 ISBN: 978-989-20-2190-4

Interface Genérica para Desenvolvimento de Jogos de Estratégia Abstractos

Generic Interface for Developing Abstract Strategy Games

Ivo Paz dos Reis[1]

Luís Paulo Reis[2]

[1,2]**FEUP – Faculty of Engineering of the University of Porto, Portugal;** [2]**LIACC – Artificial Intelligence and Computer Science Lab., University of Porto, Portugal**

{[1]**ei05021,** [2]**lpreis}@fe.up,pt**

Resumo

Os jogos de tabuleiro fazem parte das sociedades humanas desde o seu início, sendo excelentes ferramentas para transmitir conhecimentos, expressar ideais culturais e treinar processos de pensamento. Compreender a sua importância cultural e intelectual não é apenas uma parte importante dos estudos antropológicos, mas também um catalisador fundamental para avanços em ciências da computação.

Em ciências da computação, e em particular no campo da Inteligência Artificial, o estudo de jogos de tabuleiro oferece maneiras de compreender as relações entre os conjuntos de regras e estratégias de jogo. Como os jogos de tabuleiro abrangem uma quantidade enorme de diferentes tipos de mecânica de jogo, o foco deve ser colocado em jogos de estratégia abstractos para limitar a pesquisa efectuada e para contornar problemas derivados da incerteza.

A plataforma de General Game Playing *"Zillions of Games*" é capaz de analisar as regras e jogar qualquer jogo de estratégia abstracto, desde que seja escrito na linguagem orientada a jogos do *Zillions*, a *ZRF*.

O objetivo deste trabalho é criar uma interface genérica para facilitar a criação jogos de estratégia abstractos na linguagem do *Zillions of Games,* sem a necessidade de qualquer conhecimento de programação. Para tornar isto possível é necessário uma extensa investigação sobre a mecânica de jogos abstractos, com a finalidade de classificar e categorizar os diferentes tipos de jogos, regras e estratégias.

Palavras- chave: Jogos de Estratégia Abstractos, General Game Playing, Inteligência Artificial, Geração de Código

Abstract

Board games have been a part of human societies since the beginning, being excellent tools for conveying knowledge, express cultural ideals and train thought processes. To understand their cultural and intellectual significance is not only an important part of anthropological studies but also a pivotal catalyst in advances in computer sciences.

In computer sciences, and particularly in the field of Artificial Intelligence, the study of board games provides ways to understand the relations between rule sets and playing strategies. As board games encompass an enormous amount of different types of game mechanics, the focus shall be put on abstract strategy games to narrow the research and to circumvent the problems derived from uncertainty.

The General Game Playing platform "Zillions of Games" is capable of analysing the rules and play any Abstract Strategy Game, provided it is written in Zillions' own game-oriented programming language, ZRF.

The aim of this work is to create a generic interface for facilitating the creation of Abstract Strategy Games in the Zillions of Games language, without the need for any knowledge of programming languages. This can only be made possible by an extensive research on Abstract Strategy Games mechanics, in order to classify and categorize different game types, rules and strategies.

Keywords: Abstract Strategy Games, General Game Playing, Artificial Intelligence, Code Generation

1. Introduction

Board game study plays an essential role in computer sciences, especially in the progress of artificial intelligence (AI), as they are noise-free, cheap and easily replicable environments where the concepts of tactics, strategy, searching, learning and other relevant features of AI can be tested in their purest state. Board games, due to their conceptual flexibility, allow an open testing field for many different ideas concerning the AI approach [4].

The idea of having an application designed for game creation without the need of having any particular programming skills is not new. There are already some platforms that allow users to do just that, mostly for platform or role playing games. However, research in General Game Playing Systems specifically designed for Abstract Strategy Games (ASG) showed that no such applications exist [2].

There are projects concerned with generic ASG creation, like the LUDÆ or the Ludi projects, but their aim differs from the user-choice creation method of this project. The goal of the LUDÆ project is to create an arena populated by AI agents and have them play different ASG, evolving the agents in subsequent iterations using genetic algorithms [4]. The Ludi project concerns itself with the automatic generation of human-friendly ASG, using evolutionary algorithms to generate games and then applying a series of game evaluating functions, measuring aesthetics, depth, drama, and so on [1].

The capabilities of the General Game Playing platform "Zillions of Games" (ZoG) are remarkable, since they allow the creation of any abstract game via the ZRF language, and the application is then able to play the games applying a powerful rule analyzing AI engine.

This project is focused on the research of AI techniques and methods applied to ASG. The main goal of the project is to create a generic interface that supports design and specification of most types of ASG. The breakdown of the goals of the project is as follows:

- Research on ASG characteristics, focusing on similarities between rules, boards and pieces
- Develop a ZRF code-generating platform that enables users to create their own ASG

The rest of the paper is as follows. Section 2 describes how ASG are defined. Section 3 focuses on explaining the ZoG platform. Section 4 describes the paths and options taken during the implementation of the project. The final section contains an overview of the future iterations of the project as well as some of the conclusions attained so far.

