SUPINFO Academic Dept.

SUPINFO
International University

INSTITUTE OF INFORMATION TECHNOLOGY

Group Projects - Resit
1DEV – Takuzu

Project presentation

2014-2015

Version 1.0
Last update: 23/06/2015
Use: Students/Staff
Author: SAD

TABLE OF CONTENTS

1 PROJECT OVERVIEW

Howard interactive is specialized in video games adaptation of classic board games, for various platforms. They currently have several games in their catalog such as go, mastermind and chess.

Howard interactive wants to take on a new challenge with "Takuzu", a rather complex game for one player. According to Wikipedia, this game is a "logic-based number placement puzzle".

You have been personally chosen among several subcontractors to do the development. As the game will run on linux-based TV appliances, you're free to use either Python (with Pygame) or C (with SDL) for the development.

2 FUNCTIONAL EXPRESSION

2.1 THE GAME

The goal of this single player game is to fill a squared grid with 0s and 1s.

The game starts with an unsolved grid like that:

	1		0
		0	
	0		
1	1		0

The player has to complete the grid following only three rules:

- Each row and column contains an equal number of 0s and 1s.
- There is no identical row or column.
- There is no more than two identical numbers adjacent to each other.

For the previous example, the solved grid is :

0	1	1	0
1	0	0	1
0	0	1	1
1	1	0	0

2.2 FEATURES TO IMPLEMENT

Your implementation of the game must provide the following features:

- Menu
- Difficulty levels.
- Creation of the grid.
- Graphic rendering
- Rewind.
- Hint.
- Mistakes.
- End of game.

All interactions must be done with a **GUI**.

2.2.1 Menu

The player should be able to choose betwen read the rules or start the game.

2.2.2 Difficulty level

The player should be able to choose the grid size between 4, 6, 8 or 10.

2.2.3 Creation of the grid

The CPU creates a (solved) grid of the choosen size. This grid must follow the three rules exposed in the section 2.1.

The CPU hides some cells to obtain an unsolved grid which is resolvable.

2.2.4 Graphic rendering

The grid should appear appear with a white background, and the 1s and 0s should be represented by red and blue cells. So, the unsolved grid of the previous example will look like this:

	🟥		🟦
		🟦	
	🟦		
🟥	🟥		🟦

To color a cell, the player has to do one clik for the red color, two clicks for the blue, and three to discolor.

So, at the end of the game the grid will look like this:

🟦	🟥	🟥	🟦
🟥	🟦	🟦	🟥
🟦	🟦	🟥	🟥
🟥	🟥	🟦	🟦

2.2.5 Rewind

Anytime, the player can cancel his last moves with a LIFO order.

2.2.6 Hint

Anytime, the player can ask the CPU for a hint. It indicates a move which is necessarily exact.

2.2.7 Mistakes

When the player makes a huge mistake, for example fill a row with more blue cells than red cells, the CPU reports it to the player.

2.2.8 End of the game

When the grid is completely filled, the CPU congratulates the player and asks him if he wants to play again.

3 DELIVERABLES

Students should include the following elements in their final delivery:

- A zip archive with the project source code. The source code must also comes with the build system used (Project file, autotools...), if any.
- Project documentation, based on the template.
    - Technical documentation explaining your choices and/or implementation choices/details on the following items (at least):
        - Graphic engine
        - Networking
        - Algorithmic choices for the hint, mistake and victory.
    - Game manual

The first document is an academic document. Addressed to the reader as a teacher, not as a client. The last one (game manual) is addressed to the reader as a user. These documents can be in French or in English, at your option.

4 GRADED ITEMS

The project will be graded as follows, on a 33/30 scale:

Documentation (3 points)

- User documentation (1 point)
- Technical documentation (2 point)

Game initialization (10 points)

- Menu (1 point)
- Choosing a difficulty level (1 point)
- Creation of a solved grid (5 points)
- Masking some cells to obtain an unsolved grid (3 points)

Game engine (17 points)

- The player can color a cell (2 points)
- The player can cancell the last moves (2 points)
- The player can obtain a hint (5 points)
- The CPU indicates mistakes (5 points)
- The CPU indicates the end of the game (2 points)
- The CPU asks to play again (1 point)

Bonus features (3 points)