3D Heat Map

Eric Baker
Rochester Institute of Technology

1 Abstract

Many different techniques can analyze user game testing data, each with its own strongpoints and drawbacks. One of these techniques is to build a heat map visualization that is quick and easy to read. However, this technique has the drawback that the heat maps only generate into a 2D image that removes one dimension of space. This research demonstrates how to generate a 3D scene for a heat map in Maya 2010 that removes this drawback.

2 Introduction

An ever increasing aspect of any games development cycle is User-Oriented game testing.[4] Many different processes employed by developers can collect this data and then analyze it. The industry is shifting from conducting individual playtesting sessions. These sessions have several drawbacks to implementing systems that automatically record gameplay data, also known as gameplay metrics.[5] As far as analyzing this data, different techniques can make this quick, easy and effective, e.g. text-based excel tables to visualized heat maps.[6] Each technique has its benefits and drawbacks when applied to a particular game or data set.[6]

3 Background

3.1 Gameplay Metrics

Gameplay metrics have been increasingly used as the method for collecting user game testing data due to its benefits over previously used methods, like recorded playtesting or surveys.[5,6] These older methods have the drawback of only collecting a small portion of data from the total game community.[6] Moreover, these methods allow for bias from the testers based on the method used to collect the data. Gameplay metrics can eliminate these drawbacks.

Gameplay metrics are collected by implementing an automated system in the game that records data on the state of the game that the designers have decided to record. A sever solution sends data to the developers via the network connection.[2] Thus, anyone who plays the game online contributes feedback on the game, which also removes all human bias.

3.2 Heat Maps

Heat maps are a common visualization of gameplay metrics used in online first person shooters. Games, like Halo Reach and Team Fortress 2, both use heat maps, and they also provide them to their respective game communities.[8,9] A heat map represents the amount of action occurring at a particular location with a color.[4] An action can be anything that can occur in the game, e.g. player position or player death. The colors typically range from blue, indicating a low frequency of activity to red, indicating a high frequency of activity. These color values are then rendered over a 2D image of the particular level usually from a top down perspective.

While generating a heat map from a 2D perspective remains effective, it does have a major drawback: using a 2D heat map in a 3D level causes the viewer to lose one of the dimensions. When viewed a top down heat map lacks the height of the action. For levels that take place on a flat area, or games that do not allow the player any type of vertical movement, a 2D heat map has no drawbacks. However, for a level that takes place in a building with multiple floors or a game that allows the player to move in the vertical direction, a 2D heat map may give false or incomplete information.