2. Abstract Strategy Games

A key point in ASG is perfect information, which means that all players have complete knowledge of all the information of the game at all times, there is no hidden information like

hidden cards or dice rolls. This requirement makes ASG an excellent research subject for computer sciences and especially AI, since the only constraints are combinatorial in nature [4]. Other characteristics of ASG are the fact that they are built on simple and straightforward design and mechanics, and that they should also promote one player overtaking their opponent by a skillful combination of tactics and strategy [6].

ASG include an astounding variety of games, each with specific sets of rules and strategies. In order to describe the game mechanics in a general manner it is necessary to broaden the scope of the analysis, pinpointing the most usual similarities between rule sets without restricting the possibility of rules not covered by the generalization [2]. Although there is no consensus in this matter, the official classification of the International Abstract Games Organization was selected for its completeness [3]. This classification is also loosely based in the ZoG commercial application. It follows the principles of rule analysis for describing games, focusing on objectives for victory. Another way to categorize ASG is the evaluation of the features present in the games, such as piece mobility or board types [5]. The following table, adapted from the different sources described above, summarizes the classification of ASG used for this project:

ASG Categorization by			
Features		**Goals**	
Board Types	Square Hexagonal 1-D Board	***Escape***	Race Breakthrough
Piece Movements	Drops Slides Jumps Swaps Captures	***Positioning***	Connection Pattern
		Territory	Claim Occupy
		Elimination	Royal (capture / checkmate) Multi-Piece (capture / stalemate)

Table 1 – ASG Categorization by Features and Goals

3. Zillions of Games General Game Playing System

A General Game Playing System is a system that can accept a formal description of a game and play the game effectively without human intervention [2]. For this project it was necessary to choose a General Game Playing System with an advanced AI engine. The ZoG platform was the obvious choice, as it is the most successful application of this kind for running ASG. Its promotional tagline is *"The first infinitely expandable PC gaming system."* Developed by Jeff Mallett and Mark Lefler in 1998, ZoG is a game package with a 'universal gaming engine' for ASG. This allows players to play nearly any abstract board game. This engine takes as input games written in Zillions Rule Language (ZRF), specifically written for board games. After parsing the game rules, the system's AI can play the game automatically, treating all games as solitaire puzzles (ZoG, 2009).

4. Implementation

The abstract game generic creation application works as an intermediate step between users and the ZRF language. The idea is to present users with a multitude of choices in a visual interface and then translate these choices, herein the new game itself, to a ZRF format readable by the ZoG platform.

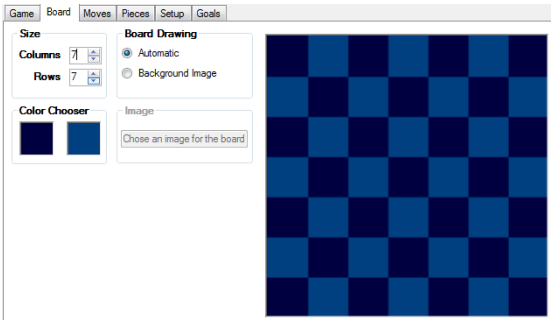

Figure 1 - Board Creation Page

The creation of ASG through a generic interface requires a thorough analysis of game mechanics, especially for finding ways to generalize options that can encompass several rules of different abstract games. Another important aspect to consider is the way game data is organized in the ZRF file. In the initial phase of development several interface sketches were created in order to try out different ideas for the project. It is important to focus on usability and to restrict, whenever possible, the actions of the user so that the options chosen won't contradict each other. Despite the capabilities of the ZoG language for describing almost any type of abstract game with a high enough degree of freedom, several restrictions to game creation must be applied, at least in the first stages of development.

4.1. Prototype

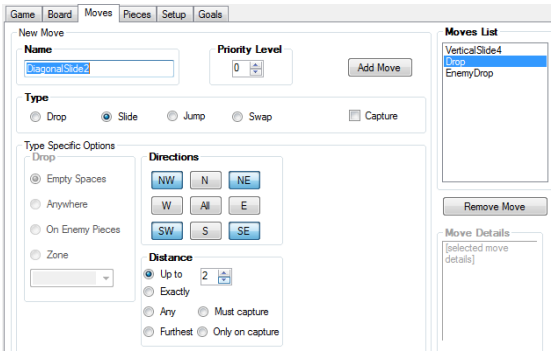

Figure 2 - Moves Page

The generic interface is designed with a similar ideology as a typical Windows Wizard form, where option filled pages advance systematically. The six pages are "Game", "Board", "Moves", "Pieces", "Setup" and "Goals". This prototype was also developed with default values for most options, which is helpful for the debugging process as well as for providing the user a chance to skip some lesser priority choices.

