

Team Condor

Authors: F. Berti, D. Campodonico, J. Ferreras, E. Kira, A. Kohan, L. Moretti

Teachers: M. Gutierrez, J. Roa. *Artificial Intelligence, 2017.*

Universidad Tecnológica Nacional, Facultad Regional Santa Fe.

Introduction

Inspired in the propositions of the situation calculus, we've designed and developed our very own Rule Engine. Each rule has matching conditions (or *situations*) to decide whether it's considered or discarded for the current shot. To decide among all the currently considered rules, each rule also contains a numerical weight based on our analysis on their potential impact.

Technology

We considered functional programming to be an ideal paradigm to build the Agent with, so we've used Kotlin as the programming language, seizing the opportunity provided with the basic game playing software being implemented in Java (fully compatible with Kotlin).

Custom algorithms

- **Await Still Scene**

We improved the overall performance by enhancing the detection of movement across the stage when waiting to plan a shot.

- **Path finder**

Within a relative range from the

default trajectory, it decides if it should re-adjust the target point for a better result.

Strategies

- **Structure Rules**

We detect the shapes of the structures (two or more blocks connected) and classify them on:

- If **Fortress**: shoot on the top left point to knock it over the pigs.
- If **Lookout**: shoot on the middle point for maximum impact.

- **Boulder Rule**

If there's a round block next to pigs (f.e. on a hill), shoot the block.

- **TNT Rule**

If there's a TNT block it shoots it.

- **Bird Rules**

Decide what type of block to hit based on the current bird.

- **Alone Pig Rule**

If there is a reachable pig alone, shoot it.