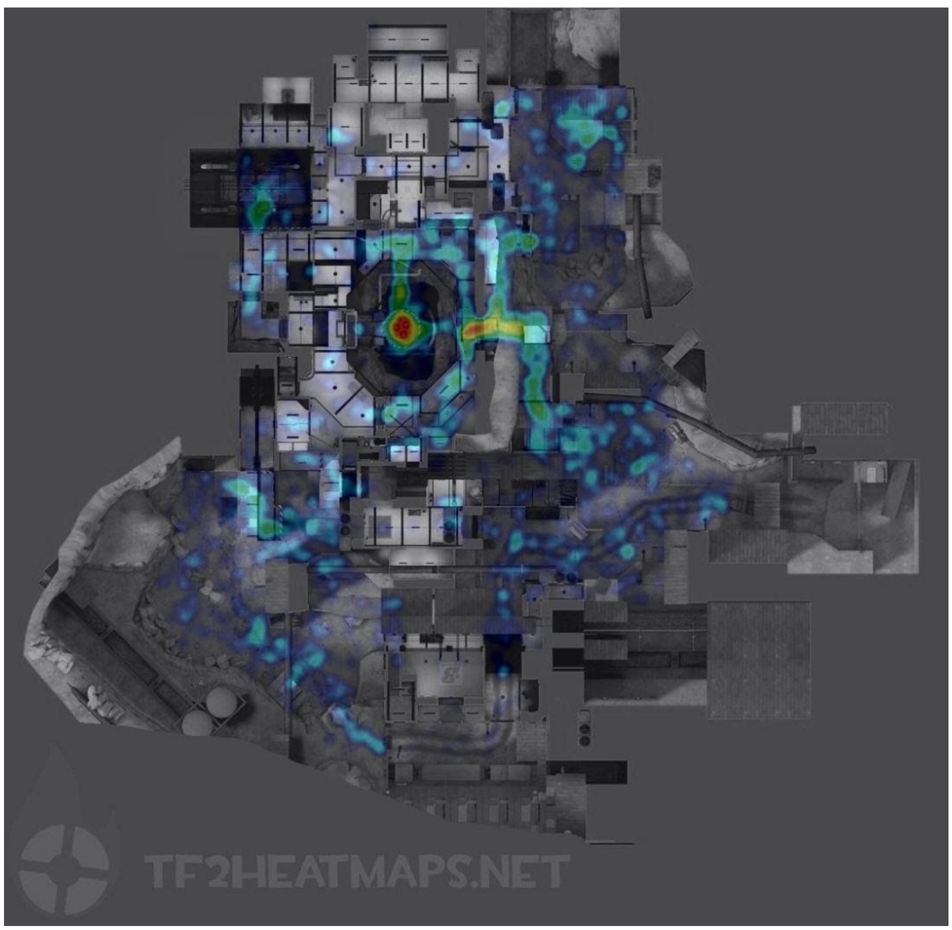

Figure 1: Team Fortress 2 heat map [8]

4 Hypothesis

Creating a heat map in three dimensions generates more accurate data, which provides more effective gameplay metrics to game designers and developers.

5 Experiments/Evaluation Methodology

The first involved determining the effectiveness creating a tool to view a 3D scene or to use preexisting 3D software. The decision had three key factors: implementation cost, ease of use, and the development team. The development and implementation of a new tool that allows the researcher to view a 3D world would take considerable time and effort. On the other hand, if the researcher was already familiar with the 3D modeling software, Maya, and the development team was using this software to create their 3D levels, the decision to use Maya would allow more time to experiment with visualizing the gameplay metric data.

5.1 Step 1: Log Gameplay Metrics

To begin visualizing the heat map in 3D requires defining the gameplay metrics to be utilized. The first step towards synthesizing this data is to determine the gameplay metrics and the method to gather them from the game. In a game like Remote Shepherd with little player movement, it is better to collect data on the AI agents in the game world. A large amount of data (AI position, current AI behavior node, cause of AI fear, etc.) would help in modifying the AI system. However, to limit these test cases I focused only on the. A logging system created for this topic stored the required data into a text file. To collect the data, each AI agent sent a message to the logging system with its position and type during its update cycle. The position data consisted of the X, Y and Z values of the agent's location in the game level. The type data was the enumerated value representing the AI agent type: Walker, Jogger, Reader, Mobster, Cop, or Bodyguard.

