Institut National des Sciences Appliquées de Toulouse

GMM

Generating maps from player data

Cristian Robu
Ronja Friman

Docentes:
Dr. Guillaume Gaudron

October of 2023

Index

Figure List

1 Introduction

for who we're doing that

EXPLAINING THE GAME

The work environment for this project is provided by PUBG: Battlegrounds [1]. PUBG is a battle royale game in which up to one hundred players parachute from a plane to an island. The plane crosses the game map from a direction that varies with each round. Players can choose at which point they want to jump off the plane, and this decision should be made rather quickly. On this island, players can search for weapons and loot to help them kill other players and avoid getting themselves killed. Loot can usually be found inside buildings but also from other sites. To avoid getting killed at the start of the game, finding weapons is critical. Another important aspect is being able to hide if necessary, which is easier behind or inside buildings.

At the beginning, the whole map is available, but players can only reach a certain distance from the plane. Over time, the available safe area in the game decreases in a circle, meaning that players must reach the circle to survive. Initially, it begins as a large circle but then becomes smaller and smaller. It is generated randomly, so players don't know the first circle at the time of deciding where they want to land. That said, the probability of reaching the circle in time is higher when close to the center of the map at the beginning.

Players can travel by running but also with vehicles that can be found in the game. Vehicles are important if a player is far away from the safe area, but they are also loud, so the usage of them has to be a strategic one. Vehicles can typically be found near major roads, especially junctions.

In PUBG, there are multiple maps, but in this project, we will focus on only one of them, Miramar. Miramar is an 8x8km city-centric map with desert and rural areas. The map can be seen in Figure 2. The playing area is surrounded by canyons and mountains in the west and northeast and the sea in the south and southeast. In the north, the map is bounded by a metal wall. Miramar features numerous cities, ranging in size, that are named in the map. The cities are connected to each other via several roads. There are also several individual buildings along the roads around the map.

- what we're doing

- why we're doing

How is this going to help

This report examines the capabilities of generative AI in the realm of gaming but also highlights the role of statistical analysis in utilizing player data. The aim is to use these tools to explore whether alternative game maps, ensuring the same richness of gameplay, can be generated from player data and the map itself. This project aims to provide a guideline for game designers, allowing them to focus in the map design itself without worrying about whether the map will be enjoyable for players.

Gameplay refers to the way players interact with a video game. It's the experience of playing the game, influenced by the game's mechanics, choices available to the players, rules, and design. In our case, we are trying to imitate the choices available to the player from a map already existent in the game, to generate a new one.

how are we're doing that

This report focused on analyzing if variables that are related to the game such as roadmap, building dispersion, distance to the map's center, and the data acquired from players jumping patterns can emulate gameplay. The way it was done was the map was divided in a 10x10 grid and it was calculated the density of buildings, roads and how far was the square from the center of the map in combination with how many players landed in that square. A model was created with the idea of predicting how player landing numbers would differ if the settings were changed, such as the buildings would be in different positions on the map.

2 Statistical Work

Data Acquisition

The foundational phase of this study involved data acquisition. In this project, the focus was on obtaining comprehensive game records from an online platform [2], emphasizing players who had participated in multiple games. This selective approach enabled to compile a richer dataset that would include players with more extensive knowledge of the game than those who had played only once or twice.

In particular, the concentration was on tracking the journey of players from their initial points of action (jumping off the plane) to their chosen destinations (landing points). These coordinates might serve as critical markers of strategic decision-making. By extracting and focusing on these key data points, the aim was to capture an aspect of player behavior. After considering whether to include trajectories of players from jumping to landing, the decision was made to focus solely on the landing points due to the complexity of incorporating the vectors of trajectory points into the created model.

For this project the data obtained includes 1952 games that has 118118 players in total. The collected data includes players' landing points as XY coordinates ranging from 0 to 800000, with X representing the horizontal position and Y representing the vertical position. Additionally, it includes the altitude of the landing point as the Z coordinate. To align with the size of the map, which is represented as an image of 4096×4096 pixels, the X and Y coordinates were scaled to range from 0 to 4096.

Density Heatmap

The hypothesis regarding landing points was that players would predominantly choose locations near cities, with a higher concentration in larger cities, as loot is typically found inside the buildings. Additionally, it was thought that more players would choose to land near the center of the map, driven by two main factors. Firstly, proximity to the safe area at the beginning of the game is greater near the center. Secondly, the trajectory of the plane usually crosses somewhere near the center of the map, increasing the probability that locations are reachable from the plane, especially at the center. Another aspect of the map that could affect the landing point is the

distance from the road. When players are far from the circle, roads provide a faster route to safety.

