Published in International Conference on the Foundations of Digital Games, May 2024

# Solution Path Heuristics for Predicting Difficulty and Enjoyment Ratings of Roguelike Level Segment

*Colan Bieber, Seth Cooper*

https://www.pcgworkshop.com/archive/biemer2024solution.pdf

# Problem

- Procedural content generation (PCG) is incredibly common technique in games, from the structure of terrain, to the layout of enemies to the assignment of rewards

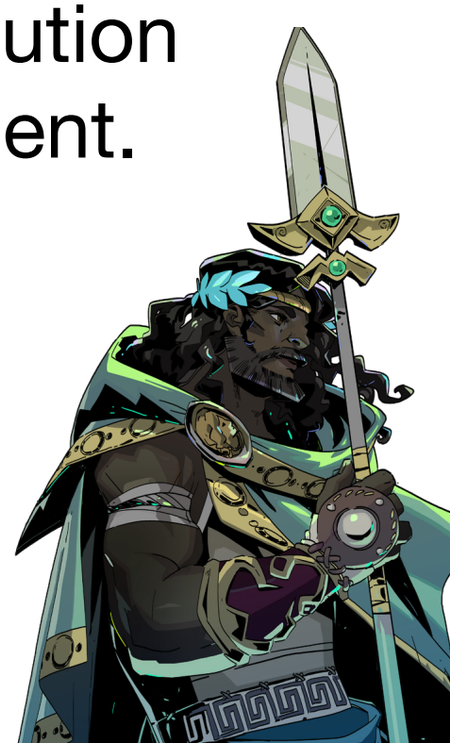
- This is none more the case in Roguelike's where to maintain playability each new run, we must randomise the levels to some degree

- These generated levels should be fun and difficult for the player

- But this poses a problem, to design a good PCG system we must have some qualitative criteria, and how do  measure something's fun-ness and difficulty?

# This Study

- Aims to provide a qualitative way to predict the fun-ness and difficulty of a level in a simple Roguelike game
- To do this, this paper pursues a relatively novel strategy of evaluating a linear regression models that takes different combinations of heuristics that are based of the solution path of the level to predict the difficulty and enjoyment.