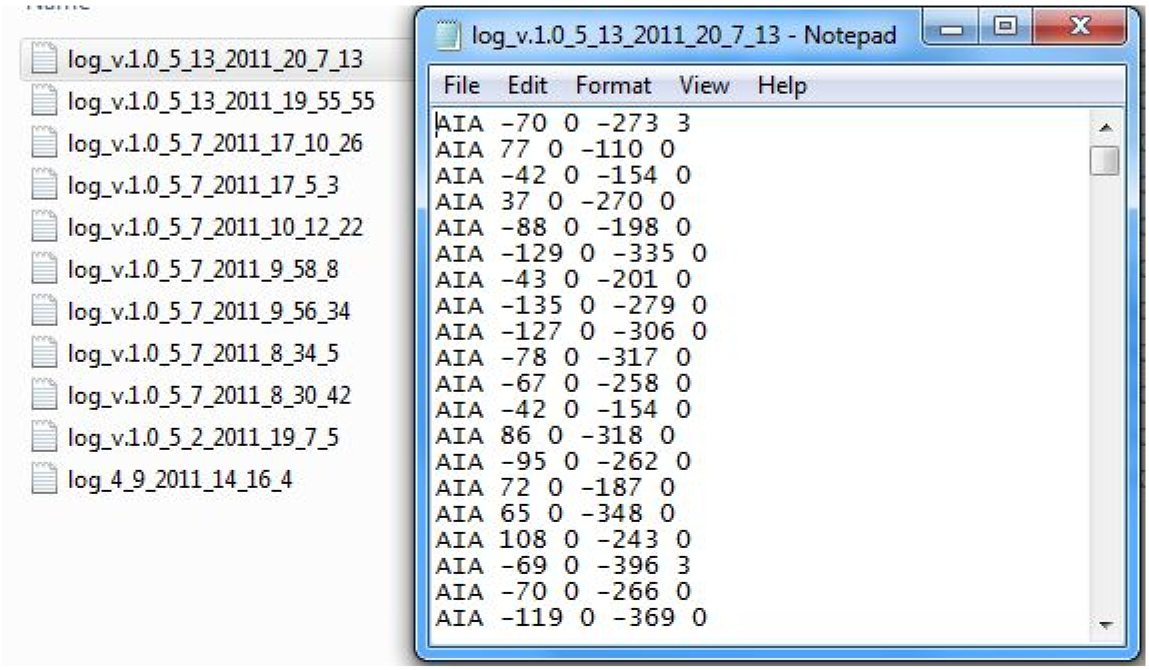

Figure 2: Log Files

5.2 Step 2: Aggregate Logs

After the logging system is implemented, the next step combined all the individual log files into a single database. At this point, two decisions were made: the type of database system to implement and where to implement it. The database system could be implemented in one of three places: in Maya, in the game engine or in a new application. In the end, I implemented the database system in Maya because it allows the system to smoothly integrate with the next step of the process. As for what type of database to implement, there were two options, SQL or a text file. I opted to use a text file database to reduce development time.

5.3 Step 3: Build Maya Add-on

The final step was to build a GUI add-on for Maya that would create the database and generate the heat map. Typically in Maya GUIs are built using a Python framework built on top of Mel. The process for building the GUI system necessary to complete this research was strenuous, as there is not much existing documentation on the process. The original goal of the system was to allow the user to apply filters on the database that would allow them to only collect the desired data for the heat map. However, this process would have required additional time to gain the necessary proficiency in Maya's Python GUI framework. Thus, the system gathers all of the data in the database and uses it in the generated heat map.

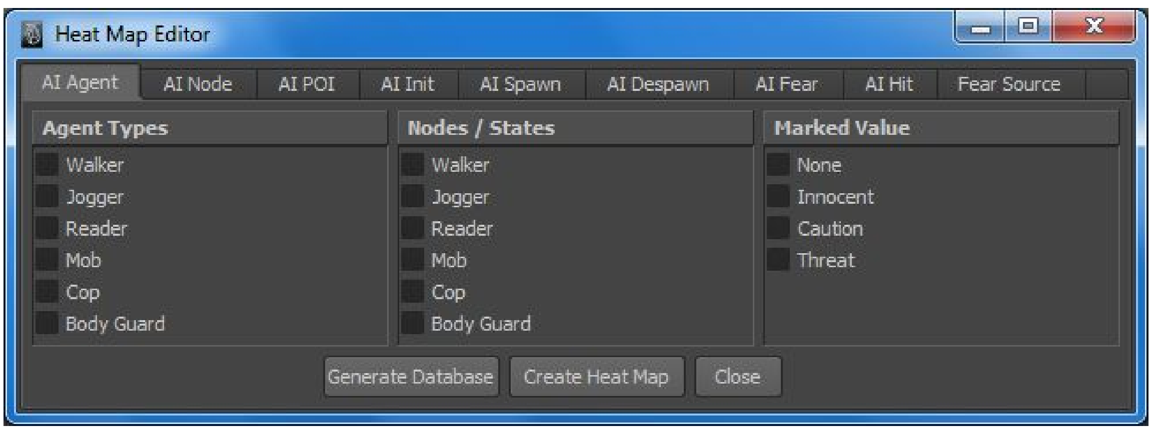

Figure 3: Python GUI

5.4 3D Heat Map Visualization

Experimentation with different visualization techniques to generate the 3D heat map began after completing the aforementioned process. The first aspect explored was manipulation of the values from the database. In previous research done on generating 2D heat maps, I discovered that the resulting image conveyed less interesting information with a liner scale or no manipulation of the values. However, applying a logarithmic scale to the values decreased the strength of the high extremes, which allowed values on the low end to become visible.

