressed, a good metric for this was directly asking the subject for feedback as to where the model failed.

Figure 19 displays a portion of the feedback from the subjects that answered the question relating to the graph switching NPCs only.

I do not know, when I saw them I engaged
Seemed fine to me, movement was a little off when moving through doors
movement in fight
didnt see any thing wrong
one npc got stuck, other than that, not complaints
Nothing noticeable
wasn't paying attention
They were fine accept when changing direction suddenly, looked like they were on ice

Figure 19. Specific Navigation Feedback

The results appear quite mixed with some not directly addressing the weaknesses of the model, however, some offer valuable insight. For instance, the moving in combat has yet to be modelled and this was indicated to have broken immersion. It should also be noted that some comments are confusing movement to animation, so the suggestion that NPCs moved like they were on ice, if the character was controlled by a human, the appearance would be the same.

The feedback shows that the 'bug' did occur and some of the subjects witnessed this and has reported it as the least believable part of the navigation modelling. This is a crucial discovery, because it shows the potential impact that a minor bug can have on the perception of the NPC, and that believability needs to be a continuous and when it fails, regaining the illusion can be difficult.

Figure 20 highlights a section of the feedback when asked which part of the NPCs was the least believable in relation to human player gameplay.

decent, but they did seem to move to straight
npc was stuck in door
one npc got stuck, other than that, not complaints

Figure 20. Feedback Specifying Bugs

This highlights just how significant bugs can influence perception and break immersion, and while it may be a minor issue, the impact can be lasting. Furthermore, it illustrates that subjects have a general level of awareness, and when this awareness detects something not quite right, it can also influence opinions.

Finally, when categorising navigation into three parts and asking subjects if any of these areas were weak. Figure 21 shows that only 36% of the subjects submitted a weakness in the navigation.

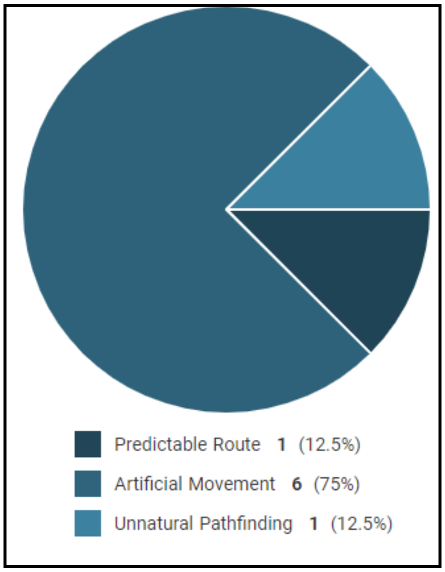

Figure 21. Model Weakness Feedback

Most of the feedback indicate that 'artificial movement' was the least believable, and when compared with some of the comments from the open-end question about navigation (Figure 19), it implies that the break in immersion could be partly from the intentional bug add to the experiment.

Graph switching represents a step forward in imitating human navigation behaviour, it showed that a multi-model strategy is required to imitate the various behaviours associated with the multitudes of navigational purposes. While for this experiment it proved a good solution and encapsulating more behaviours should improve the perception of the NPCs and yield even better results. However, eventually, this methodology could result in exponential complexity, if too many navigational behaviours are required. It could therefore be argued that a better approach moving forward would be to tie individual navigation behaviour models to the decision-making modelling, this would give the decision-making modelling direct control over the navigation and instruct it which subset of models it needs to be using at a given time. If used with Finite-State Machine (FSM) or Goal-Orientated Action Planner (GOAP), a graph could be modelled or linked to each goal or state.

VII. Conclusion

The objective of this paper was to explore if human player navigation data can be used to create a heatmap, for the purpose of adjusting the weight cost in an A* graph to influence the pathfinding.