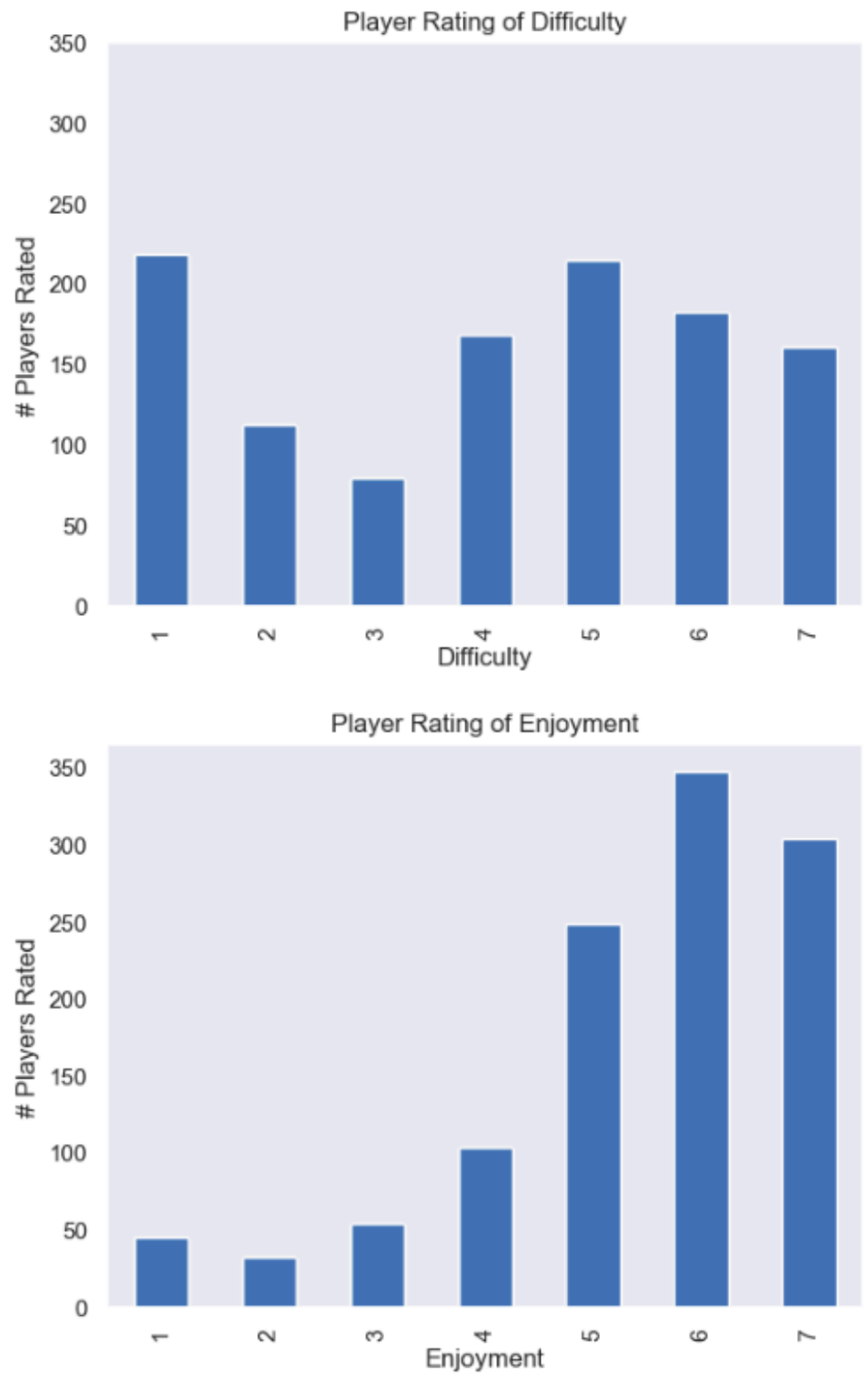
# Method

1. Get players to rate the easiness and enjoyment for a set of levels

2. Find a set of heuristics and evaluate levels

3. Input the heuristics into a linear regression model with the player ratings as the training data and conduct an ablation study

