Chapter 4 - Frontiers of GVGAI Planning

Exercises

The GVGAI Framework is available in a Github repository[1]. Use the release 2.3[2] in order to run the same version presented in this chapter. Models and data are for the Win Prediction work are also available at a Github repository[3] and you can use a specific checkpoint for the version presented here[4,5].

1 Current Problems in GVGAI Planning

Section **??** presents the main challenges for GVGAI planning at the moment. Interesting projects can arise from trying to tackle the following points.

- Compute game-related features that use an efficient but more accurate measure than Euclidean distance, in order to improve existing methods with them.
- Modify a successful algorithm so it uses a more abstract set of actions, either macro-actions or policies that aim for a higher level objectives (i.e. move the avatar to a specific location, avoid an enemy, collect certain items, etc).
- Implement an heuristic that is able to switch between different methods depending on certain game-based features. The heuristic should detect and switch to the most appropriate algorithm to play a given game in real time.

2 General Win Prediction

The work presented in Section 4 can be enhanced in different projects.

- Determine new features to use for the prediction models. Do they improve the algorithms that foresee the outcome of the game? Would they be game- or agent-based features?

[1] `https://github.com/GAIGResearch/GVGAI`
[2] `https://github.com/GAIGResearch/GVGAI/releases/tag/2.3`
[3] `https://github.com/rdgain/ExperimentData/tree/GeneralWinPred-CIG-18/`
[4] `https://github.com/rdgain/ExperimentData/commit/dc354e6047e378833ef852d0a053aa9215cc6a6b`
[5] These exercises are also available at this book's website: `https://gaigresearch.github.io/gvgaibook/`

- Create a similar system to the one presented here that predicts which algorithm would play better the current game, based on agent-based features only. This new system, in combination with the former one, would be able to establish which algorithm the agent should switch to once the outcome predictions come negative.
- Use different classifiers to train the models. It would be interesting to find out if using deep learning methods provides better predictions than the ones presented here.