The next aspect examined was the shape of the object to represent a point on the heat map. In a 2D heat map, each point is represented by a pixel, as they are an image, whereas the 3D heat map was a complete 3D scene. A sphere was first selected as it would it would appear the same from any viewing angle and would not obstruct the grid space for the heat map point. Using a sphere turned out not to be a practical option, as it took the most time for Maya to generate the mesh. The uniform appearance was still important so the number of subdivisions of the sphere were reduced to the minimum to speed up generate time. At this low poly form, the sphere lost most of the uniform appearance which influenced the decision to use cube. A cube was the fastest 3D object for Maya to generate which allowed for the grid size of the heat map could be lowered to 1’x1’x1’.

At this point the focus shifted from generating the heat map to making the heat map easier for the user to examine. The first technique applied was to remove all color from the level so that the only color came from the heat map. This technique is used in other heat maps and serves as an effective way to make the heat map easier to read, because the eye is not mislead by colors of the level. This technique also allowed each point in the heat map to stand out from the level. However, there was still a problem with objects in the level blocking the view of some points. This obstacle is something that does not affect 2D heat maps because they are drawn on top of the image of the level.

As challenges with visualization of the 3D points through objects occurred, it was fortunate that the heat map was implemented in Maya and the level designer placed meshes into categorized layers. Maya has the built-in ability to change visibility attributes and selectability of all meshes or objects in a layer, which allows for all the meshes in the background of the level to be made invisible. The ability to make objects invisible, made movement in the scene faster and removed unimportant geometry from view. The objects inside of the level that previously blocked the view could also be rendered in wireframe, which made all the heat maps visible. To finalize the process all meshes and objects from the level were made unselectable to make selecting particular points in the heat map easier. This also made navigating the heat map much easier.

The final technique applied made the size of each cube reflective of its heat map value. This technique was used to achieve two benefits: to increase the heat map's readability and to improve the ability to view down a row of points. Ultimately, this technique did increase the heat maps readability by acting as a level of detail for the heat map. A glance and a wide focus makes the areas of high action stand out and as the user narrows the focus the points of low action become more noticeable. Unfortunately, this technique was not visibly successful in allowing the users to view down a line of points.

6 Results

Through the process 3D heat maps were generated that gave valuable information on how his AI agents were behaving in the level. Prior to the development of the 3D heat map, the game development lacked a way to freely navigate the level and observe how the AI agents behaved. The heat map provided information used to adjust the frequency of obstacle avoidance, as the agents were walking off the path.

It also explained behaviors that at first glance appeared to be random, until viewed in the heat map. Figure 4 displays an area where the AI agents walked off the navigation path because they were avoiding a collision with another AI agent. The small blue cubes show that a very small amount of the AI agents traveled to this location, and the ones off the black path indicate that only a few of the agents are traveling off the navigation mesh. The larger green and yellow cubes show us the areas where there was average to high amount of AI agent's activity. Around the wireframe bridge on the left, a high number of AI agents got "confused," and some of them walked off the navigation mesh.