4. Find the combination of heuristics that best predict the easiness of enjoyment



# DungeonGrams

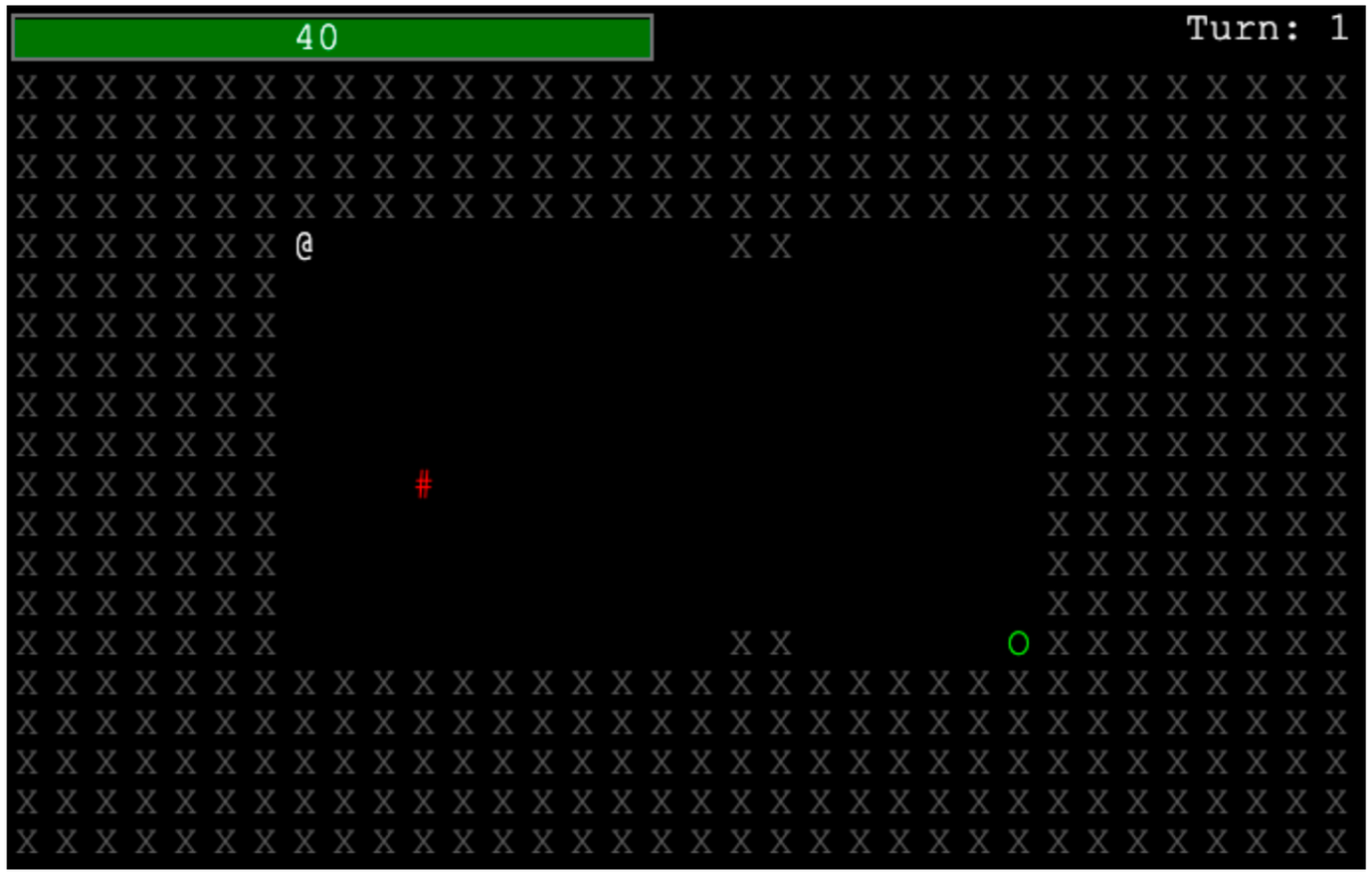


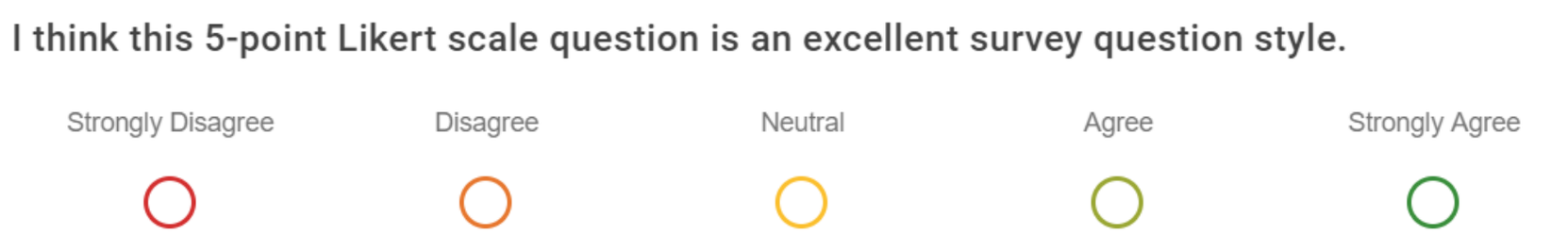

**Figure 1: Tutorial level for *DungeonGrams*. The player is represented by the '@' symbol.**

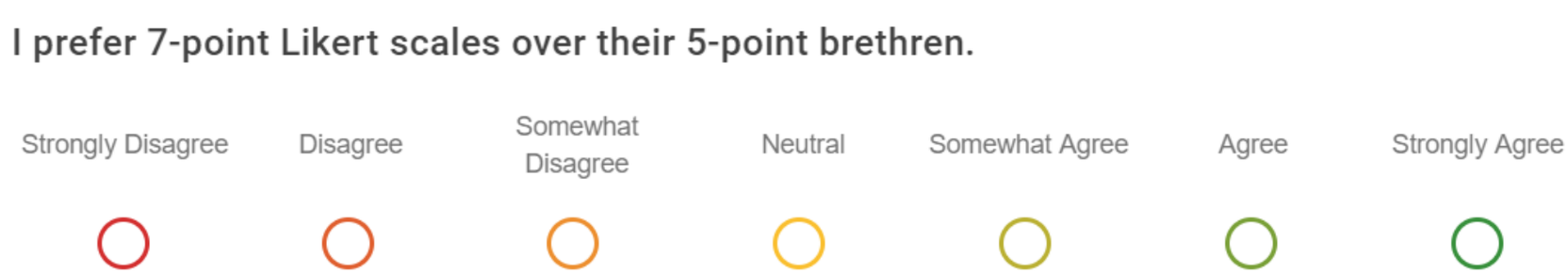
- Open source roguelike game used for research
  - '@' represents player
  - '#' represents enemy
  - '&' represents food
  - 'O' represent the portal ending the level

- The game starts with the player at the top left. A portal is at the bottom-right of the level and the goal is to unlock the portal by hitting every switch while avoiding enemies
- There is a stamina mechanic where the player starts with forty stamina and every movement costs 1 stamina. The player loses if their stamina goes down to zero or if an enemy or spike comes in contact with them.
- However, the player can gain stamina by coming into contact with food.
- It has 191 open source levels
- And a pre-implement A* search tree to find a solution!

# Player Study

- 150 people were recruited from Machine Turk, 143 actually played some levels
- For each level, whether or not they managed to beat it, players were ask to rate their enjoyment and the difficulty of the level on a 7-point Likert Scale (ie. strongly disagree to strongly agree)
- Overall players were supposed to play the tutorial level depicted on the previous slide, then 10 levels afterwards, but the average player only completed 7.9 levels
- The difficulty of the game was quite high, as 41% of players gave up on the levels they played. And the win rate for a level was only 18%
- Agreement of the difficulty was 73.4% while agreement for the enjoyment was 87.2%



# Heuristics

- **path-no-enemies** - Difference between the path length of the original level and the path length of the level with no enemies.
- **path-nothing** - Difference between the path length of the original level and the path length of the level with no enemies and no switches.
