

TX52 - scientific project report

Create a Real-Time Strategy Game Engine with
SARL, an agent oriented language

Olivier Villequey, Romain Dulieu

April 18, 2016

Contents

1	Project Definition	5
1.1	Introduction	5
1.2	State of the art	5
1.2.1	RTS history	6
1.2.2	Environment	6
1.2.3	Path Finding	7
1.2.4	AIs in RTS	7

Chapter 1

Project Definition

1.1 Introduction

The subject of our project is to create a Real-Time Strategy (RTS) game engine using SARL language, an agent oriented language based on java. Firstly, we have to define what is a RTS game and what are the important specificities or components of this kind of games :

- * Real-time strategy : include an enormous number of possible game actions that can be executed at any given time. The result or effect of any action is unknown and changes as the time goes by. Furthurmore, RTS games usually manipulate a large number of units, controlled by an AI, which is an important aspect of our project.
- * We consider that RTS games include only partially obervable environments. It means that one aspect of the strategy is to collect informations by exploring the environment.
- * A map : an environment composed of different objects static (walls, trees, cliffs, etc.) or dynamic (units) with specific 2D/3D environment features such as the topography etc.
- * Different types of AIs : strategic level or global AI that takes abstract decision making to achieve the main goal : deciding the size of the force necessary to move/attack a position, the kind of forces needed etc. On the other hand, the tactical level or local AI concerns more concrete decisions (handle the actions : moving,attacking,producing etc.) and is responsible for achieving the objectives defined at the strategic level.

1.2 State of the art

First of all, as the gaming tools and games are developped in private companies it is difficult to get recent public papers about the actual technologies used

in RTS games. That’s why our sources may be a little bit old but it is easier to find public sources from a couple of years ago.

1.2.1 RTS history

Real-time strategy (RTS) games are known to be one of the most complex game genres for humans to play, as well as one of the most difficult games for computer AI agents to play well. To tackle the task of applying AI to RTS games, recent techniques have focused on a divide-and-conquer approach, splitting the game into strategic components, and developing separate systems to solve each. This trend gives rise to a new problem: how to tie these systems together into a functional real-time strategy game playing agent.

Traditional games such as Chess and Go have for centuries been regarded as the most strategically difficult games to play at a top level. High-level play involves complex strategic decisions based on knowledge obtained through study and training, combined with online analysis of the pieces on the current board. Top players are able to “look ahead” a dozen or more moves into the future to decide on an action, often under strict time constraints, with clocks for each player ticking away as they think make their decision. Let us now imagine a genre of game in which the playing field is 256 times as large, contains up to several hundred pieces per player, with pieces able to be created or destroyed at any moment. On top of this, players may move any number of pieces simultaneously in real-time, with the only limit being their own dexterity. What we have just described is a real-time strategy (RTS) game, which combines the complex strategic elements of traditional games with the real-time actions of a modern video game.

A relatively new genre, the first RTS games started to appear in the early 1990s with titles such as Dune II, WarCraft, and Command and Conquer. Originally introduced as a single-player war simulation, their popularity exploded as the internet allowed for players to compete against each other in multiplayer scenarios. With the creation of StarCraft in 1998, RTS games had reached a level of strategy unseen in other video game genres.

1.2.2 Environment

RTS games take place on a map, composed of a finite or infinite number of cells with a position organised in a grid. On this basis, you can choose to create a continuous or discontinuous environment space. Most of the RTS are still based on a two-dimensional map even in 3D engines. In fact, newer games didn’t innovate much on the initial concept but tend to emphasize more on the basic RTS elements such as higher unit cap, more unit types, larger maps, etc. Environments can implement climate changes, different types of terrain that impact certain types of movement etc.

1.2.3 Path Finding

1.2.4 AIs in RTS

Inspired by military command structures, tasks are partitioned among modules by their intuitive strategic meaning (combat, economy, etc.), with vertical communication being performed on a “need to know” basis. High level strategy decisions are made by the global AI by compiling all known information about the current game state. Commands are then given to local AI which are directly in charge of completing the low-level task.

References

Incorporating Search Algorithms into RTS Game Agents, David Churchill and Michael Buro