
Chapter 7 - Automatic General Game Tuning

Exercises

The GVGAI Framework is available in a Github repository¹. Use the release 2.3² in order to run the same version presented in this chapter. The NTBEA algorithm can be found at its Github repository³ and you can use the checkpoint ⁴ for a stable version of the code. Additionally, you can also find a Python version of NTBEA at another Github repository^{5,6}.

1 Parameterizing VGDG Games

VGDG game spaces can be found in the `examples/gameDesign` folder of the GVGAI repository.

1.1 Editing a Game Space

You can play-test parameterized games running the class `tracks.gameDesign.TestGameSpace`. Take one of the games in `examples/gameDesign` and open the VGDG description. Study how modifying the values of the parameters change the game when you play it.

1.2 Parameterize a VGDG game

Take any game from `examples/gridphysics`, `examples/contphysics` or `examples/2player` and copy it to `examples/gameDesign`. Then, parameterize this game considering the possible values the parameters may have. Remember to do the following.

- Change the game type (first line) from `BasicGame` to `GameSpace`.
- Add the set `ParameterSpace`, in which you will define all parameters that you want to add to the game.

¹ <https://github.com/GAIGResearch/GVGAI>

² <https://github.com/GAIGResearch/GVGAI/releases/tag/2.3>

³ <https://github.com/SimonLucas/ntbea>

⁴ <https://github.com/SimonLucas/ntbea/commit/2bb178d5ea57c4219b12a49aea32d59ed596612a>

⁵ <https://github.com/bam4d/NTBEA>

⁶ These exercises are also available at this book's website: <https://gaigresearch.github.io/gvgaibook/>

- For each boolean, integer or floating point value available in the `SpriteSet`, `InteractionSet` and `TerminationSet`, you may add a new parameter to the game space. Think about the impact that it has in the game, and define feasible minimum and maximum values for the parameter, as well as possible intermediate values defining an increment step.

2 Optimize VGDL Games

Use NTBEA to tweak a parameterized VGDL game so it behaves in a determined manner. Examples are as follows.

- The game should be easier to complete for high skilled agents than low skilled ones.
- The game should provide a determined player experience. Score trends was studied in [1], but other options are possible: feel of danger, number of different solutions, pace, etc.
- Evolve game parameters so they provide a specific *bot* experience. For this, you may look at the agent experience features described in Chapter 4. The objective game configuration should aim at providing certain values for these features (i.e. reducing the decisiveness of the agent).

You can also tweak the NTBEA parameters and observe if in any of the above scenarios the optimization converges faster to a desirable solution.

References

1. K. Kuanusont, S. M. Lucas, and D. Perez-Liebana, “Modelling Player Experience with the N-Tuple Bandit Evolutionary Algorithm,” in *Artificial intelligence and Interactive Digital Entertainment (AI-IDE)*, 2018.