- **jaccard-no-enemies*** - Jaccard similarity of the path of the original level and the path of the level with no enemies.
- **jaccard-nothing*** - Jaccard similarity of the path of the original level and the path of the level with no enemies and no switches.
- **proximity-to-enemies [35]** - Using the completion path found for the original level, each position in the path was used to search for enemies up to three tiles away in every direction. For each enemy, one over the Manhattan distance to that enemy from the tile on the path was summed. This means that the more enemies near the optimal completion path, the larger the value of this heuristic. The sum was then divided by the length of the path.
- **proximity-to-food** - The same as proximity-to-enemies, except it searches for food rather than enemies.
- **stamina-percent-enemies** - Percent difference between the ending stamina for the tree search on the original level and the stamina left for the level with no enemies.
- **stamina-percent-nothing** - Percent difference between the ending stamina for the tree search on the original level and the stamina left for the level with no enemies and no switches.
- **density [31]** - Number of solid tiles, including spikes, divided by the total number of tiles in the level.
- **leniency [31]** - Number of enemies, spikes, and switches divided by the total number of tiles in the level.
- **food-density** - Number of food tiles divided by the total number of tiles in the level.

$$\text{JaccardSimilarity(A,B)} = \frac{|A \cap B|}{|A \cup B|}$$

- Jaccard Similarity was used to create novel heuristics for this application
- In the case of the JS between two paths it is the number of common points between two paths divided by the number of unique points for both paths
- It helps determine between paths of the same length, that path length, capturing more information.
- [31] and [35] are similar to heuristics in cited papers

# Training

- Heuristic values for levels were dataset with median player-rated difficulty and median player-rated enjoyment.

