

COMP 204

Programming Studio



Instructor: Prof. Dr. Muhittin Gökmen

Teaching Assistants: Mustafa Ersen and Fatih Said Duran

Student Assistants: Onat Kaan Atılgan, Yusuf Kartal and Beyzanur Yıldız

Project 2

Tetris 2048

Tetris 2048

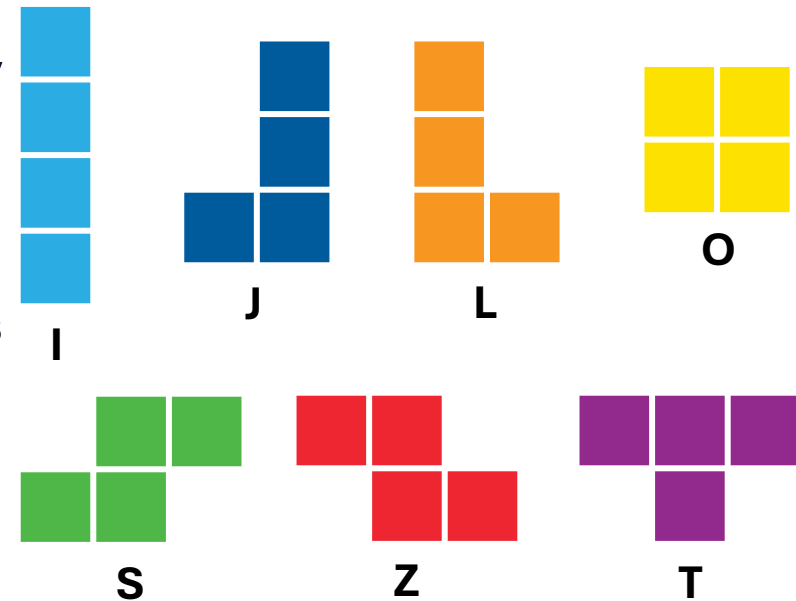
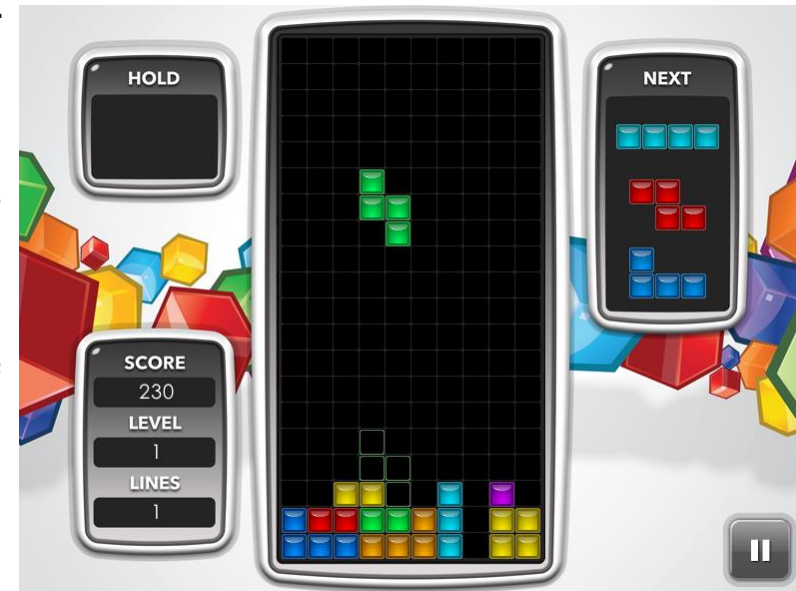


- In this project, you will develop a game in **Python 3** by using **the standard drawing module (stdraw.py)** that is provided by the textbook:
 - [Introduction to Programming in Python: An Interdisciplinary Approach](#) Approach by Robert Sedgewick, Kevin Wayne and Robert Dondero.
- The game you will develop is a combination of 2 well known games: Tetris and 2048.
 - First you will develop the Tetris game with its basic properties.
 - Then you will combine it with some features from 2048.
- An example image for the game is shown on the right.
 - The game includes standard Tetris shapes (tetrominoes) each of which consisting of 4 square tiles.
 - The tiles have numbers on them as in the game 2048.
 - The score and the next shape are shown on the right panel.



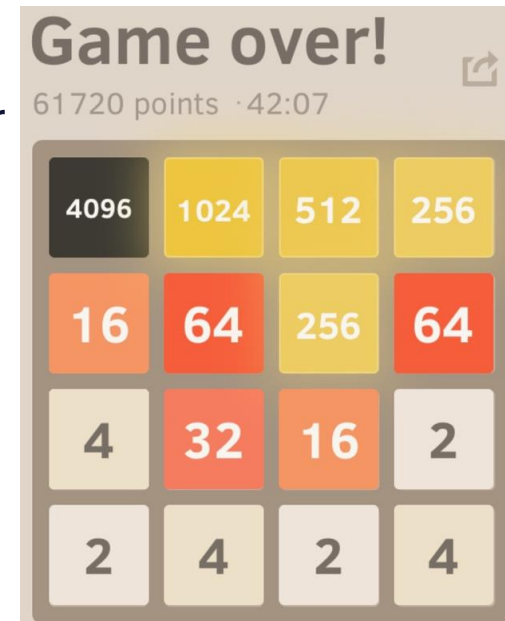
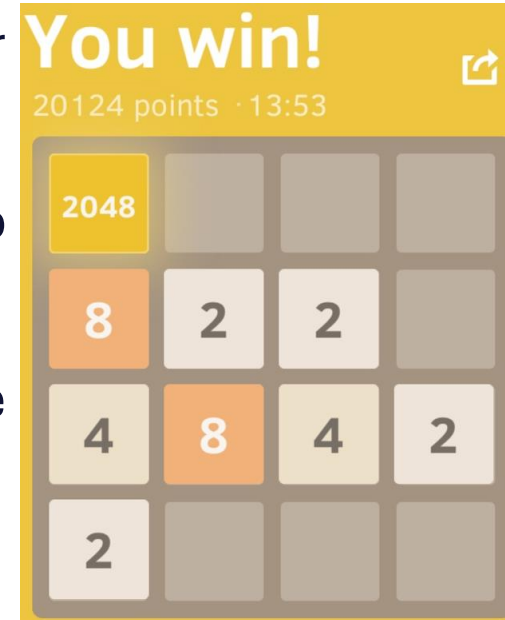
About Tetris

- The first version of Tetris was born from the imagination of computer programmer Alexey Pajitnov in 1984.
- It is a falling block puzzle game that lets players arrange puzzle pieces in real time as they are falling from the top of the playing field.
- The game environment for a typical Tetris game is shown as an image on the upper right.
- The main goal of Tetris is **clearing as many horizontal lines of blocks as possible**.
 - A horizontal line is cleared when it is full of blocks without any empty space.
 - It is possible to clear multiple lines at once.
- The player can move, rotate and (soft/hard) drop 7 different shapes known as tetrominoes each of which consisting of 4 square blocks (as shown on the lower right) in the 2D playing field.



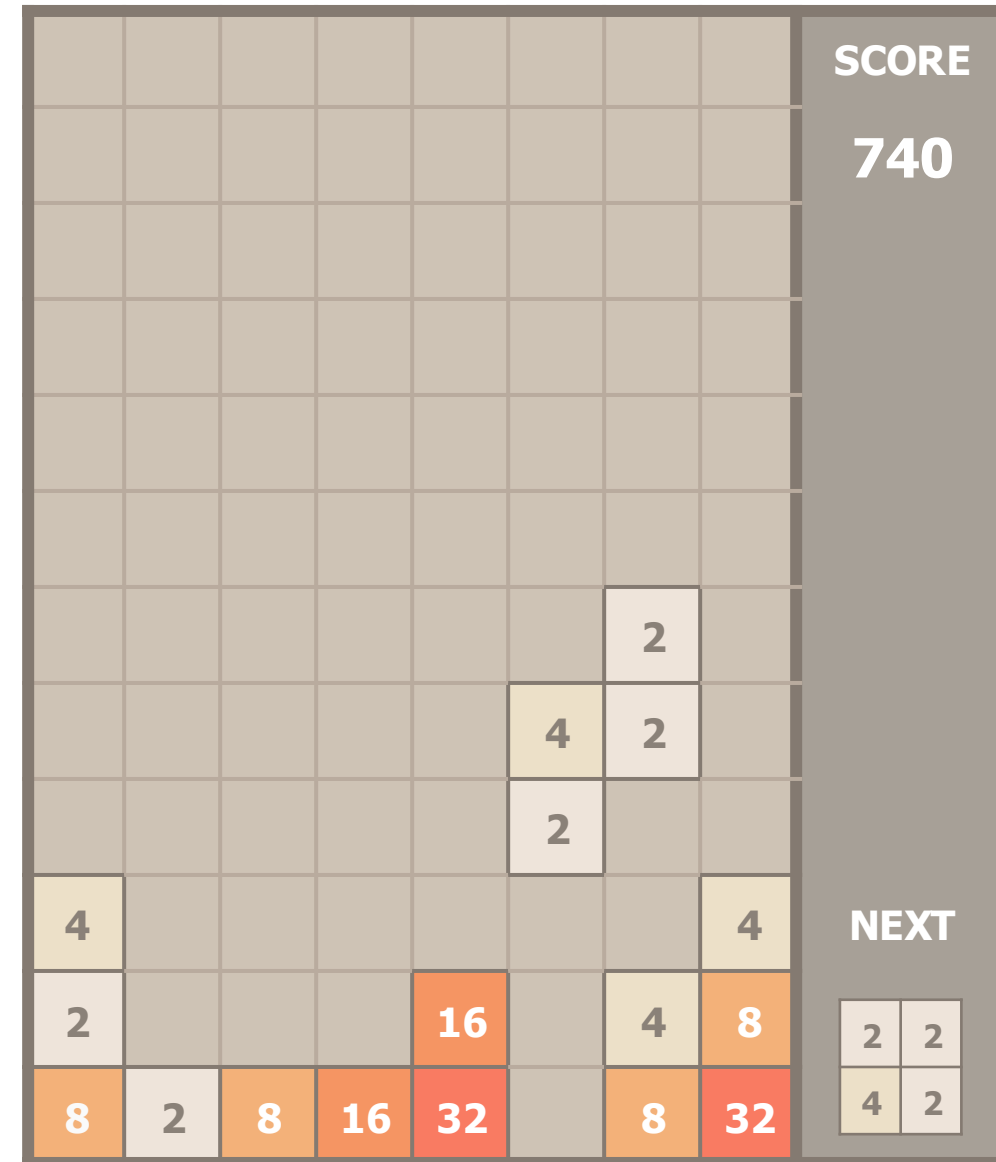
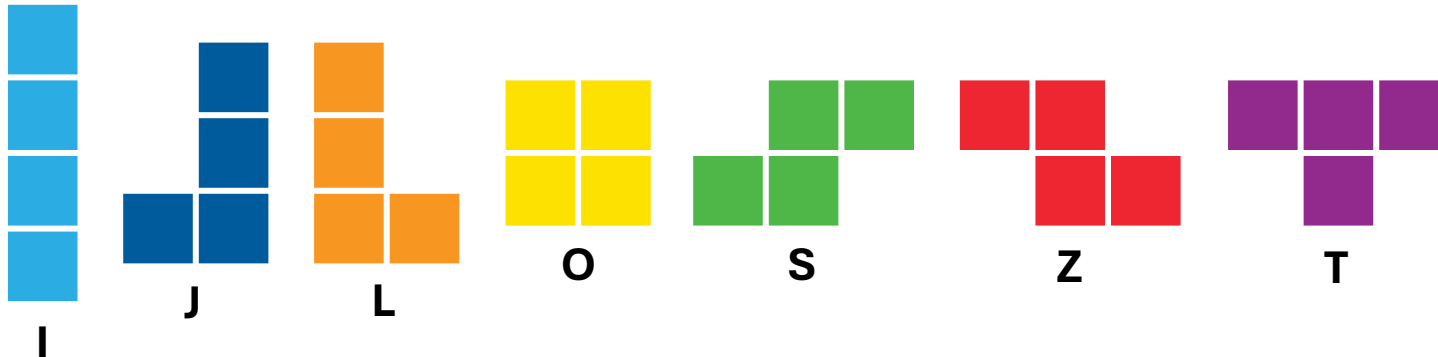
About 2048