After collecting data, the objective was to identify patterns in players' landing locations to verify and validate the data and the initial hypothesis. One approach to achieving this goal was to create a heat map of landing points, as shown in Figure 1. For the heatmap, the density of the players' landing points was normalised to better distinguish the areas of the map where landings are common.

The heatmap clearly indicates that players tend to favor landing near the center of the map, especially at specific points. Comparing the heatmap with the game map in Figure 2, it can be inferred that these points correspond to cities on the map. The most frequented landing points include San Martin and Pecado, which, although not the largest cities, are very close to the center of the map. Another popular location is Power Grid, which, while not a city itself, can be considered as one due to the presence of several buildings. Larger cities, such as Los Leones and El Pozo, are also popular landing points, but being slightly farther from the center makes them less popular. It's important to note that the larger cities may appear less dense on the heatmap, potentially due to players scattering across a larger area. The significance of being near roads when landing is not evident in the heatmap, but a more detailed statistical analysis on this aspect will be conducted later in this report.

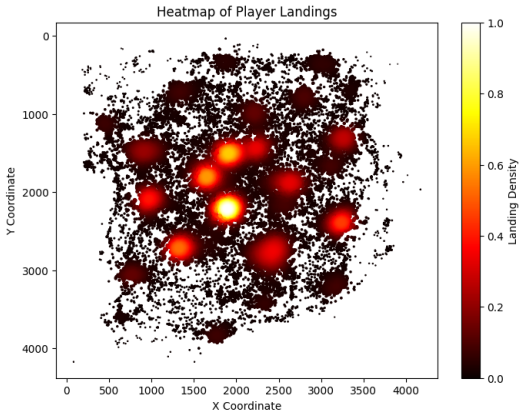

Figure 1: Density Heatmap of Player Landing Locations Indicating High-Frequency Areas

Figure 2: PUBG Miramar map

As we proceed, our next steps involve exploring additional variables that may impact player behavior, such as the density of buildings within these named map locations and the layout of roads. By analyzing building density, we aim to understand how the potential for loot and cover influences landing choices. Similarly, examining the road network will shed light on how players plan their movements post-landing, considering accessibility and escape routes.

Model Building

Obtaining buildings and roads

To study the factors influencing players' landing points, building and road density were needed to calculate. To obtain these variables, an analysis of the map image was conducted. Given that buildings and roads are represented in different colors than the basic land in the map, the differentiation method involved creating a new image from the map and only drawing the pixels corresponding to the colors of buildings or roads. Images for roads and buildings are presented in Figures 3 and 4. As seen, there was an issue with the names of the cities, as they were represented with almost the same color as the roads, impacting the accuracy of the road map slightly.