- Median instead of the mean because it better represents the ordinal data from a Likert survey

- Trained using k-fold cross-validation, where $k=5$, and minimised the mean square error (MSE) with a train-test split of 0.8

- **Ablation** study was run using the training portion of the data

- The study then presents the best 10 combinations out of 2,047 potential combinations, which are compared against the baseline value (the mean of difficulty and enjoyment)

| Heuristic | Difficulty | Enjoyment |
|---|---|---|
| *density* | **1.0** | 0.0 |
| *food-density* | 0.0 | 0.0 |
| *jaccard-no-enemies* | 0.0 | **0.5** |
| *jaccard-nothing* | **1.0** | 0.0 |
| *leniency* | 0.2 | 0.0 |
| *path-no-enemies* | 0.0 | 0.0 |
| *path-nothing* | 0.4 | **0.8** |
| *proximity-to-enemies* | **1.0** | **1.0** |
| *proximity-to-food* | 0.2 | **0.4** |
| *stamina-percent-nothing* | 0.2 | **0.4** |
| *stamina-percent-enemies* | **0.5** | 0.0 |

**Table 1: Percent heuristic usage for the top-ten best performing heuristic combinations for predicting difficulty and enjoyment. The top four heuristics for both are bolded. Note the tie for *proximity-to-food* and *stamina-percent-nothing* for predicting enjoyment.**

# Difficulty Results

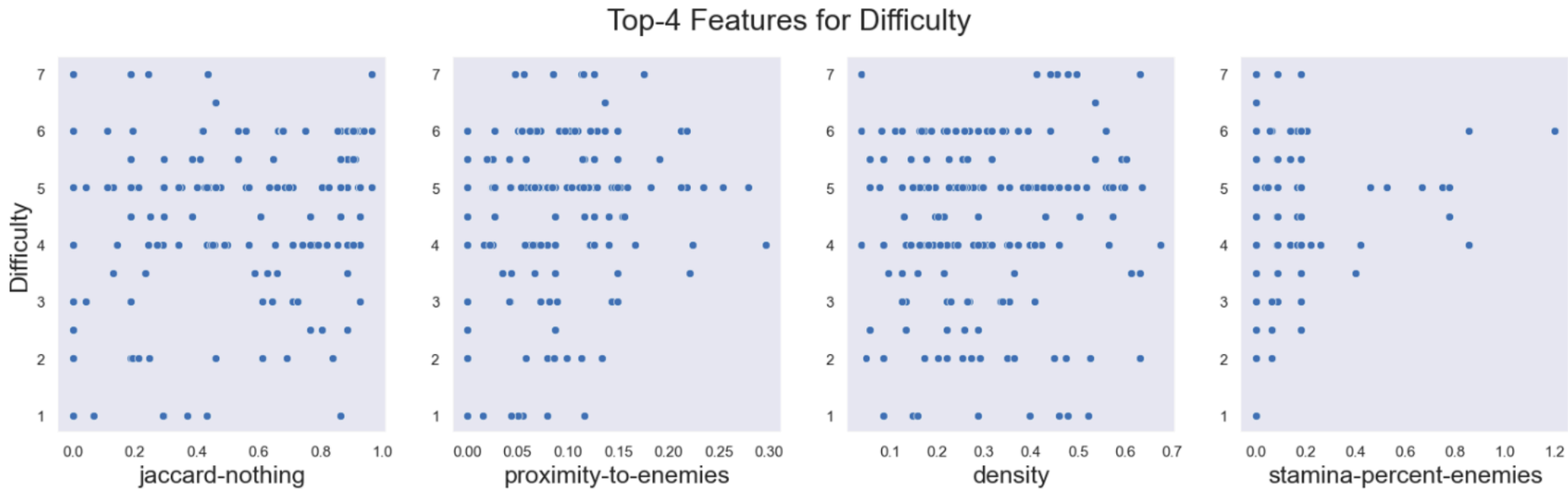



**Figure 3: Top four features used to predict difficulty.**

No vertical lines :)