- [2048](#) is a small puzzle game (created by Gabriele Cirulli) in which the player moves tiles on a 4x4 board.
- While moving the tiles, if two tiles with the same value touch, they merge into a single tile with their value doubled.
- For each move, a random tile is generated on the board with a random value as either 2 or 4.
- The criteria for winning the game is reaching the value 2048.
 - The game continues after reaching 2048 allowing to obtain much greater values.
- If the 4x4 board becomes full of tiles and no more move is possible, the player loses the game.



Tetris 2048

- Tetris 2048 combines some properties of Tetris and 2048.
- The game play is similar to Tetris as the player can move and rotate 7 different shapes each consisting of 4 square tiles (tetrominoes shown below) in the 2D playing field (grid).
- The differences are that the tiles forming tetrominoes have numbers (positive powers of 2) and the number on each tile determines its color (See the image given on the right).
- Each tetromino entering the grid consists of 4 tiles randomly numbered using either 2 or 4 (as in 2048).



Tetris 2048

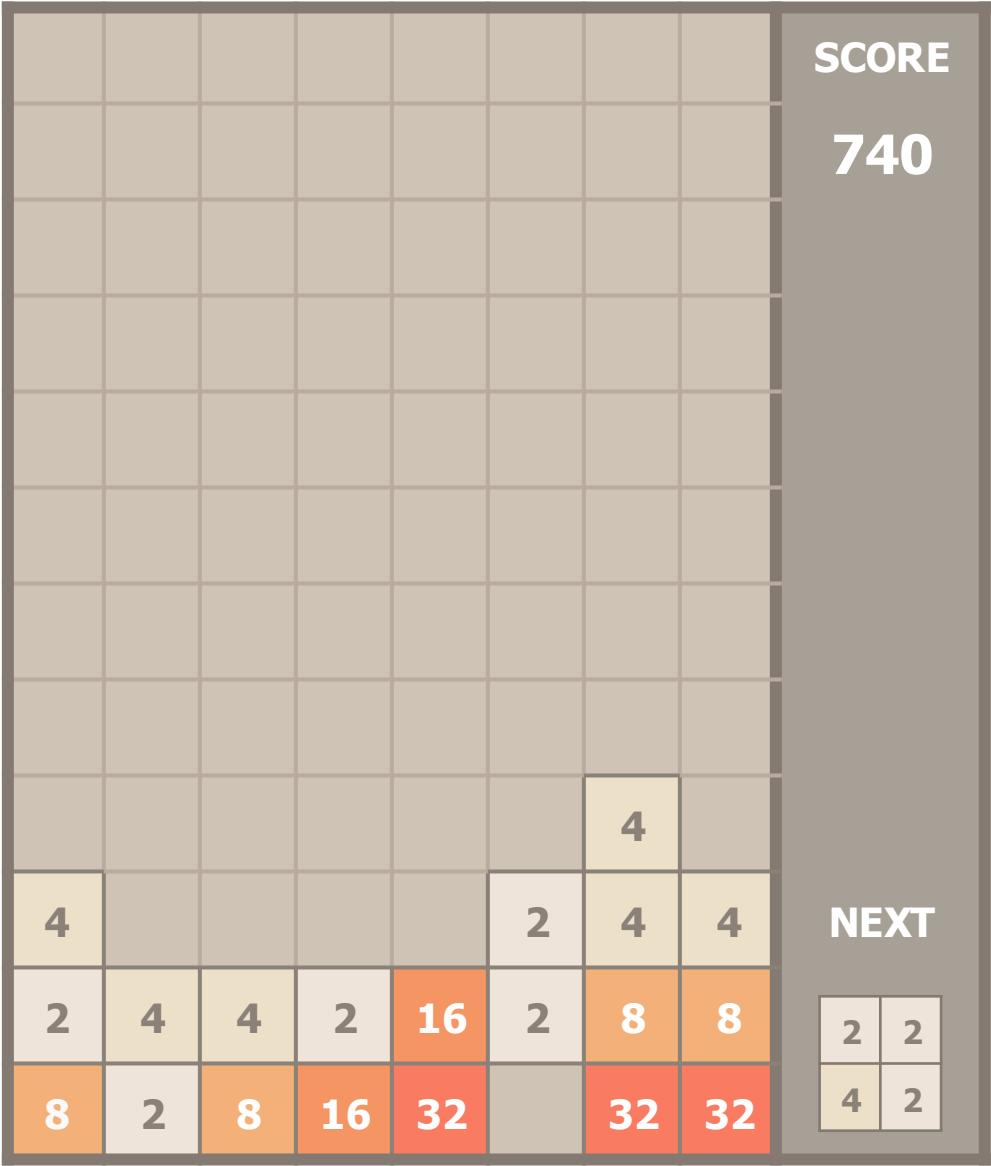
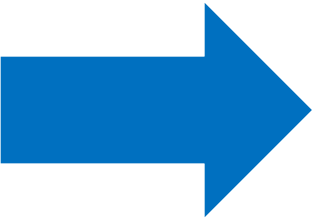
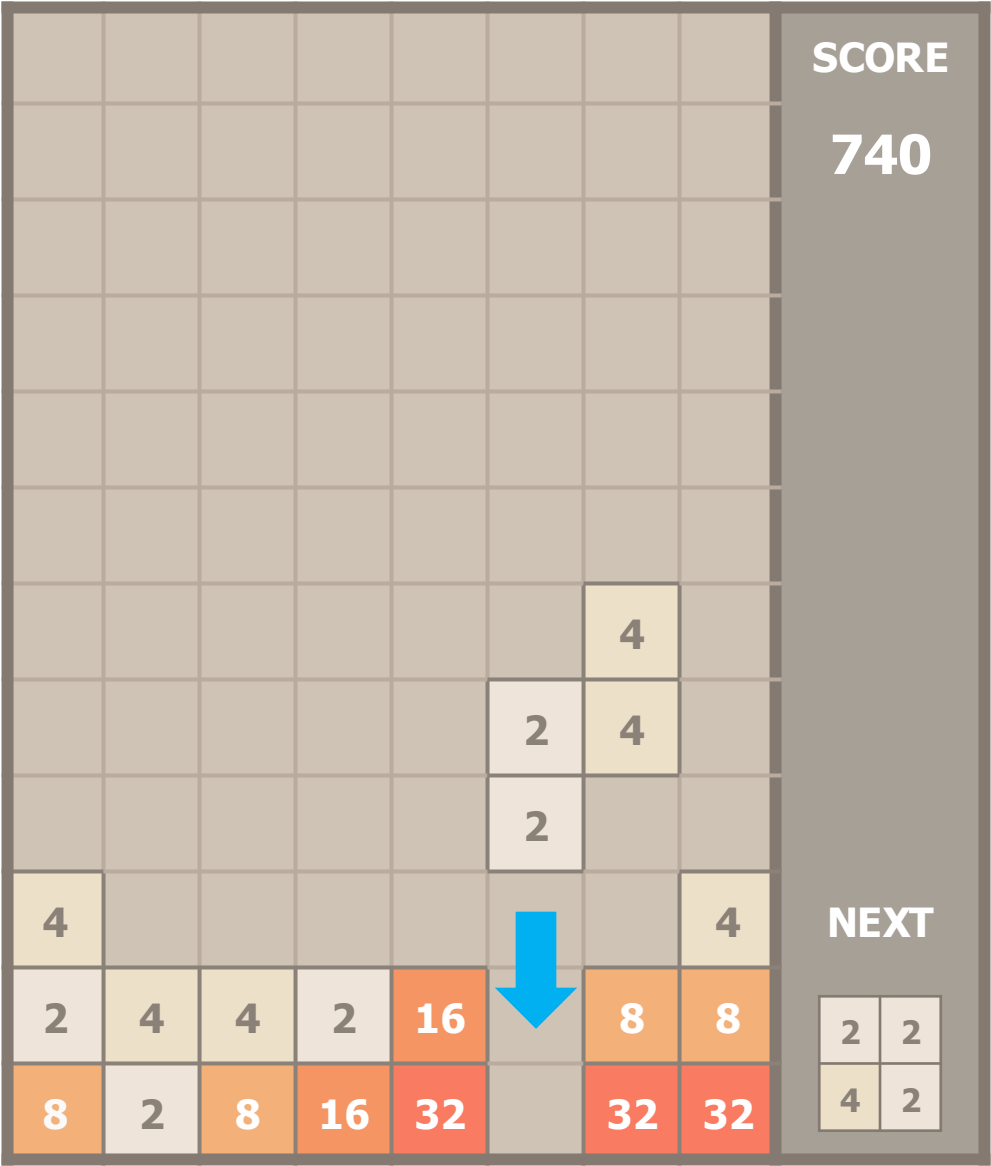
- The **score** is determined both from the **merged tiles** as in 2048 and the **cleared horizontal rows** as in Tetris.
 - The outcome (the resulting merged number) of each merge operation on a pair of tiles is added to the score.
 - The sum of all the numbers in each cleared horizontal row is also added to the score.
- **Merging tiles** has a **higher priority** than **clearing rows**.
 - After each landed tetromino onto the game grid, first, determine which tiles will be merged (a pair of column-wise touching tiles with the same number) and chain merge these tiles from bottom to top.
 - Then, check for horizontal rows to clear (each row that is full of tiles with no empty space) and remove these rows.
 - Do not forget to update the score based on these operations.
- The game can be won by obtaining a 2048 valued tile (as in 2048).
- The game is over when any tile is landed above the topmost row of the game grid (as in Tetris).
- **Note:** The grid on the right is just an example. You can use a grid with a different size and colors. You can add more features to the game such as changing the speed of the tetrominoes (increasing the difficulty as the game progresses), holding tetrominoes etc.