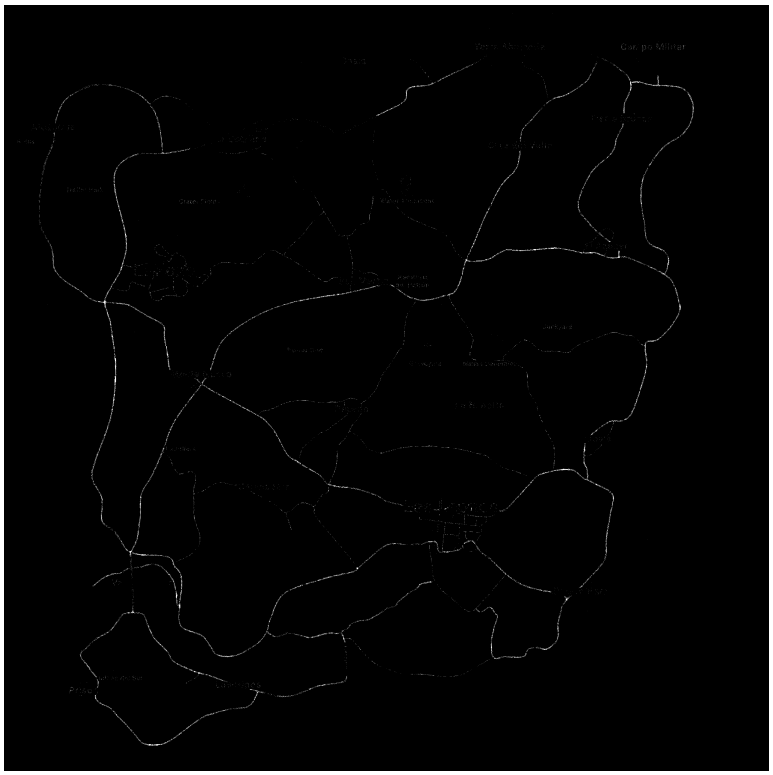

Figure 3: Roads in Miramar

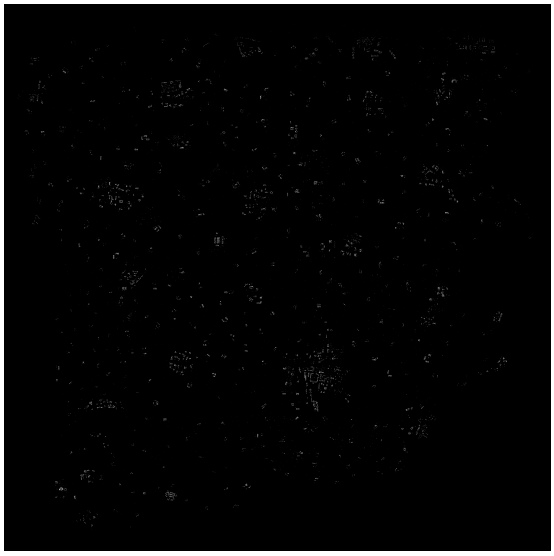

Figure 4: Buildings in Miramar

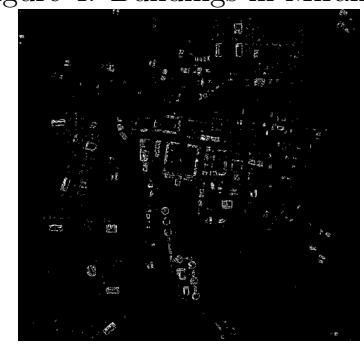

Figure 5: Closer View of a City in Miramar

Variable transformations

After that, the first step made was transforming the obtained images into a matrix, with each pixel containing the information if it's black or white, white being a building or a road.

In order to be able to work with these variables, transforming the images into matrix's wasn't enough, so after much thought the way forward, allowing us to work with all variables in the same format, was by dividing the map in a 10x10 grid.

Grid 10x10

Figure 6: Miramar map divided in a 10x10 grid

From now on, the variables will always be represented in combination with a 10x10 grid. For each square in the grid map was associated the pixels from the road and building map containing the amount of white pixels. With the grid we also introduced one more variable, distance from each square to the middle of the map trying to capture if landing closer to the center was something players preferred.

All the variables were then normalized, making possible working with percentages which is beneficial since the variables aren't all in the same scale neither represent the same units. After normalization, the target variable, which represents the number of players landing in each grid square, was categorized using intervals of 10% (0 to 10%, 10 to 20%, and so forth up to 100%). It was later noted that there is areas with absolutely no player activity being put in the same group as places with minimal activity, trying to distinguish that difference the categorization was adjusted. The first category was divided and defined as 0 to 5% and 5 to 10%, the rest of categories were still divided in intervals of 10%.

	Road_Count	Building_Count	Grid_Distance	Player_Category
(0, 0)	0.000000	0.013284	0.994589	0
(0, 1)	0.253899	0.173309	0.890699	0
(0, 2)	0.128763	0.063021	0.804112	0
(0, 3)	0.260791	0.076305	0.740921	0
(0, 4)	0.000000	0.008650	0.707423	0
...	...	...	...	...
(3, 4)	0.153065	0.268150	0.245363	2
(3, 5)	0.508886	0.176398	0.246582	0
(3, 6)	0.140733	0.487488	0.332443	5
(3, 7)	0.237577	0.050355	0.457547	0
(3, 8)	0.504534	0.119246	0.597743	0
(3, 9)	0.187885	0.063021	0.744553	0
(4, 0)	0.022851	0.170528	0.707423	1
(4, 1)	0.078346	0.133148	0.551891	0
(4, 2)	0.117156	0.168984	0.397285	0
(4, 3)	0.484947	0.446710	0.245363	9
(4, 4)	0.030831	0.086500	0.108105	4
...	...	...	...	...
(9, 5)	0.000000	0.067346	0.711648	0
(9, 6)	0.000000	0.089589	0.745760	0
(9, 7)	0.000000	0.000000	0.809314	0
(9, 8)	0.000000	0.000000	0.896067	0
(9, 9)	0.000000	0.000000	1.000000	0