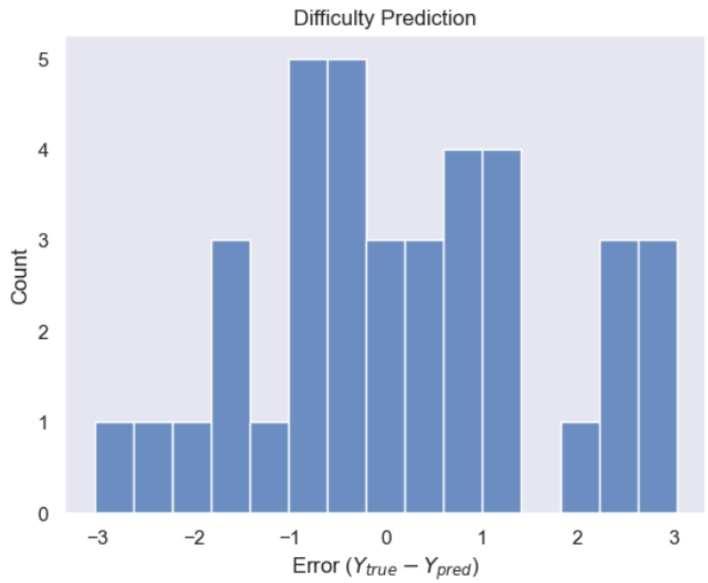



**Figure 4: Difficulty prediction histogram for $Y_{true} - Y_{pred}$.**

# Difficulty Evaluation

- The best combination of heuristics was **jaccard-nothing**, **proximity-to-enemies**, **stamina-percent-enemies**, and **density**. First 3 were used in all top 10 models.

- A linear regression model using best 4 was tested on the test set and found that the median square error was 0.96, and the max square error was 8.68.

- A value of less than one for the median indicating that the model was generally close to the user's rating of difficulty.

- The baseline is the mean difficulty from the player study. When run on the test set, it had a median square error of 1.27 and a max square error of 5.80. The linear regression model had a lower median square error but a higher max square error.

