THE UNIVERSITY OF SYDNEY

INFO1113 Assignment 1 - 2048

Due: 13 April 2025, 11:59PM AEST

This assignment is worth 8% of your final grade.

Task Description

Implement the 2048 game using the Processing library for graphics and Gradle as a dependency manager. Setup of Gradle and a demonstration of the graphics capability of Processing will be done in the week 5 tutorial. You can access the documentation from here. As with any assignment, make sure that your work is your own, and do not share your code or solutions with other students.

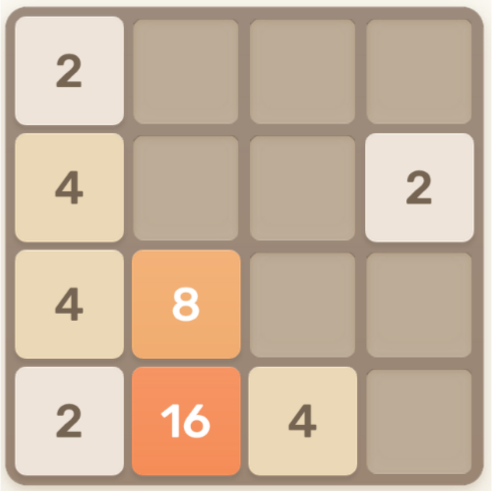

Figure 1. 2048 Example Game State

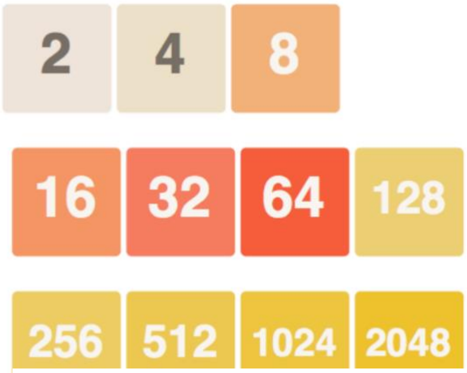

Figure 2. 2048 Tile Colour Reference. Any value higher than 2048 uses the same colour as 2048.

The game takes place on an $n \times n$ grid, initially with 2 randomly placed blocks (which are randomly selected from either a 2 or 4 with equal chance). The parameter n should be retrieved from command line arguments. If it is invalid or not provided, use the default which is n=4.

Each turn, the player may make one of four moves with the arrow keys: UP, DOWN, LEFT or RIGHT. This shifts all blocks in that given direction, and also merges adjacent blocks with the same number into a higher value (eg, 4 and 4 makes 8). You should ensure that block movement is smoothly transitioning from one cell to the next (as in the example game linked above), OR that only one block moves one space per frame.

After the player's move is finished, a new random block will spawn in one of the empty spaces. This can be either a 2 or a 4. In addition, the player may also spawn any amount of new blocks into empty cells themselves by clicking on an empty space, and a 2 or 4 will then spawn there (randomly chosen with 50% chance). If the player is unable to make any moves that change the position of blocks, the game is over.

A timer in the top right corner of the screen should keep track of the number of seconds since the game began. When the game ends, this timer stops, and the text “GAME OVER” is displayed in the centre of the screen. The player can press ‘r’ to restart the game.

Marking Criteria (8%)

Your final submission is due on Sunday 13 April 2025 at 11:59PM. To submit, you must upload your build.gradle file and src folder to Ed. Do NOT submit the build folder. Ensure src is in the root directory with the other files, and not part of a zip, then press MARK.

Shown during tutorial in week 5 when Gradle and Processing are covered. (-1 deduction if your submission breaks these features)	• Window launches and shows initial layout correctly with brown cells. • Hovering over tiles with cursor changes the brown tile to a lighter colour to highlight it • Left clicking on a cell places a ‘2’ or a ‘4’ randomly in that position
1 mark	• Board size adjusts based on command line argument input (default 4x4 if not provided) • Timer counts up every second in the top-right corner of the screen. It stops when the game ends.
0.5 marks	• Tile colours are correctly shown to differentiate different values
1.5 marks	• Arrow keys result in tile movement in the corresponding direction (LEFT, RIGHT, UP, DOWN) if there is empty space for the tiles to move – 0.5 mark deduction for each with bugs or not working
1.5 marks	• Adjacent tiles are merged correctly after the player makes a move in either LEFT, RIGHT, UP or DOWN directions – 0.5 deduction for each not working
1 mark	• Random block spawns in an empty cell after the player’s move – either a 2 or a 4
1 mark	• If the player is unable to make any moves that change the position of blocks, the game is over. Display “GAME OVER”. Player can press ‘r’ to restart the game.
1.5 mark	• Movement is progressive such that blocks smoothly transition from one cell to the next (such as in https://play2048.co/) OR only one block can move one space on each frame.

Academic Declaration

By submitting this assignment you declare the following:

I declare that I have read and understood the University of Sydney Student Plagiarism: Coursework Policy and Procedure, and except where specifically acknowledged, the work contained in this assignment/project is my own work, and has not been copied from other sources or been previously submitted for award or assessment.

I understand that failure to comply with the Student Plagiarism: Coursework Policy and Procedure can lead to severe penalties as outlined under Chapter 8 of the University of Sydney By-Law 1999 (as amended). These penalties may be imposed in cases where any significant portion of my submitted work has been copied without proper acknowledgment from other sources, including published works, the Internet, existing programs, the work of other students, or work previously submitted for other awards or assessments.

I realise that I may be asked to identify those portions of the work contributed by me and required to demonstrate my knowledge of the relevant material by answering oral questions or by undertaking supplementary work, either written or in the laboratory, in order to arrive at the final assessment mark.

I acknowledge that the School of Computer Science, in assessing this assignment, may reproduce it entirely, may provide a copy to another member of faculty, and/or communicate a copy of this assignment to a plagiarism checking service or in-house computer program, and that a copy of the assignment may be maintained by the service or the School of Computer Science for the purpose of future plagiarism checking.