Figure 7: Data frame

Correlation between the variables

To see which variables are correlated with player category, the correlations were calculated and are shown in Figure 8. From the table, it can be concluded that, as predicted before, the density of roads (road count) does not have much of a correlation with the density of landing points (player category), as the correlation is less than 0.2. On the other hand, it is shown that building density (building count) and distance from the centre (grid distance) are correlated with the density of landing points. The density of landings increases as the number of buildings increases and as the distance from the centre decreases, which can be confirmed from the correlations that are close to 0.6 and -0.6 respectively.

	Road_Count	Building_Count	Grid_Distance	Player_Category
Road_Count	1.000000	0.377503	-0.282380	0.121757
Building_Count	0.377503	1.000000	-0.533450	0.593206
Grid_Distance	-0.282380	-0.533450	1.000000	-0.586563
Player_Category	0.121757	0.593206	-0.586563	1.000000

Figure 8: Correlations of Variables in Data frame

	Road_Count	Building_Count	Grid_Distance	Player_Category
Road_Count	1.00000	0.377503	-0.282380	0.121757
Building_Count	0.377503	1.00000	-0.533450	0.593206
Grid_Distance	-0.282380	-0.533450	1.00000	-0.586563
Player_Category	0.121757	0.593206	-0.586563	1.00000

Table 1: Correlation Matrix

Statistical models

Having **Player_Category** as Y and **Road_Count**, **Building_Count**, **Grid_Distance** as X, it was applied some statistical models.

The first statistical model tested was the simplest, linear regression with a slight variation, the Ordinary Least Squares (OLS).

Linear regression models the relationship between a dependent variable and the independent variables by fitting a linear. OLS is a type of linear regression that focuses also on minimizing the sum of the squared differences between observed and predicted values.

First the summary of the model was analyzed, after the correlation value of the variable road count being low it's important to verify if in fact this variable is necessary for the model.

	coef	std err	t	P>|t|	[0.025	0.975]
const	2.2107	0.531	4.164	0.000	1.157	3.265
Road_Count	-1.0392	0.533	-1.949	0.054	-2.098	0.019
Building_Count	5.0960	1.052	4.846	0.000	3.009	7.183
Grid_Distance	-3.0544	0.679	-4.495	0.000	-4.403	-1.706

Figure 9: Linear regression (OLS) summary

With a confidence level of 95% (corresponding to a significance level of 5%), and observing a p-value of 0.054, it was decided to retain the variable in the model. The rationale behind this decision is that the p-value is only marginally higher than the threshold of 0.05, suggesting that the variable may still be relevant despite not meeting the conventional criterion for statistical significance.

These statistical methods were evaluated using **Mean Squared Error** (MSE), **R-squared** (R^2), and **Accuracy** which are common metrics to assess the

performance of models. Lower MSE indicates better model fit,R^2 measures the proportion of variance explained by the model, and Accuracy shows the percentage of correct predictions.

Parameter	**Linear Regression(OLS)**
Mean Squared Error (MSE)	1.49
R^2	0.4537322188004105
Accuracy	52.0%

Table 2: Model Performance

Linear regression results: The metrics in Table 2 show a low values of **Accuracy** and $\boldsymbol{R^2}$ paired with a high value **MSE** it was done three more models to see which one would be able to obtain better predictions.

Random Forest: An ensemble method that builds multiple decision trees for robust classification.

K-Nearest Neighbors (KNN): A simple method that classifies a sample based on the k closest data points, with k being a value chosen by the user.

Support Vector Machine (SVM): A classifier that finds the optimal hyperplane to separate different classes.

Parameter	**Random Forest**	**KNN**	**SVM**
Mean Squared Error (MSE)	0.22	1.01	1.17
R^2	0.919	0.630	0.571
Accuracy	84.0%	66.0%	74.0%

Table 3: Model Performance Comparison

The metrics presented in Table 3 highlight that the **Random Forest** model exhibits the highest **Accuracy** and $\boldsymbol{R^2}$ values, coupled with the lowest **Mean Squared Error (MSE)**. This results signifies a good predictive capability, making the **Random Forest** model the best performer among the evaluated models.

