Object-Oriented Programming 2

Rolf Haenni & Annett Laube

Project

1 Project Description

The goal of this project is to implement a simple one-player game in Java. The game is developed in groups of two (except in case of an odd number of students, in which there will be one group of one). The following two deliverables will be evaluated.

Java Source Code

- Deadline: **Sunday, June 11th** (midnight)
- Compilable and executable
- Application runs properly
- Coding conventions (class/variable/method names, private/protected/public, etc.)
- Clean code (no unused variables, no warnings, proper use of data structures, programming patterns etc.)
- JavaDoc
- Tests

Presentation

- Date: **Wednesday, June 14th** (participation mandatory for everyone)
- 10–15 minutes (details follow)
- Short demo
- Overview of implementation (e.g. UML class diagram)
- Interesting code snippets (e.g. observer pattern)
- Extra features

2 Evaluation and Grading

The weight of the project is 12.5% of your final grade. You'll receive up to 10 points if all minimal features are implemented properly, and up to 2.5 points for extra features (max. 15 points). If you do not show up to present your project, 5 points will be deducted.

IMPORTANT: Any form of plagiarism (code copied from other groups or the Internet) will be reported to the study direction and punished by grades F for all group members.

Minimal Requirements

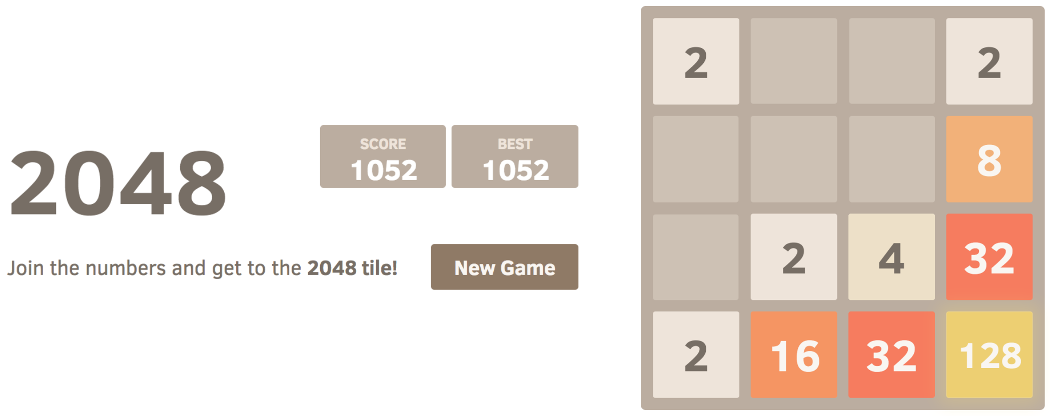
- Game rules properly implemented
- JavaFx user interface: visualisation of game board, scores, buttons, menu, etc.
- Simple AI players with different strategies (random/greedy/protective/...)
- Game statistics, high scores (XML)
- Separation of user interface and application logic (MVC, observer pattern)

Extra features

- Various board sizes
- Advanced AI player
- Automated evaluation of different AI strategies
- Animations
- Graphical statistics

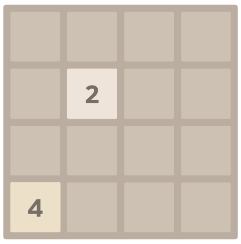
3 Game Description

"2048" is a one-player game often played on tablet computers. It can also be played in the web browser, for example on http://gabrielecirulli.github.io/2048/.

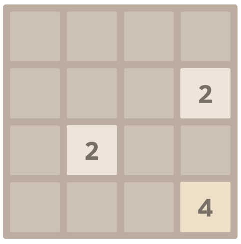


Game Setting and Basic Rules

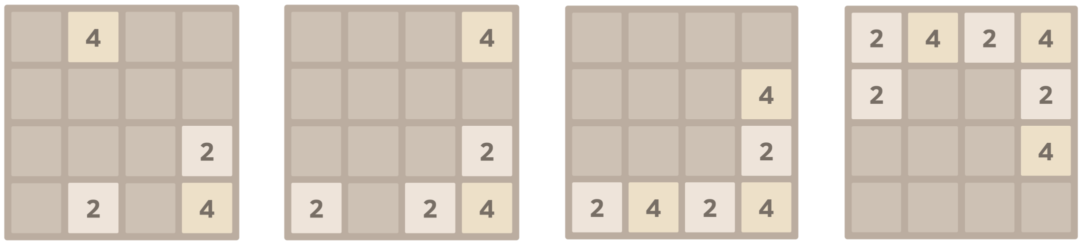
In the standard version, the game consists of a 4-times-4 square of tiles. Each tile contains a number $x \in \{2, 4, 8, 16, \ldots\}$. In the beginning of the game, the board contains exactly two tiles containing two random number $x_1, x_2 \in \{2, 4\}$. For example:



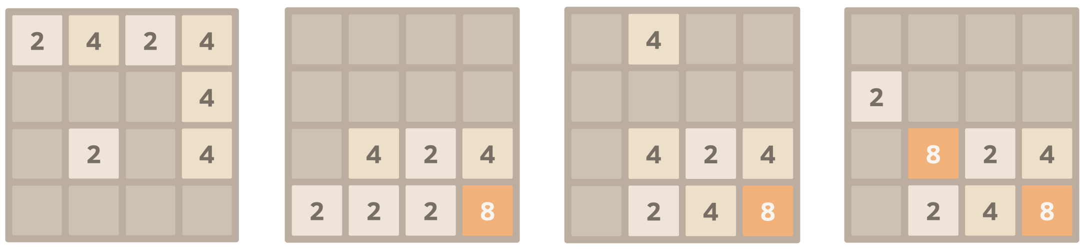
The player then has up to four different playing options: *left*, *right*, *up*, or *down*. Depending on the chosen option, the tiles are moved accordingly and a new tile containing a random number $x \in \{2, 4\}$ is inserted in a random position. In the original 2048 game, the probabilities are $P(x = 2) = 0.9$ and $P(x = 4) = 0.1$. For example, choosing *right* in the initial position shown above may lead to the following new game position:



The tile with the 4 on the bottom row and the tile with the 2 on the second row from top are moved rightwards (as far as possible) and a new tile containing a 2 is inserted. The following sequence of game positions is obtained from the precedent position by selecting *down*, *right*, *down*, and *up*.



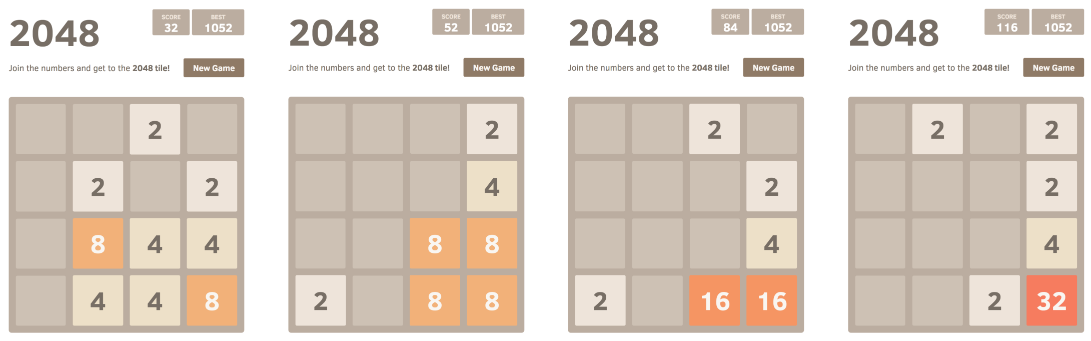
When adjacent tiles on the same row and with identical numbers x are moved to the left or to the right, they are merged into a new tile containing the number $2x$. The same happens for identical adjacent tiles on the same column, when they are moved up or down. The following sequence of game positions is obtained from the precedent position by selecting *right*, *down*, *right*, and *down*:



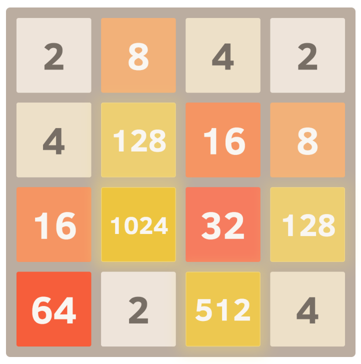
Note that if there are three adjacent identical tiles on one row or column, then only two of them are merged. In the above example, this happened in the third step by moving to the right, where the two rightmost tiles containing a 2 on the bottom row have been merged into a tile containing a 4. Moving to the left would have merged the two leftmost tiles. In a row or column with four identical tiles, they are merged into two tiles.