# Enjoyment Results

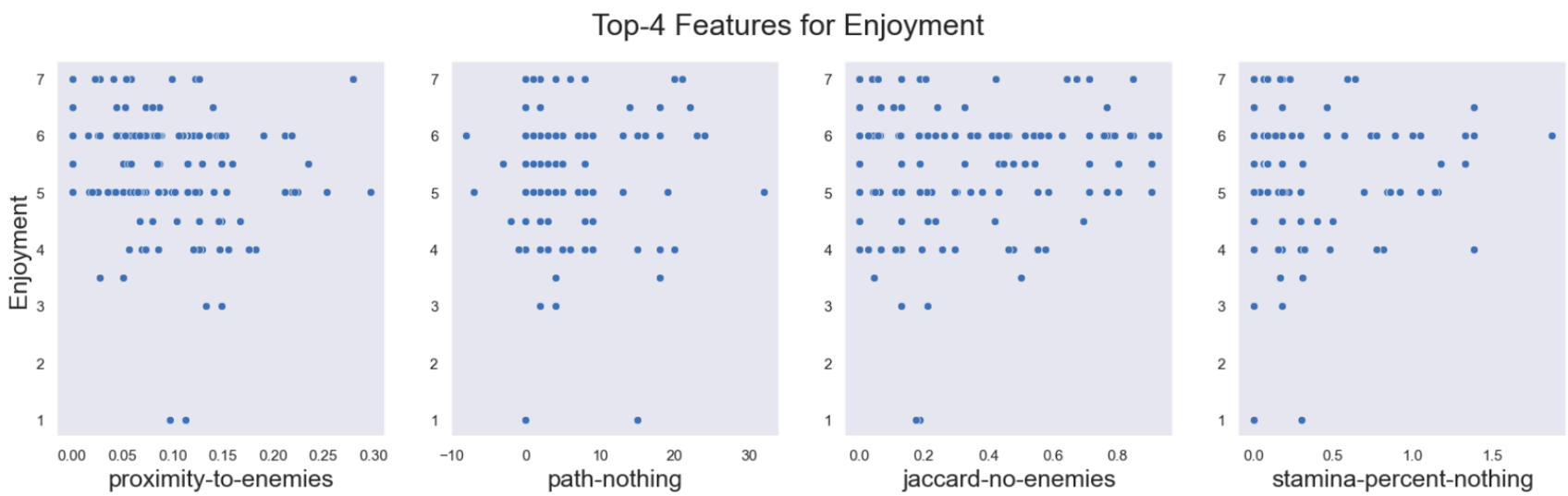



**Figure 5: Top four features used to predict enjoyment.**

Vertical lines :(

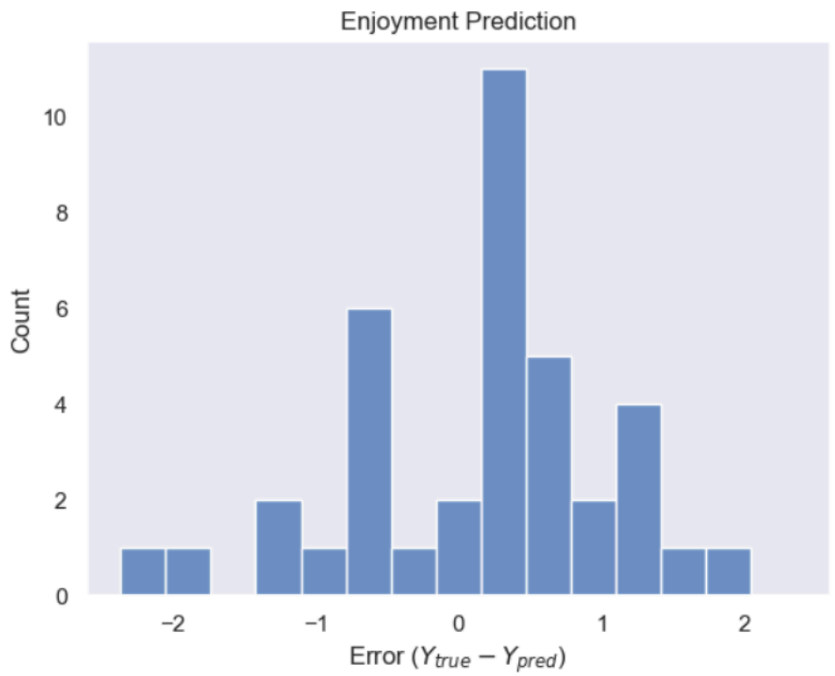



**Figure 6: Enjoyment prediction histogram for $Y_{true} - Y_{pred}$.**

# Enjoyment Evaluation

- The best heuristics was **proximity-to-enemies**, while the second best was **path-nothing**. Otherwise only two other heuristics were used to predict enjoyment: **proximity-to-food** and **stamina-percent-nothing**.

- A linear regression model using best 2 was tested on the test set and found that the median square error was 0.79, and the max square error was 5.27. This indicates that the model was more easily able to predict enjoyment than difficulty.

- But the baseline is the mean enjoyment from the player study. When run on the test set, it had a median square error of 0.70 and a max square error of 2.84. Showing the baseline outperformed the models in both cases.

- This suggests that Enjoyment is just easier to predict, this is enforced by the distribution of the responses.

- It is also interesting that the consensus for enjoyment is so high among participants.

# Coloration between difficulty and enjoyment

- Despite past works suggesting a positive correlation for other games, the paper actually find no correlation between difficulty and enjoyment.

- Pearson correlation coefficient was -0.083 with a p-value of 0.005, the Kendall rank correlation coefficient was -0.122 with a p-value of 0.033, and the Spearman rank correlation coefficient -0.194 with a p-value of 0.007.

- The consistently low p-values shows that this lack of correlation is statistically significant.

- This could be for a variety of reasons such as the nature of the game and the difficulty it contains.

- The diagram shows this dichotomy subjectively –>

- Difficulty brings lack of choice.