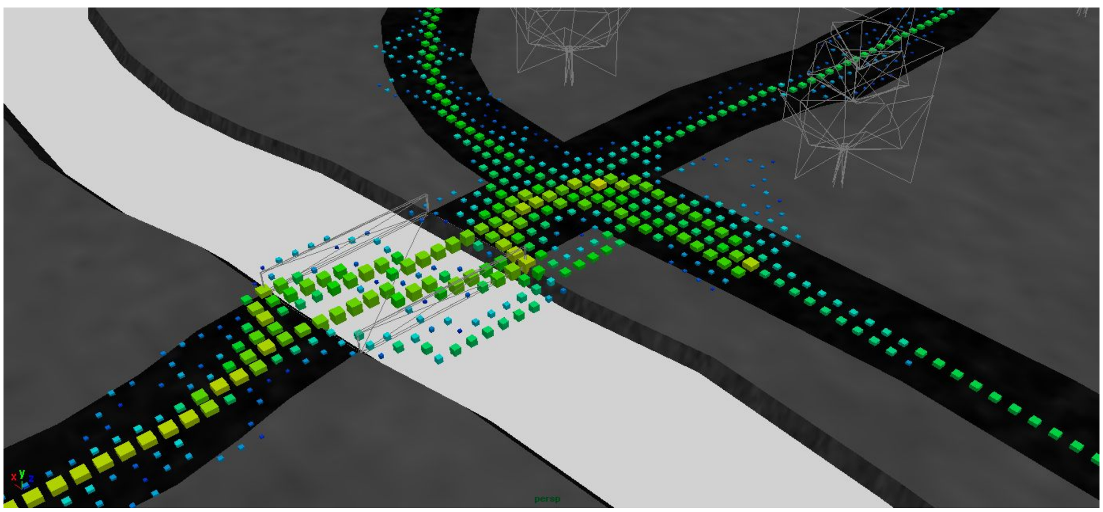

Figure 4

The 3D heat map was also a valuable tool to balance the level for the AI agents. As previously mentioned, the heat map shows areas (e.g. Figure 4) where the path was too narrow for the AI agents to maneuver around each other. The heat map provided a guide to edit the level to avoid the AI agents walking off path. Another use for the data can be seen in Figure 5, which displays points of interests that saw a high amount of use, while others that saw little to none. The green cubes show the areas that were traveled the average amount by the AI agents, where they use most of the area created here. However, the large red cubes show areas were traveled by an extreme amount of AI agents. In this case, it is a concern because the AI agents do not use all of the benches in this area. Designers can use that information to improve the AI behavior.

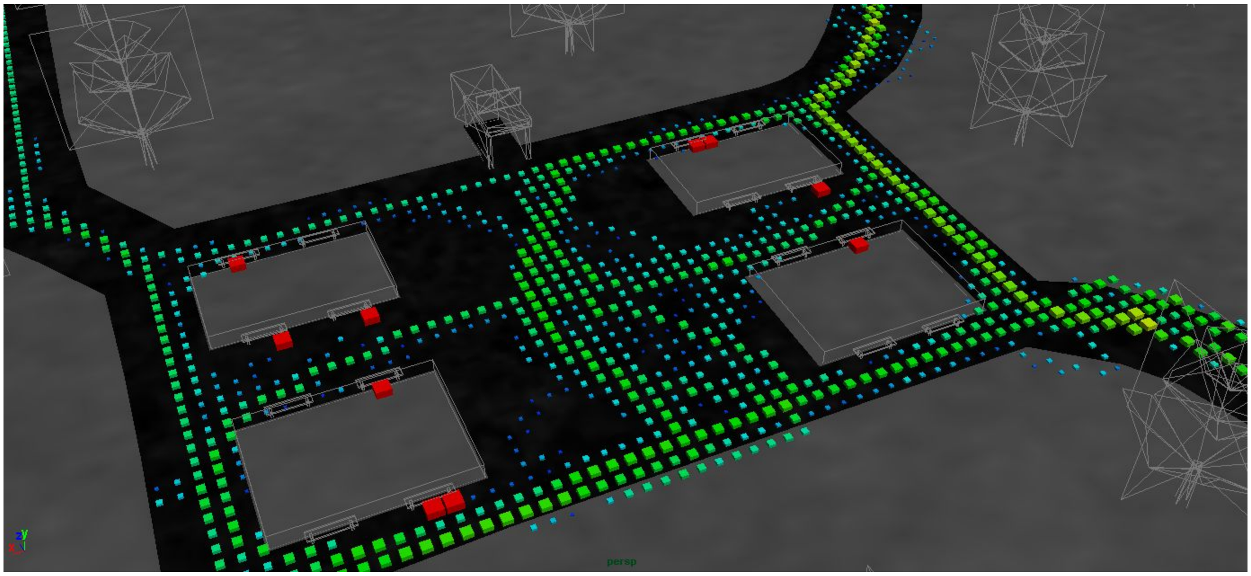

Figure 5

7 Conclusions

Through this research, I determined that a successful 3D heat map can be generated using Maya as the scene viewer. A potential drawback to using a 3D heat map is the issue of perception and where a point truly lies if it is not adjacent to another mesh, which is demonstrated in Figure 6. It is possible to address this issue with Maya by selecting that point, focusing the camera on it and then rotating around that point to determine its position. This drawback may be the reason there is no current documentation of this type of heat being used in the game industry. A definite conclusion as to the reason for a lack of documentation cannot be proven at this time as the above mentioned ideas into the causes are merely speculation and would require further research into the area to discover why 3D heat maps are not currently in use.