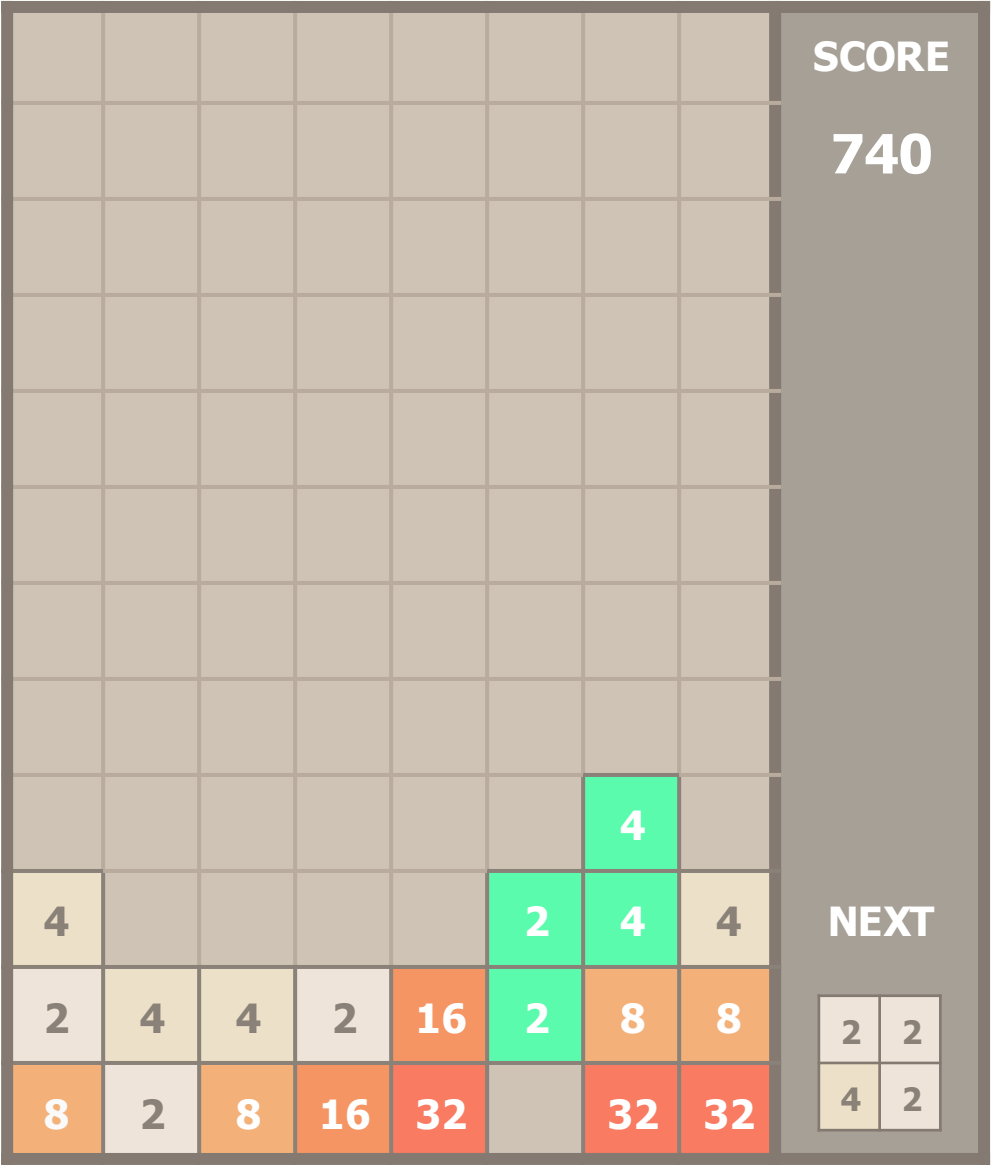
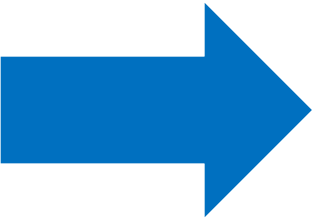
Example

An Illustration of the Game Operations on a Sample Gameplay

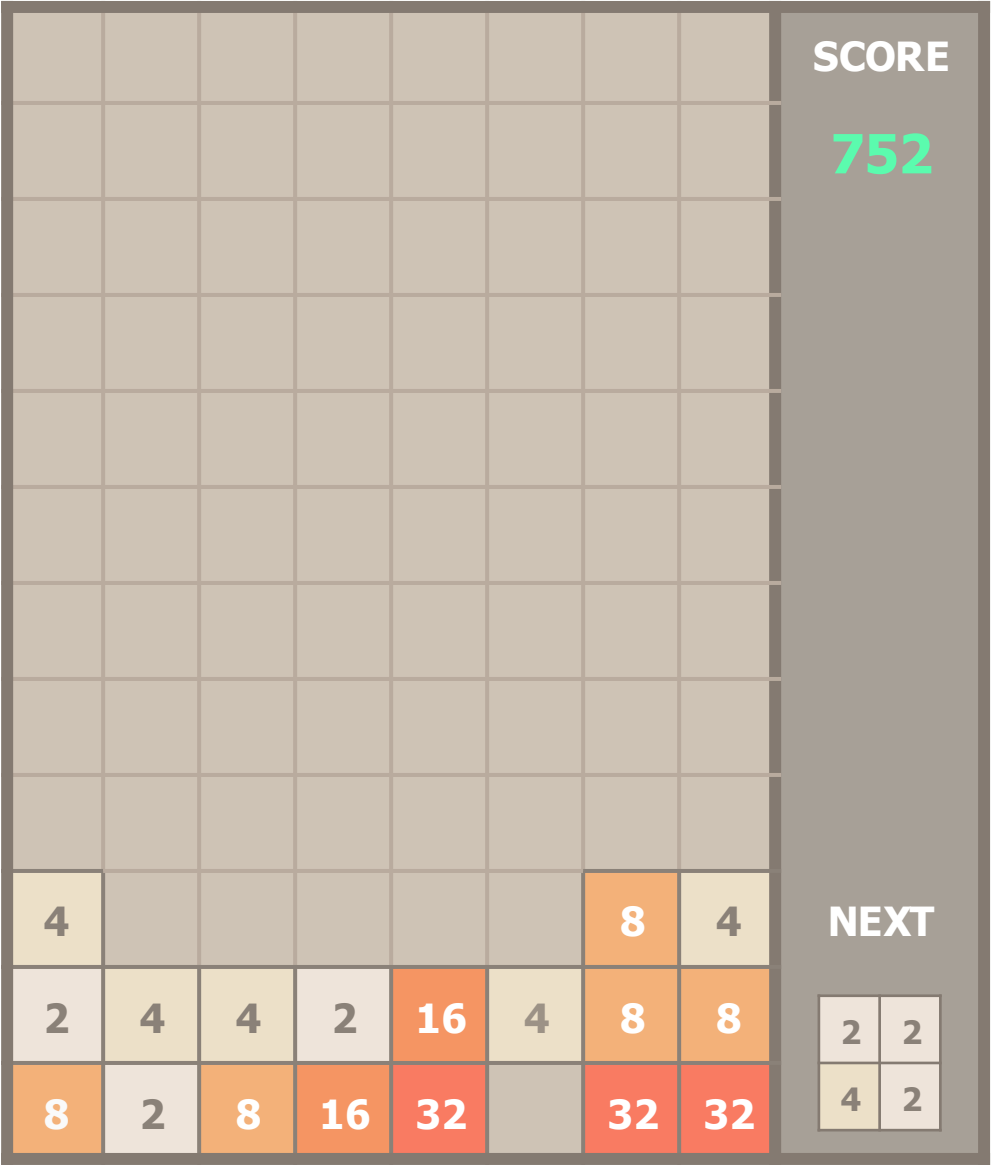
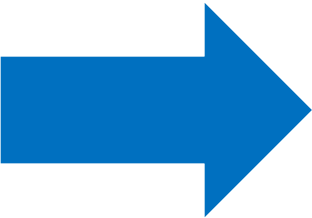
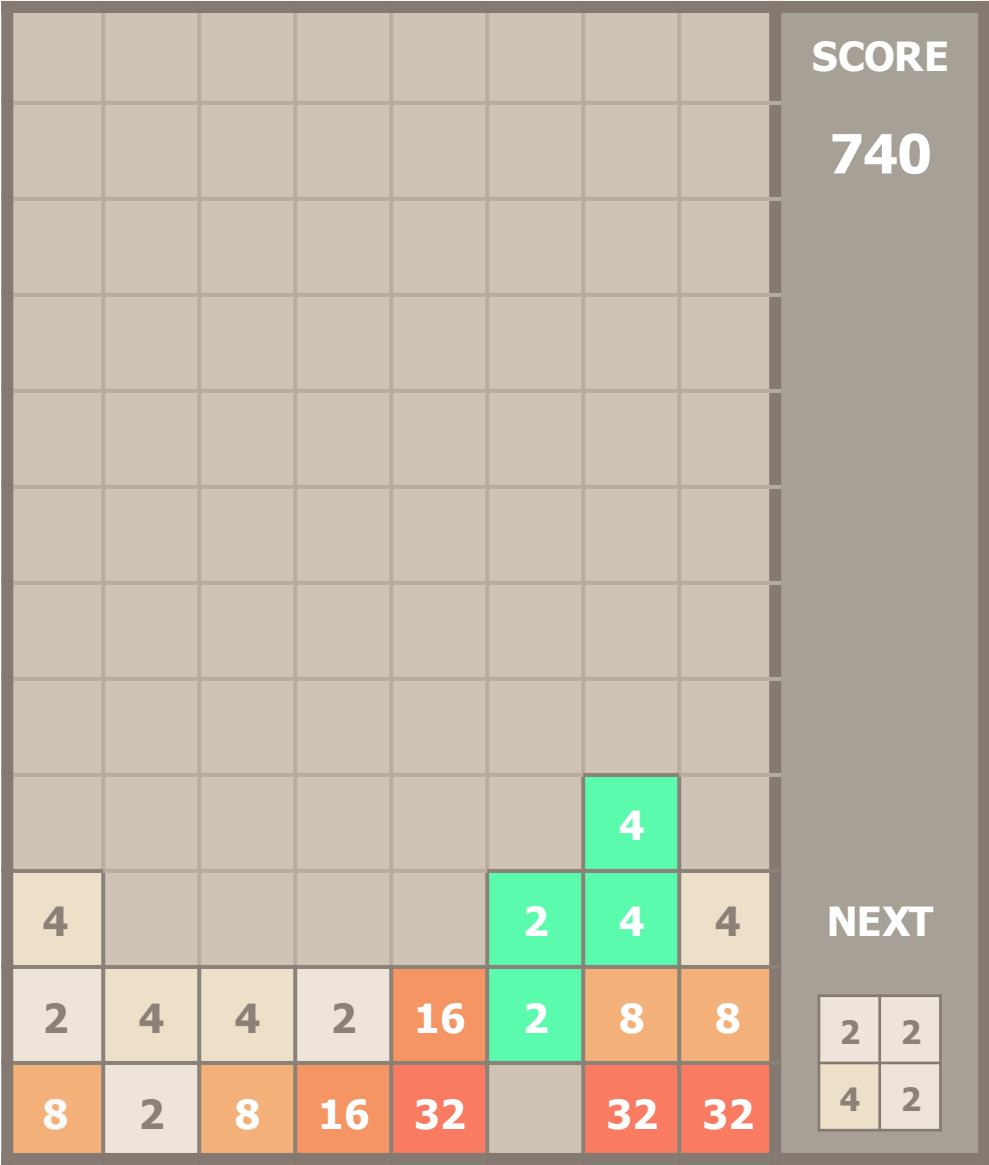
Merging Tiles and Clearing Rows