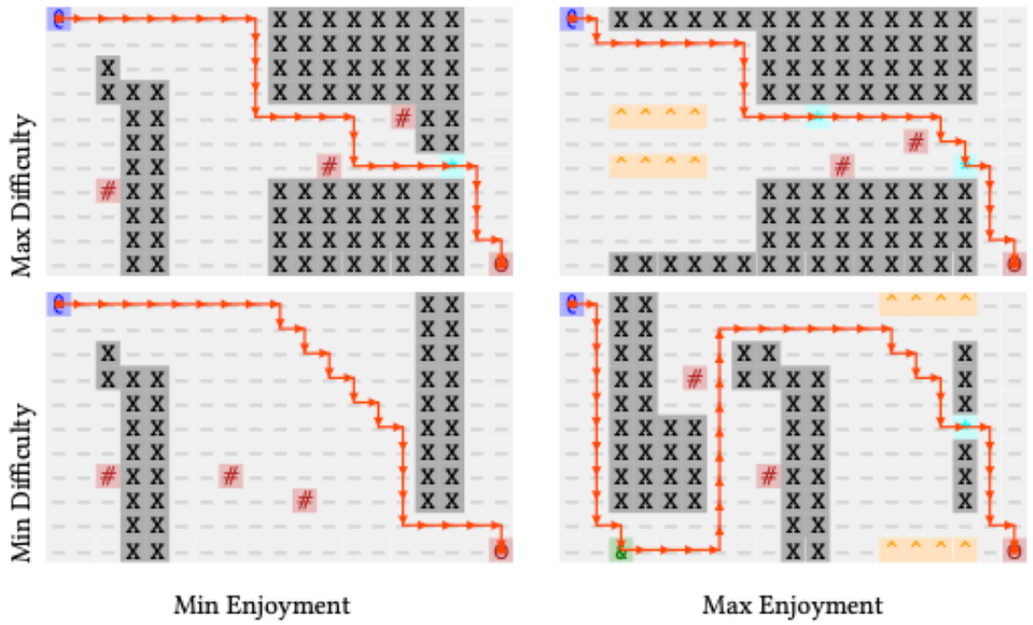



**Figure 7: Example levels for min and max of difficulty and enjoyment based on mean ratings from study participants. Red arrows show path found by the A* agent to complete the level.**

# Novelty of Method

- Overall the top performing model found can predict difficulty model which can predict and enjoyment within one point on the Likert scale based on the median square error. However, the model for predicting enjoyment did not beat our baseline approach.

- While some previous research used linear regression models to determine difficulty, some didn’t used heuristics or only one heuristic, some only used input heuristics instead of path heuristics, some used time as the value for difficulty, some didn’t have player validation.

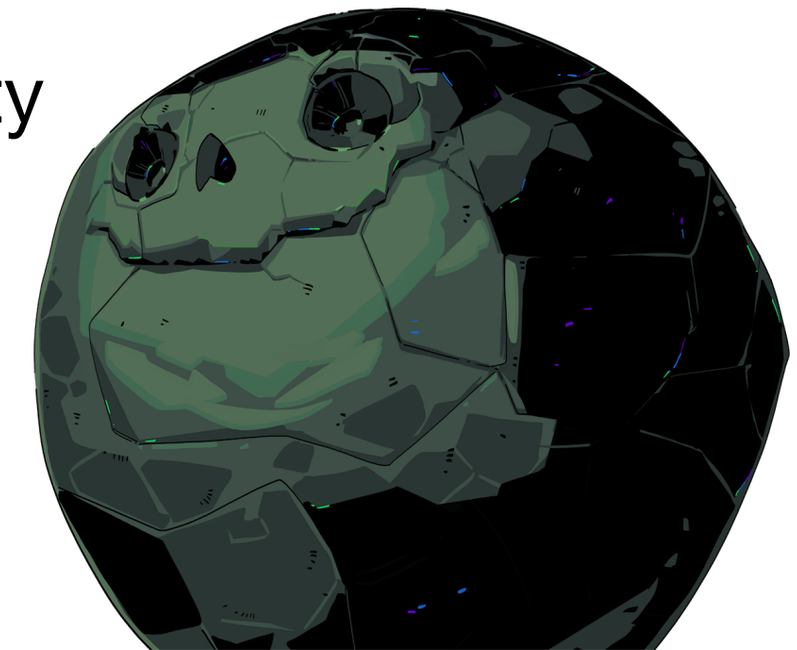
- Ultimately while not completely novel, the combination of – path heuristics, combination of heuristics and validation by difficulty ratings from Player Study – is novel.

- Additionally applying the same methods to enjoyment in completely novel.

- Despite the heuristics it produced’s effectiveness, the use of Jaccard’s Similarity is completely novel in the field of games and difficulty evaluation

# Limitations

- Very simple game
- The bounds of the linear regression problem
- Didn’t take into account the learning curve of player beyond tutorial level
- Limited number of possible solutions
- Doesn’t model other complexities of difficulty



# Conclusion

**Thank you for listening!**