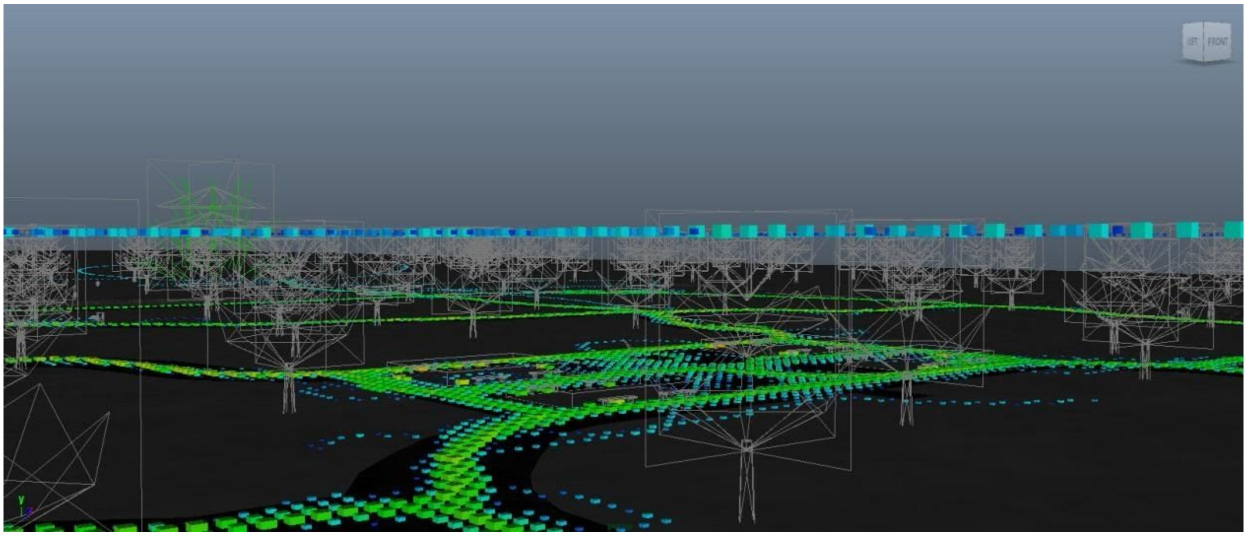

Figure 6

8 Future Work

For future heat map generation, there are four aspects that can be improved upon to increase the speed of the application and expanded its capability. The first would be to replace the text log files and text database system with an SQL database. This change could improve the speed of the application and could improve the ability to create filters on the data to be collected from the database. Another change would be implementing a system to allow the user to filter what data is collected and used to generate the heat map, which would also vastly increase the systems use to the user. Most importantly this system needs to be tested in a game that has movement in all three dimensions. The AI agents in this game could only move on the XZ plane, and this factor limited the amount of work that could be done to improve the drawback of perspective in a 3D scene.

9 References

1. Anders Drachen, Alessandro Canossa: Towards Gameplay Analysis viw Gameplay Metrics. (2009)
2. Anders Drachen, Alessandro Canossa: Analyzing Spatial User Behavior in Computer Games using Geographic Information Systems. (2009)
3. Ben Medler, Michael John, Jeff Lane: Data Cracker: Developing a Visual Game Analytics Tool for Analyzing Online Gameplay. (2011)
4. Ben Medler, Brian Magerko: Analytics of Play: Using Information Visualization and Gameplay Practices for Visualizing Video Game Data. (2011)
5. Erik Andersen, Yun-En Liu, Ethan Apter, François Boucher-Genesse, and Zoran Popovi´c: Gameplay Analysis through State Projection. (2010)
6. Nacke, L., Ambinder, M., Canossa, A., Mandryk, R., Stach, T: “Game Metrics and Biometrics: The Future of Player Experience Research” Panel at Future Play. (2009)
7. Simon Joslin, Ross Brown, Penny Drennan: Modelling Quest Data for Game Designers. (2006)
8. http://www.tf2heatmaps.net/. TF2heatmaps.net. (2011)
9. http://www.bungie.net/stats/reach/globalheatmaps.aspx. Halo Reach: Heatmaps. (2011)