Comparison of the Best Model's Predictions with the Test Dataset

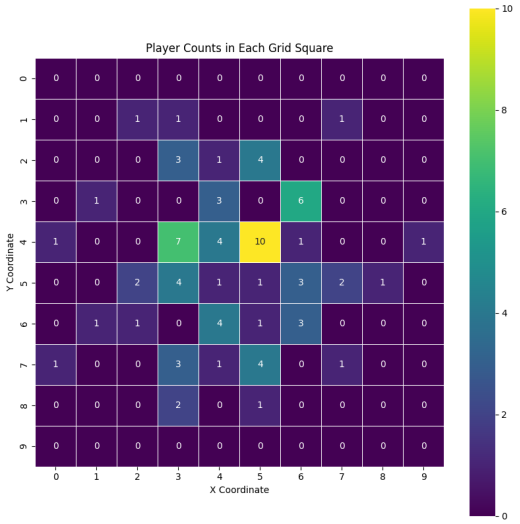

Figure 10: Heat Map of the Test Dataset

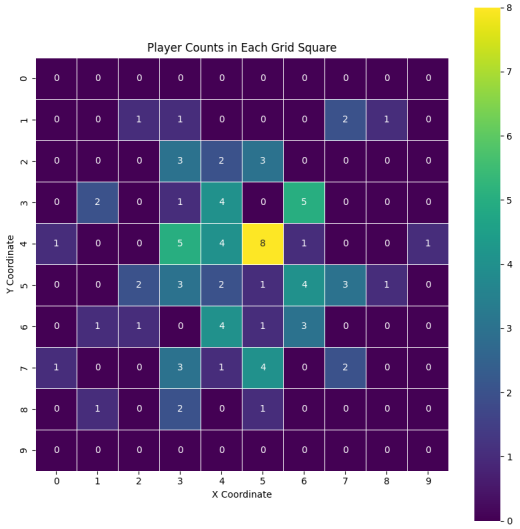

Figure 11: Heat Map of Random Forest Model Predictions

FOR LATER Neural network

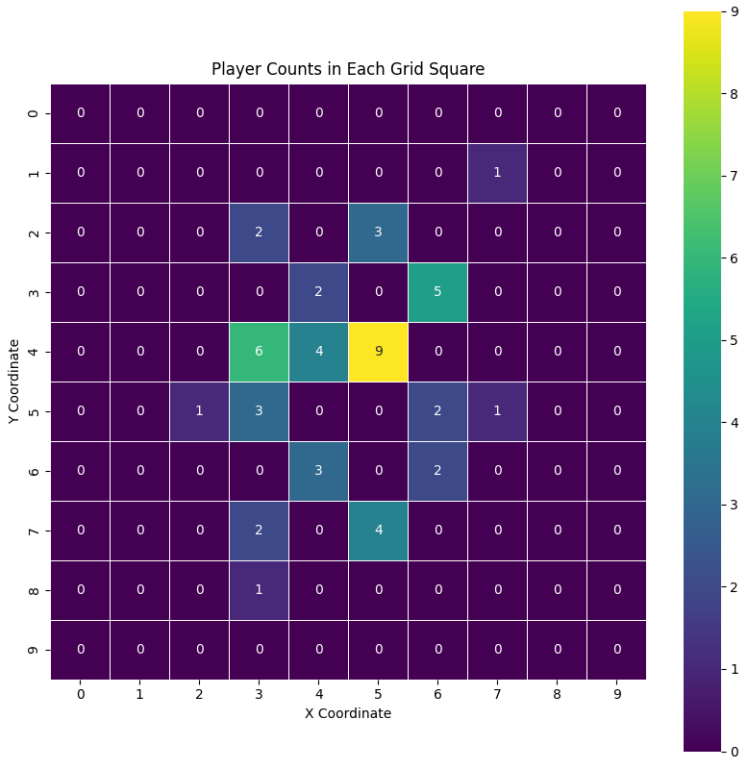

Figure 12: Data frame

Parameter	**Neural Network**
Mean Squared Error (MSE)	0.03
R^2	0.984
Accuracy	97%

Table 4: Model Performance Comparison

References

[1] Pubg: Battlegrounds. https://pubg.com/en/main. Accessed: 2023-12-11.

[2] pubg.sh. https://pubg.sh/. Accessed: 2023-12-11.