Merging Tiles and Clearing Rows



Merging Tiles and Clearing Rows



Merging Tiles and Clearing Rows

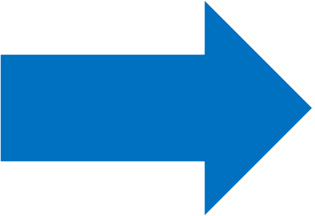
| | | | | | | | |
|---|---|---|----|----|---|----|----|
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| 4 | | | | | | 8 | 4 |
| 2 | 4 | 4 | 2 | 16 | 4 | 8 | 8 |
| 8 | 2 | 8 | 16 | 32 | | 32 | 32 |

SCORE

752

NEXT

| | |
|---|---|
| 2 | 2 |
| 4 | 2 |



| | | | | | | | |
|---|---|---|----|----|---|----|----|
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| 4 | | | | | | 8 | 4 |
| 2 | 4 | 4 | 2 | 16 | 4 | 8 | 8 |
| 8 | 2 | 8 | 16 | 32 | | 32 | 32 |

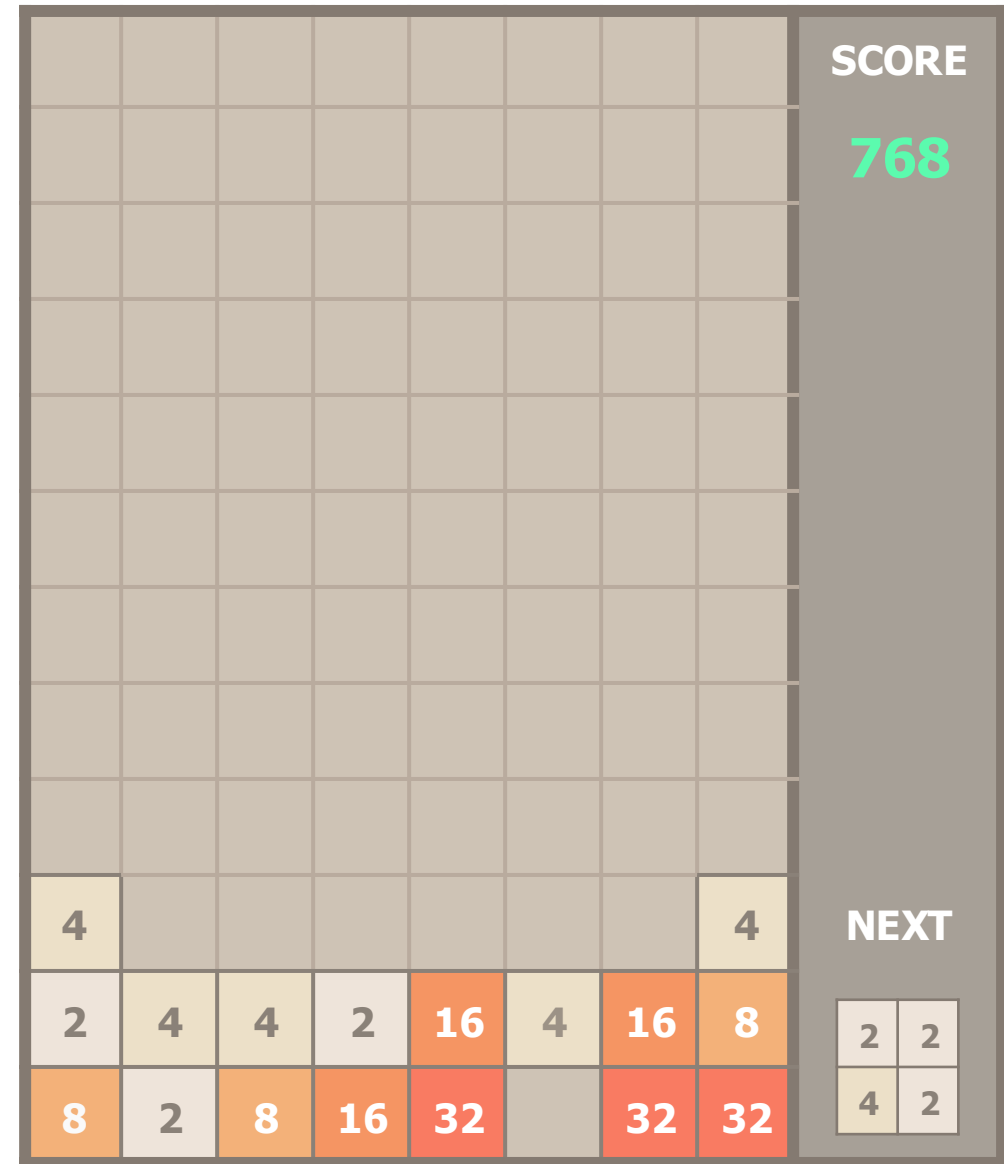
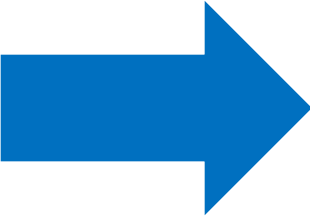
SCORE

752

NEXT

| | |
|---|---|
| 2 | 2 |
| 4 | 2 |

13



Merging Tiles and Clearing Rows

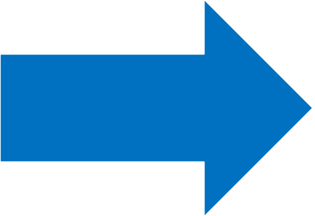
| | | | | | | | |
|---|---|---|----|----|---|----|----|
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| 4 | | | | | | | 4 |
| 2 | 4 | 4 | 2 | 16 | 4 | 16 | 8 |
| 8 | 2 | 8 | 16 | 32 | | 32 | 32 |

SCORE

768

NEXT

| | |
|---|---|
| 2 | 2 |
| 4 | 2 |



| | | | | | | | |
|---|---|---|----|----|---|----|----|
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| 4 | | | | | | | 4 |
| 2 | 4 | 4 | 2 | 16 | 4 | 16 | 8 |
| 8 | 2 | 8 | 16 | 32 | | 32 | 32 |

SCORE

768

NEXT

| | |
|---|---|
| 2 | 2 |
| 4 | 2 |

Merging Tiles and Clearing Rows

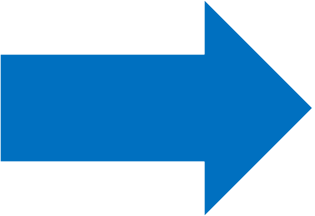
| | | | | | | | |
|---|---|---|----|----|---|----|----|
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| 4 | | | | | | | 4 |
| 2 | 4 | 4 | 2 | 16 | 4 | 16 | 8 |
| 8 | 2 | 8 | 16 | 32 | | 32 | 32 |

