Computer Language Concepts

Oz project - Captain Sonar

Antoine Rime & Youri Mouton - Group 98

Implementation details

Main

We implemented both TurnByTurn and Simultaneous gameplays. They support all the required features. They've been thoroughly tested with the AI player provided on Moodle and ours.

Difficulties

The project was a bit difficult at first because we didn't really know what to implement but the teacher's AI allowed us to understand better how the game should behave, which also made the developement of AIs much easier.

Most difficulties we've had are due to the obscurity of Oz: we couldn't find answers to basic problems. In the end we managed to solve most of our issues by rethinking our design to fit de functional paradigm and thus learning a great deal about programming in general.

The resulting Main.oz is pretty concise - it relies heavily on pattern matching with the `case` statement for conditionals.

We had a hard time finding ideas on how to implement the AIs so we were helped by other groups.

AI

SeekAndDestroy

Our first AI first gathers information about the other players then starts chasing them. It is a rather offensive player. When the player is in `seek` mode it will chare sonars and it will charge missiles in `destroy` mode.

Dumb

Our second AI is, as the name implies, dumb. Most of it's operations are randomly decided.

Interoperability Tests

We tested the following project team's AI with success:

- Samuel Monroe, Rémy Voet (Groupe 31)
- Antoine Lambot, Nicolas Vanvyve (Groupe 10)

Additional features

We implemented a random map generator in Input.oz.