This first section of the interface, "Game", will act as a welcoming screen, providing the user with choices like the name of the game to be created, the names of each player and some optional textual information about the game. The following page is the "Board", where the user is given the choice of automatically creating any rectangular board by choosing the number of rows and columns, as well as defining an image as the board background. The "Moves" section allows the creation of several different types of piece movements, which are the most commonly used in ASG, as described in Table 1 of this document. The available

options are: drops (placing an off board piece in the game), slides (moving a piece in a direction without any obstacles), jumps (move in a direction jumping over obstacles) and swaps (two pieces switching places). For every type of move there is the option to make it a "capture" move (except for swapping moves), which in turn modifies the created move in order to eliminate an opponent piece as the move is performed. When a move type is selected, specific options related to that move type become available, like direction or distance for jumps and slides. To prevent logical mistakes, some options will be grayed out if incompatible ones are selected.

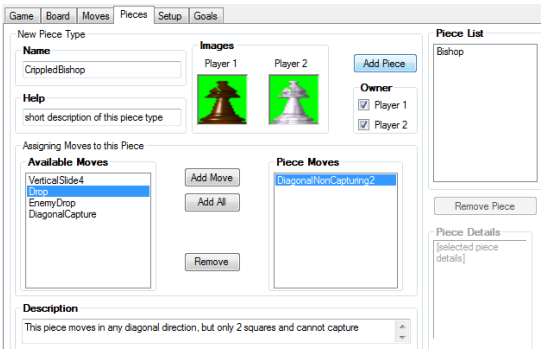

Figure 3 - Piece Creation Page

The "Piece" section is used for adding pieces to the game, and for each piece it is necessary to choose from the moves created in the "Moves" section, as well as deciding other useful attributes about the piece itself, such as name and images associated with it. The "Setup" section's purpose is to aid the user in deciding the initial game setup (like off board and on board pieces, and their respective spaces). Finally, the "Goals" page tries to apply the concepts derived from the classification system mentioned in Table 1. There are ten different generic goal types to choose from, hopefully encompassing most of the existing ASG' endgame situations.

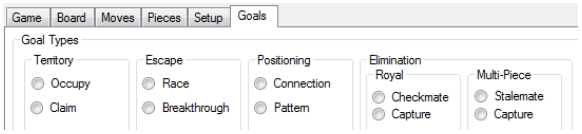

Figure 4 - Goals Page

5. Conclusions and Future Work

The prototype described in the previous section is still in its early development stages, as the final project requires that an increasing number of features be implemented. For instance, in what regards board creation, it is necessary to allow board types to be constituted of any type of shape so that it is possible to create games with hexagonal or triangular boards. This feature will only be possible by designing a powerful board drawing tool. Extending board creation even further by adding an option to link cells indiscriminately would allow for the simulation of spherical, conical and cubic boards, fully implementing the 3D features of some ASG. Another interesting aspect of this is the possibility of having warping spaces in a board.

A crucial addition to the program is the possibility of manually editing the ZRF generated code, enabling more advanced users to explore unimplemented features of the ZoG language. Another essential step is a broader definition of endgame goals, explicitly declaring draw

conditions and possibly combining several goal types in order to create hybrid games with multiple ways to reach victory.

Zones play in important part in defining rules for some games, it is thus necessary to define them in order to provide promotion and special movements triggered by piece positions on the board. Multiple piece movements and stacking are also common place amongst several games, as well as composite moves with more than one direction. All of these features are examples of how this interface can benefit from the ZRF capabilities while keeping to its objective of generalizing game creation.

Despite the number of features still in development, this prototype already shows that the project has an enormous potential to simplify ASG creation. As more and more options are included in the generic interface program, the closest the project will be of achieving its goal of becoming a truly universal abstract game generating tool, notwithstanding the limitations of the Zillions game engine.

However, the creation of a game by an unreasonable collection of options does not always generate a playable ASG, as some rules oppose one another, or simply because the game may lack in determinism or a logic succession of playable moves. The Zillions engine will still try to solve the game, but that doesn't necessarily mean that the game is playable, especially if it falls out of the most basic definition of an ASG: a solvable puzzle.

References

[1] Browne, C., Maire, F. *Evolutionary Game Design*. IEEE Transactions on Computational Intelligence and AI in Games, Vol.2, Issue 1, pp. 1-16, 2010.

[2] Genesereth, M., Love, N. *General Game Playing: Overview of the AAAI Competition*. AI Magazine, Vol. 26, Number 2, pp. 62-72, 2005.

[3] *Definitions of Abstracts*. International Abstract Games Organization. 2009. Available from: abstractgamers.org/wiki/definitions-of-abstracts (Retrieved 02/2010)

[4] Neto, J. *The LUDÆ Project*. 2003. Available from: homepages.di.fc.ul.pt/~jpn/ludae/ (Retrieved 02/2010)

[5] Neto, J. *World of Abstract Games*. 2010. Available from: homepages.di.fc.ul.pt/~jpn/gv/ (Retrieved 02/2010)

[6] Thompson, J. *Abstract Games*. 2010. Available from: home.flash.net/~markthom/html/abstract_games.html (Retrieved 02/2010)

[7] *Zillions of Games*. Zillions Development Corporation, 2009. Available from: www.zillionsofgames.com/ (Retrieved 02/2010)