SCORE

768

NEXT

| | |
|---|---|
| 2 | 2 |
| 4 | 2 |



| | | | | | | | |
|---|---|---|----|----|--|----|----|
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| 4 | | | | | | | 4 |
| 8 | 2 | 8 | 16 | 32 | | 32 | 32 |

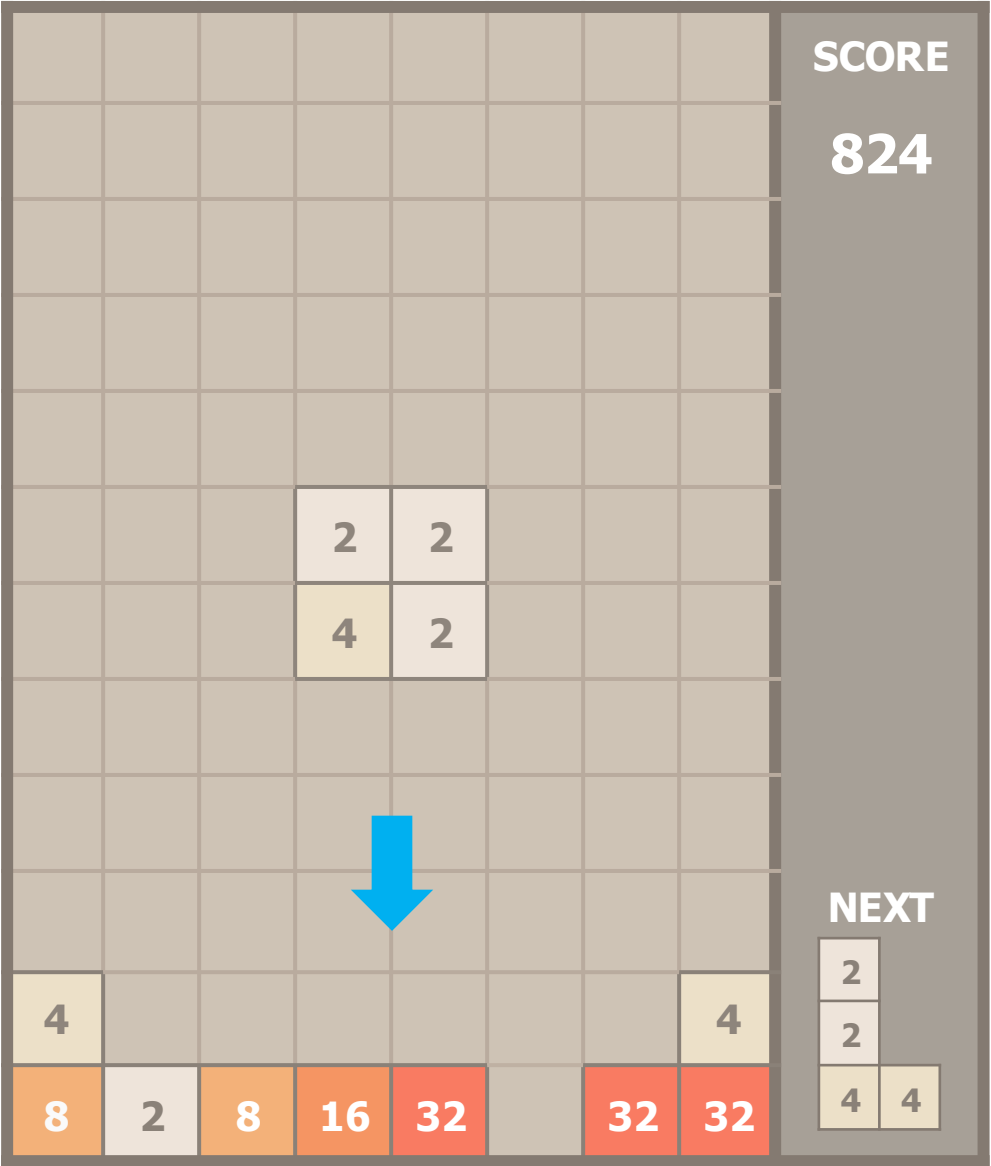
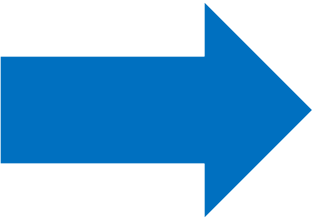
SCORE