The roaming model demonstrates that using a method, which restricts pathing decisions based on what it can see, and using a heatmap weighted A* algorithm, a good imitation of the general roaming behaviours of a human player can be achieved. This hybrid model was able to take advantage of tags in the environment so the NPC could make decisions based on what was in view. Thus, removing omniscient characteristics often associated with NPCs and which can be clearly identified by players. The heatmap weighted A* offers a unique approach to influencing pathfinding so that NPCs use frequently travelled areas, making the NPC interactions more natural, than scripted interactions that can appear forced.

However, it was clear that using a designated A* roaming graph had negative implications when used for other navigation tasks. Therefore, the practical application of the model would need to incorporate a multiple graph solution, in which the A* would be applied to an appropriate graph, based on the task the NPC is undertaking.

The graph switching indicated that it was a viable solution for changing the navigation behaviour of NPCs, depending on the type of task they were undertaking. While more research will be required to evaluate the full usefulness of this technique, modelling specific navigational models could enable NPCs to have more depth to their gameplay. Furthermore, it will allow developers to have greater control over how the NPC should be perceived and when modelled on human-based data, could become more believable.

References

[1] P. Williamson and C. Tubb, "Heatmap Weighted A* Algorithm for NPC Pathfinding," In SIMUL 2022: The Fourteenth International Conference on Advances in System Simulation, pp. 15-21, 2022.

[2] I. Millington and J. Funge, "Artificial intelligence for games," CRC Press. 2018.

[3] P. Williamson, "Modelling Human Behaviour in a Virtual Environment," [Unpublished Masters Thesis]. University of South Wales, 2019.

[4] Z.A. Algfoor, M.S. Sunar and H. Kolivand, "A Comprehensive Study on Pathfinding Techniques for Robotics and Video Games," International Journal of Computer Games Technology. 2015.

[5] D. Foead, A. Ghifari, M. B. Kusuma, N. Hanafiah and E. Gunawan, "A systematic literature review of A* pathfinding," Procedia Computer Science, pp. 507-514, 2021.

[6] W. Ziqiang, H. Xiaoguang and L. Xiaoxiao, "Overview of Global Path Planning Algorithms for Mobile Robots," Comput. Sci., vol. 48, pp. 1-16, 2021.

[7] S. Permana, K. Bintoro, B. Arifitama and A. Syahputra. "Comparative analysis of pathfinding algorithms a*, dijkstra, and bfs on maze runner game," IJISTECH International J. Inf. Syst. Technol., vol. 1, no. 2, pp. 1, 2018.

[8] Y. Sazaki, H. Satria and M. Syahroyni. "Comparison of A∗ and dynamic pathfinding algorithm with dynamic pathfinding algorithm for NPC on car racing game," 11th International Conference on Telecommunication Systems Services and Applications, pp. 1-6, 2017.

[9] I. Makarov, P. Polyakov and R. Karpichev, "Voronoi-based Path Planning based on Visibility and Kill/Death Ratio Tactical Component," In AIST, pp. 129-140, 2018.

[10] N. R. Sturtevant, D. Sigurdson, B. Taylor and T. Gibson, "Pathfinding and abstraction with dynamic terrain costs," In Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, vol. 15, no. 1, pp. 80-86, 2019.

[11] H. Pan, "Pathfinding and Map Feature Learning in RTS Games with Partial Observability," In AIIDE Workshops 2021.

[12] K. Anderson, "Additive heuristic for four-connected gridworlds," in Proceedings of the 3rd Annual Symposium on Combinatorial Search, pp. 10-15, 2010.

[13] D. Daylamani-Zad, L. B. Graham and I. T. Paraskevopoulos, "September. Swarm intelligence for autonomous cooperative agents in battles for real-time strategy games," 9th International Conference on Virtual Worlds and Games for Serious Applications, pp. 39-46, 2017.

[14] X. Chen, et al., "Towards believable resource gathering behaviours in real-time strategy games with a memetic ant colony system," Procedia Computer Science, vol. 24, pp. 143-151, 2013.

[15] A. Rafiq, T. Kadir and S. N. Ihsan, "Pathfinding algorithms in game development," In IOP Conference Series: Materials Science and Engineering. vol. 769, no. 1, pp. 1-12, 2020.