Counting and End of Game

Whenever two tiles are merged, the new value of the merged tile (or the sum of the values of the two merged tiles) is added to the score. The following sequence of game positions shows the development of the score from 32 to 52 (by moving to the right), from 52 to 84 (by moving down), and from 84 to 116 (by moving to the right):

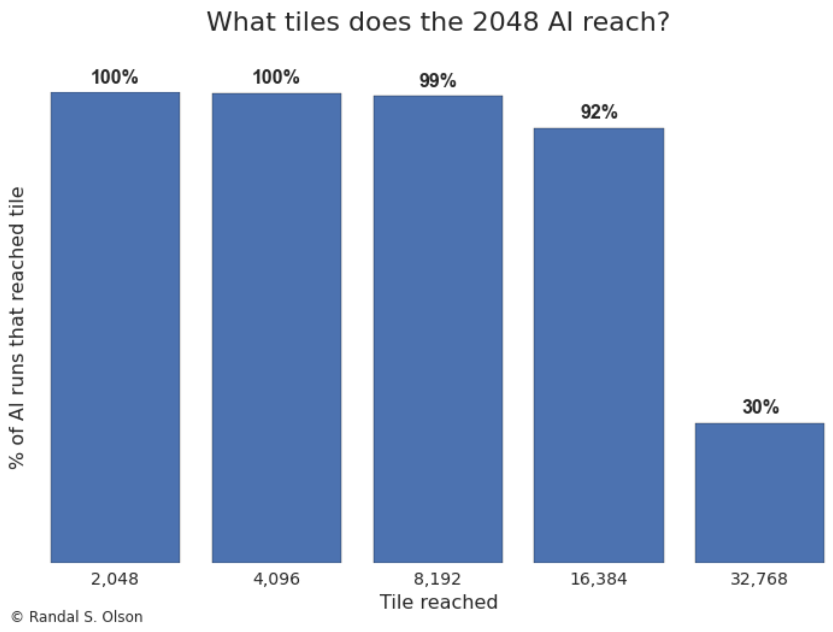

2048
SCORE 32
BEST 1052
Join the numbers and get to the 2048 tile!
New Game
2048
SCORE 52
BEST 1052
Join the numbers and get to the 2048 tile!
New Game
2048
SCORE 84
BEST 1052
Join the numbers and get to the 2048 tile!
New Game
2048
SCORE 116
BEST 1052
Join the numbers and get to the 2048 tile!
New Game

The game is over if no more moves are possible. This is the case in the following game position:

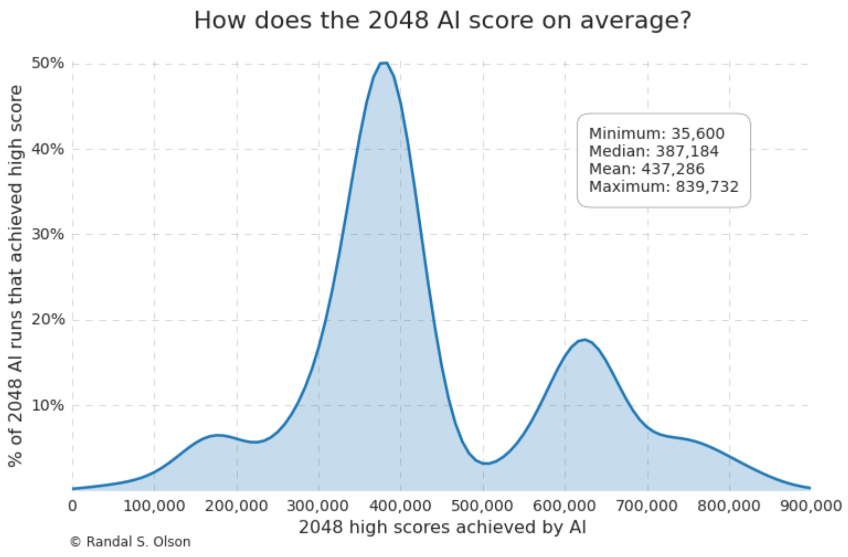


AI Players

The minimal goal of the game is to obtain a tile with the value 2048. Note that experienced player may go up to the 4096 or 8192 tile (theoretically, the highest possible tile is $2^{17} = 131,072$). There are implementation of AI players, which perform much better than most human players. The following statistics of one of the best AI implementations available shows, that reaching 32768 is not unusual (for more information on AI players and general playing strategies, see article on http://www.randalolson.com):



Another goal of the game is beating the hight score. The distribution of the high scores reached by the above AI players are shown in the following diagram:


How does the 2048 AI score on average?
Minimum: 35,600
Median: 387,184
Mean: 437,286
Maximum: 839,732
% of 2048 AI runs that achieved high score
2048 high scores achieved by AI
© Randal S. Olson