824

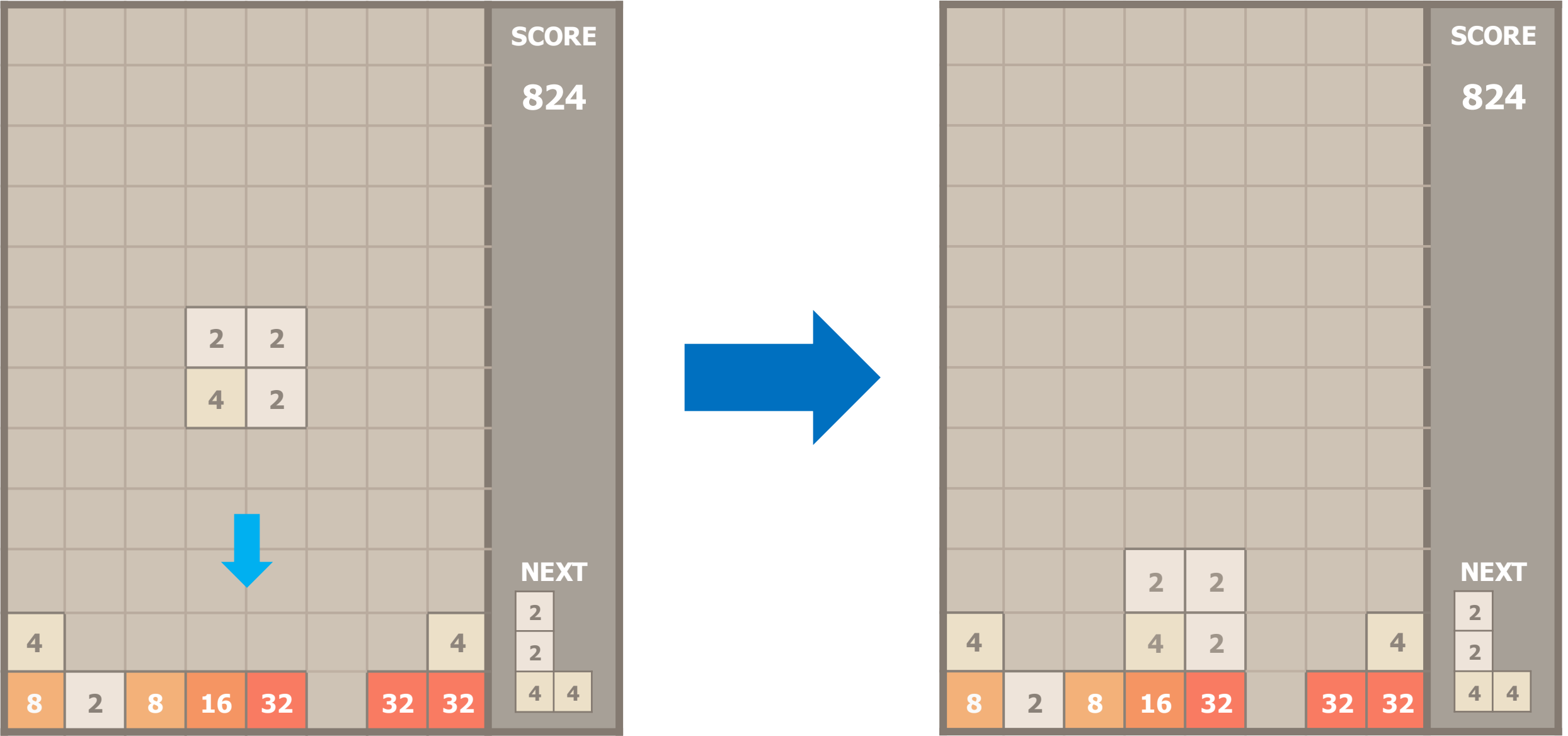
NEXT

| | |
|---|---|
| 2 | 2 |
| 4 | 2 |

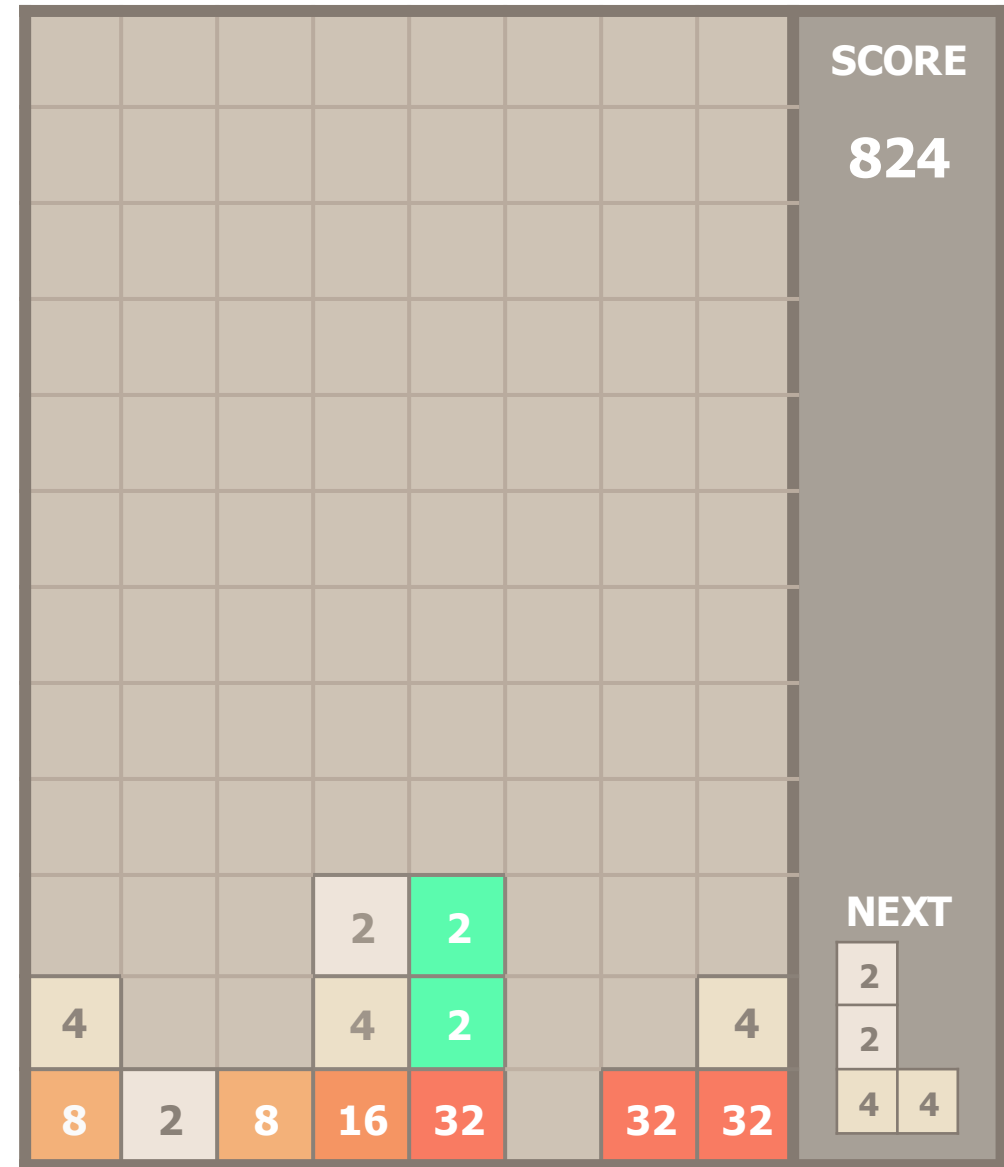
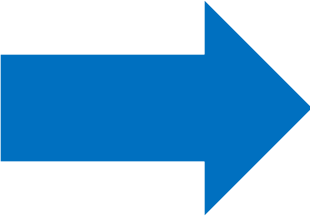
Merging Tiles and Clearing Rows



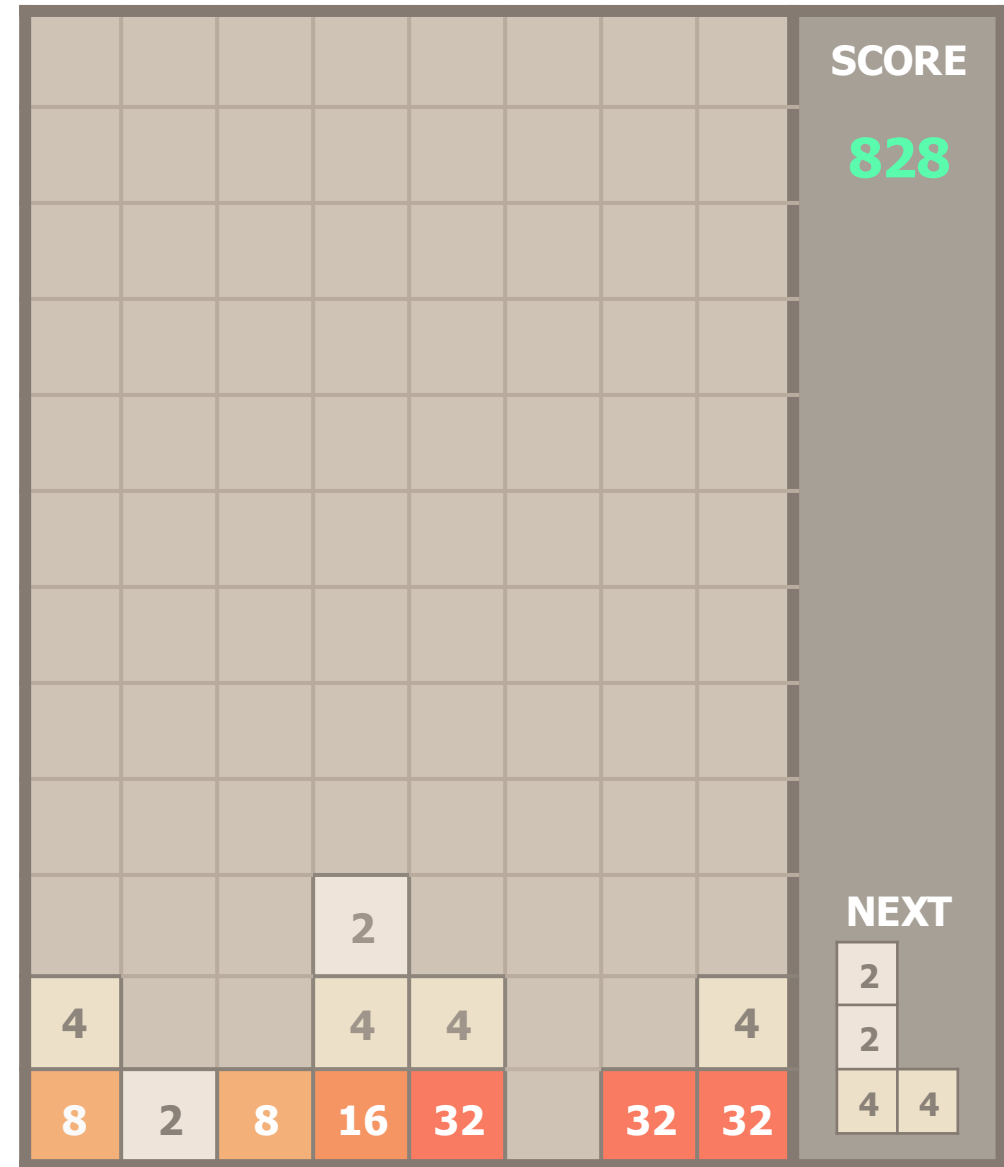
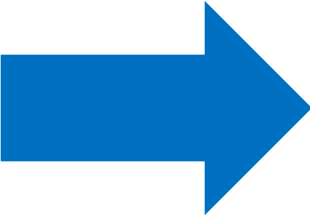
Merging Tiles and Clearing Rows



