Introduction

Abstract

This project seeks to optimize qw’s berserk logic on the first floor of the dungeon. Figure 1 depicts the game.

Berserk Logic

- Berserking is essential for survival in some cases, but has serious - sometimes lethal - drawbacks
- qw’s logic for berserking is numerical and well suited for computation:
  - if player.level < monster.scareness, then berserk

First Floor

Small set of possible monsters to encounter (≈12)
- Completing the first floor of the dungeon takes only a few seconds, enabling generation of large amounts of data and faster validation
- Considered one of the more interesting and challenging segments of the game for human players

Analysis

Statistical analysis of results was conducted with Chi-squared testing at a significance level equal to 0.05.
- The first SGD model was 1.01% worse than the default behavior with P = 0.0141.
- The second SGD model was 0.88% worse than the default behavior with P < 0.0001.
- The third SGD model is similar to the default behavior; it is 0.17% smaller, but with P = 0.2404, we cannot reject the null hypothesis.
- The LogisticRegressionCV model performed 0.08% higher than the default behavior, but P = 0.6186. Therefore, the model does not show a statistically significant difference in performance from the default behavior.

Discussion

• The attempted optimizations of qw’s berserk logic failed to improve the success of human berserkers in traversing the first floor of the dungeon.
- Initial testing with a weaker character race, Deep Elves, shows promise that the LogisticRegressionCV model can indeed create significant performance improvements for the focused optimization.
- Deep Elf testing showed a 2.16% improvement in success rate with P = 0.0028, showing statistically significant improvement.
- Under default behavior, Deep Elves succeeded at a rate of 86.86%, much less than the Human rate of 96.05%.
- This indicates that such optimization methods may be more impactful when initial rates of success are lower.

Future Work

• Further tuning of hyperparameters could improve the results of SGD modeling.
- Creating models from data based on weaker character races shows promise based on initial Deep Elf analysis.
- This sort of parameter tuning could be performed for different parts of qw, optimizing progression through more of the game than the first floor.
- These techniques also have potential applications for software optimization in fields unrelated to gaming, such as search and rescue drone navigation.

Software Components

DCSS - open source turn-based game with procedurally generated content and permanent character death
qw - Expert AI hand coded to beat crawl with victory rate of approximately 17%
Scikit-learn - open source, python based machine learning toolkit used computer optimum values for qw’s parameters
Bash - used to run crawl and aggregate data

Hypothesis

Utilizing machine learning techniques, a more optimal set of weights for the danger level of monsters in DCSS can be ascertained so as to improve the ability of qw to succeed in clearing the first floor of the dungeon.

Results

Machine learning models were generated using the collected data as input. The first three models were generated with Scikit-learn’s Stochastic Gradient Descent (SGD) classifier, and another model was generated with Scikit-learn’s Logistic Regression Cross Validation classifier. After creating models, every possible permutation of monster scareness weights was tested and the highest probability weight for each model was selected for trials. Table 2 contains the results of these trials.

<table>
<thead>
<tr>
<th>Experiment</th>
<th>Trials</th>
<th>Success Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default Settings</td>
<td>29,717</td>
<td>96.05%</td>
</tr>
<tr>
<td>SGD_1</td>
<td>2,474</td>
<td>95.04%</td>
</tr>
<tr>
<td>SGD_2</td>
<td>49,394</td>
<td>95.17%</td>
</tr>
<tr>
<td>SGD_3</td>
<td>49,451</td>
<td>95.88%</td>
</tr>
<tr>
<td>LogisticRegressionCV</td>
<td>28,510</td>
<td>96.13%</td>
</tr>
</tbody>
</table>

Future Work

• Further tuning of hyperparameters could improve the results of SGD modeling.
- Creating models from data based on weaker character races shows promise based on initial Deep Elf analysis.
- This sort of parameter tuning could be performed for different parts of qw, optimizing progression through more of the game than the first floor.
- These techniques also have potential applications for software optimization in fields unrelated to gaming, such as search and rescue drone navigation.

Acknowledgements

Thanks to Dr. David Schwab for his motivational and informative advising. Dr. Surfy Jiang for her constant guidance and leadership, and Dr. Jose Ignacio Pando for his inspiration and encouragement. Additional thanks to all of my peers in the computer science department and to the DCSS community, including the #Kraken and #Berserk-dev IRC channels. Special thanks to Sigmund for staying out of the first floor of the dungeon.