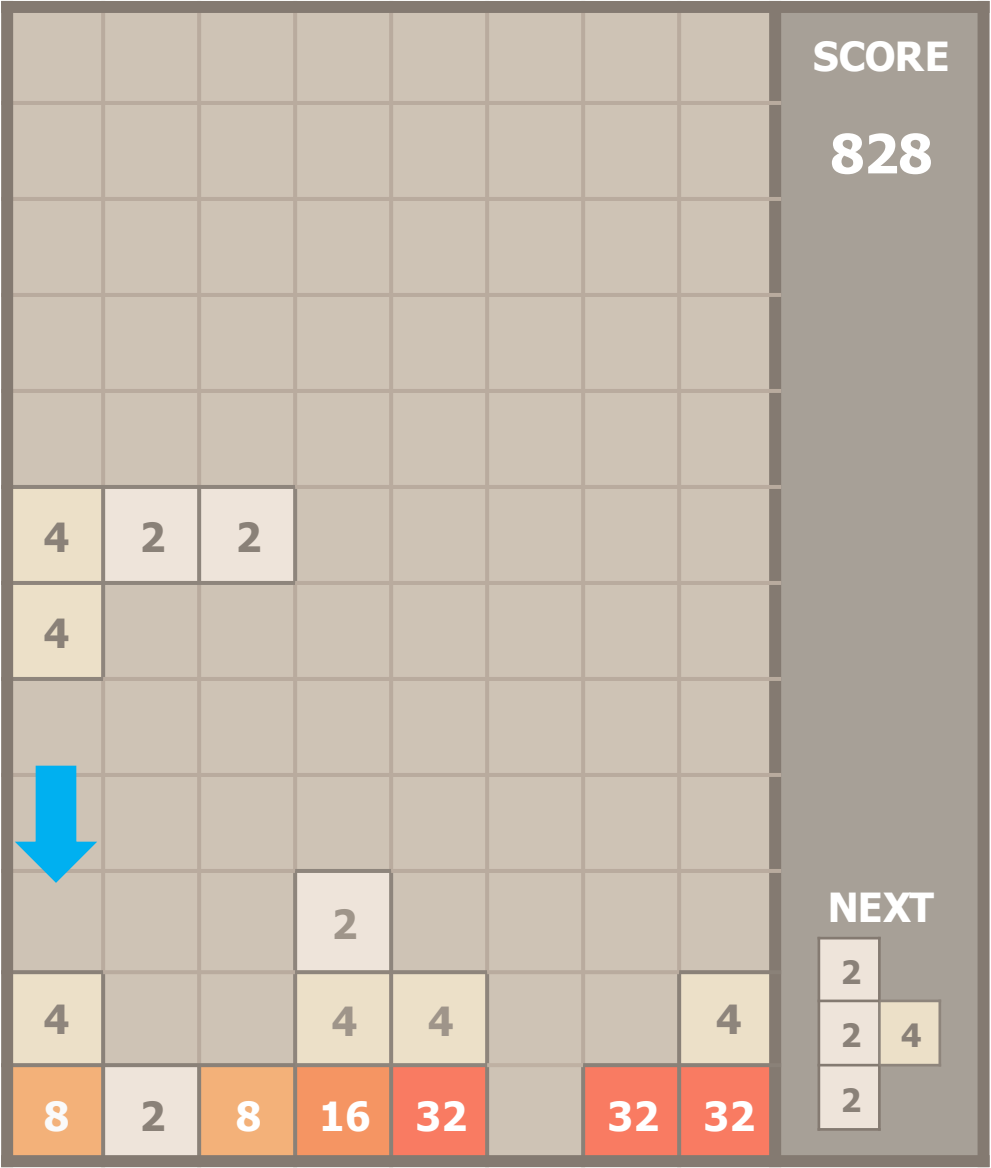
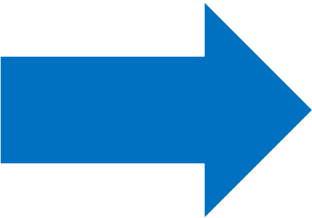
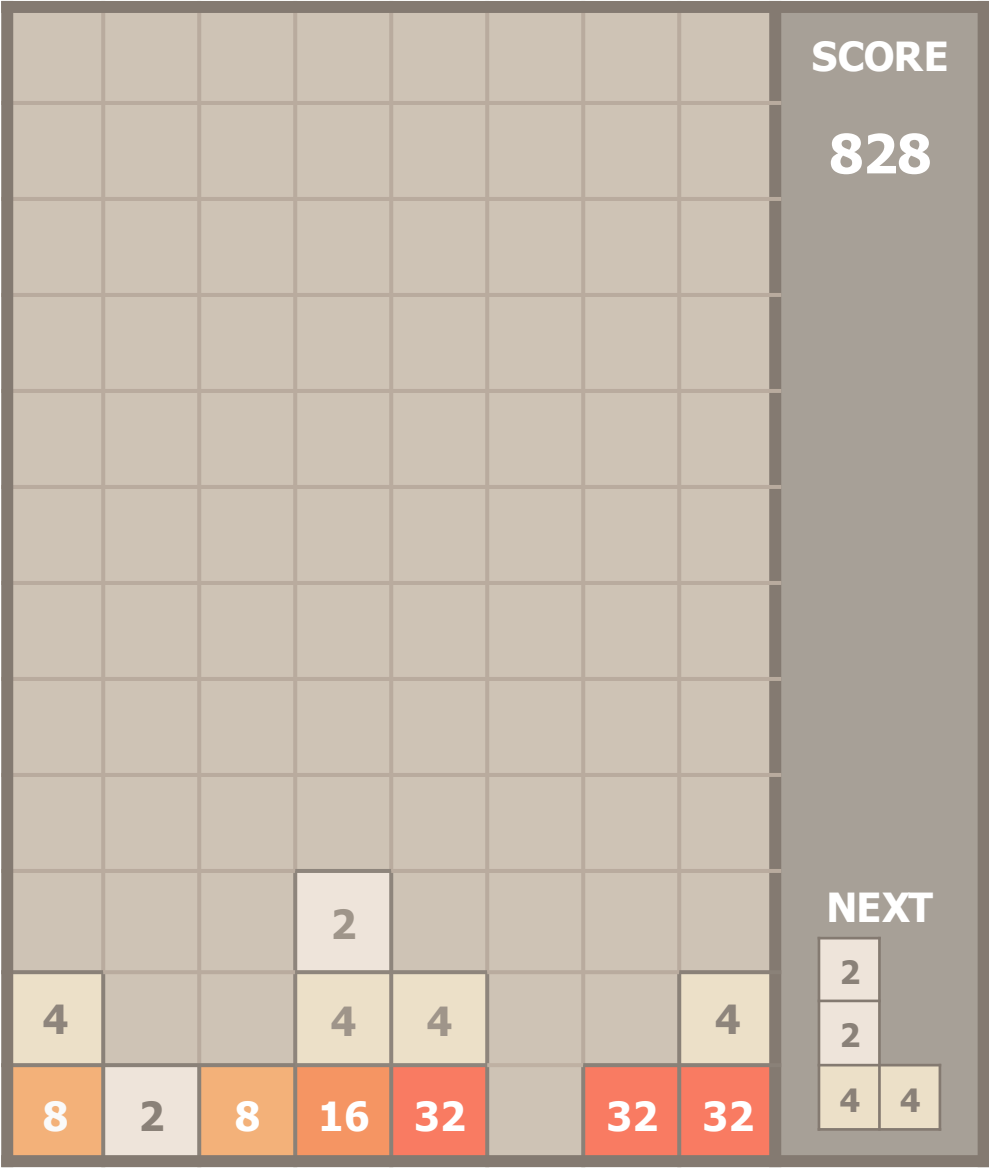
18



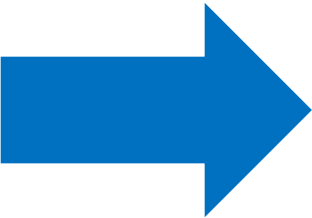
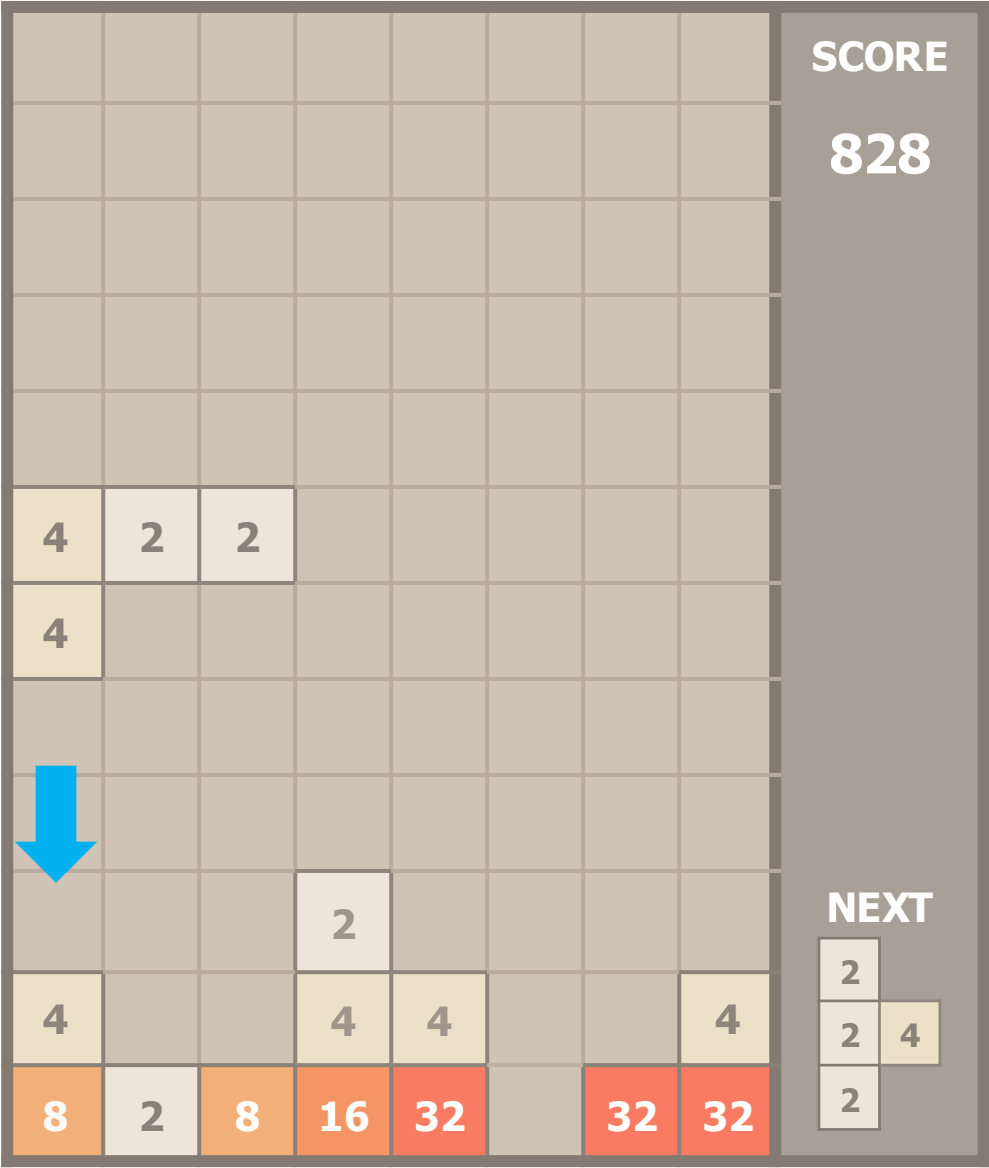
19



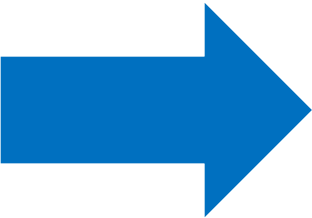
Chain Merging



Chain Merging



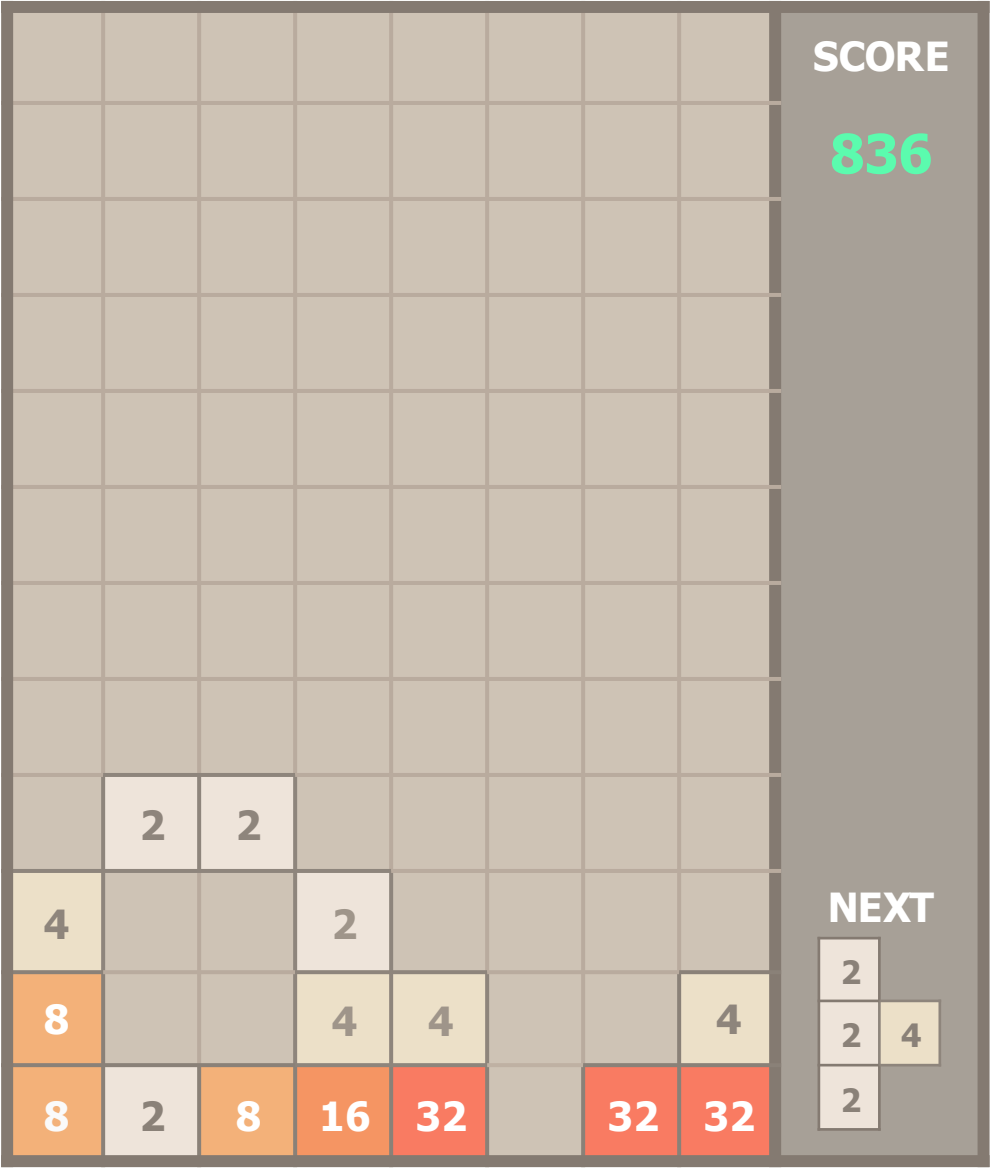
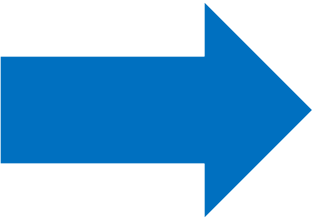
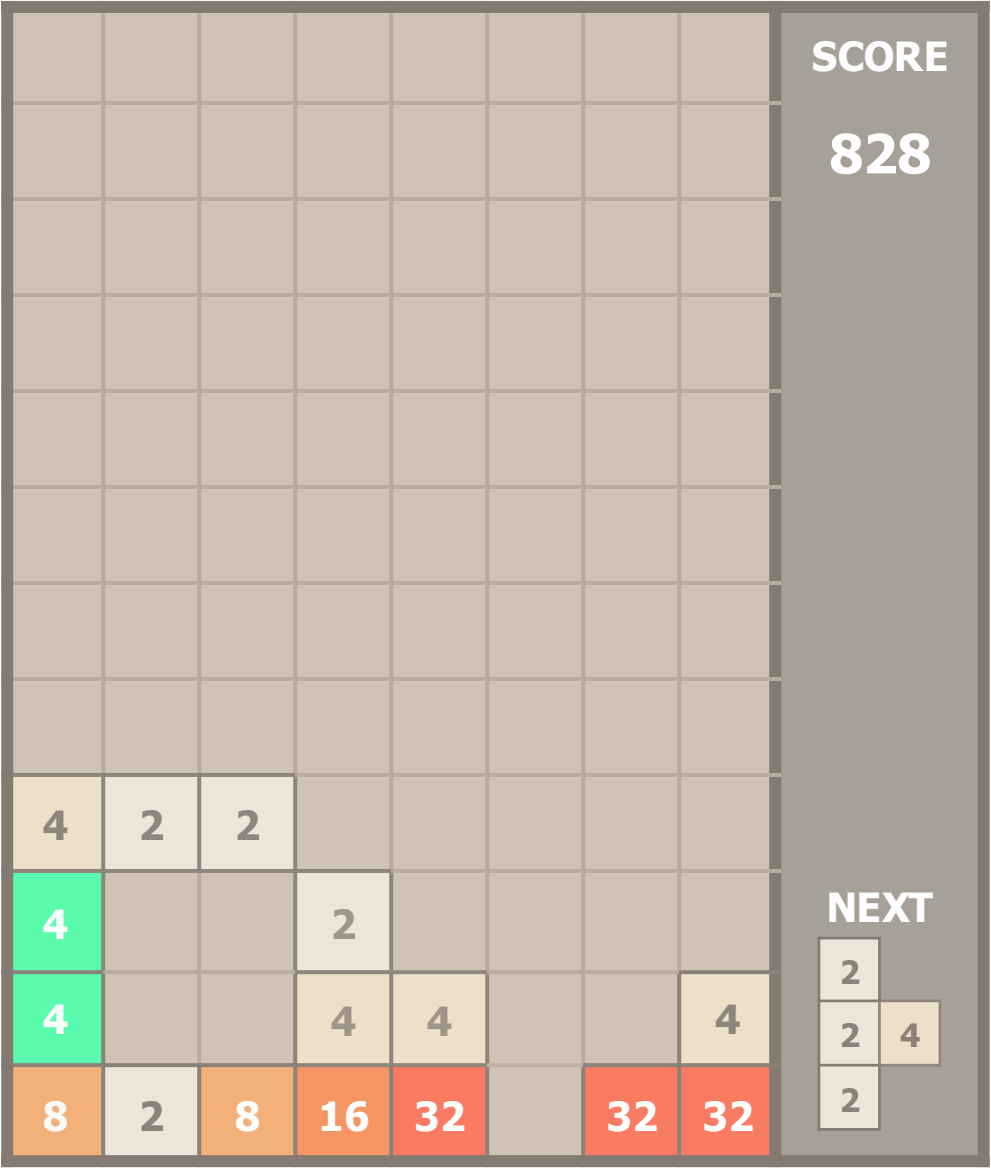
Chain Merging



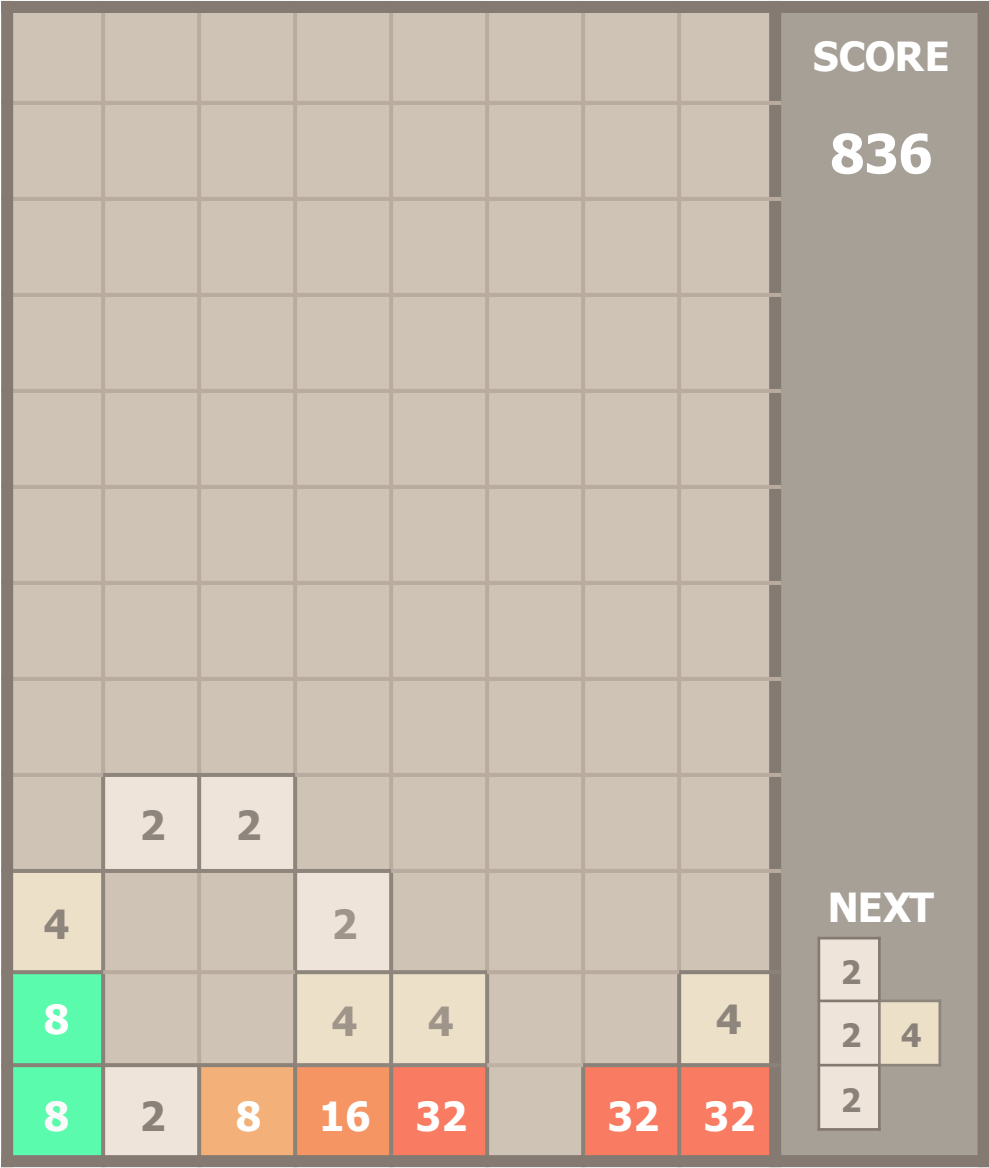
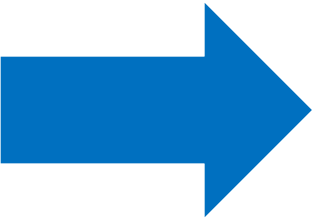
Chain merging is performed from bottom to top



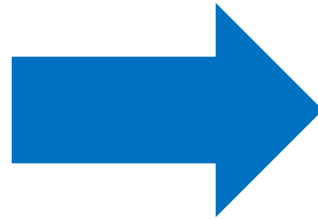
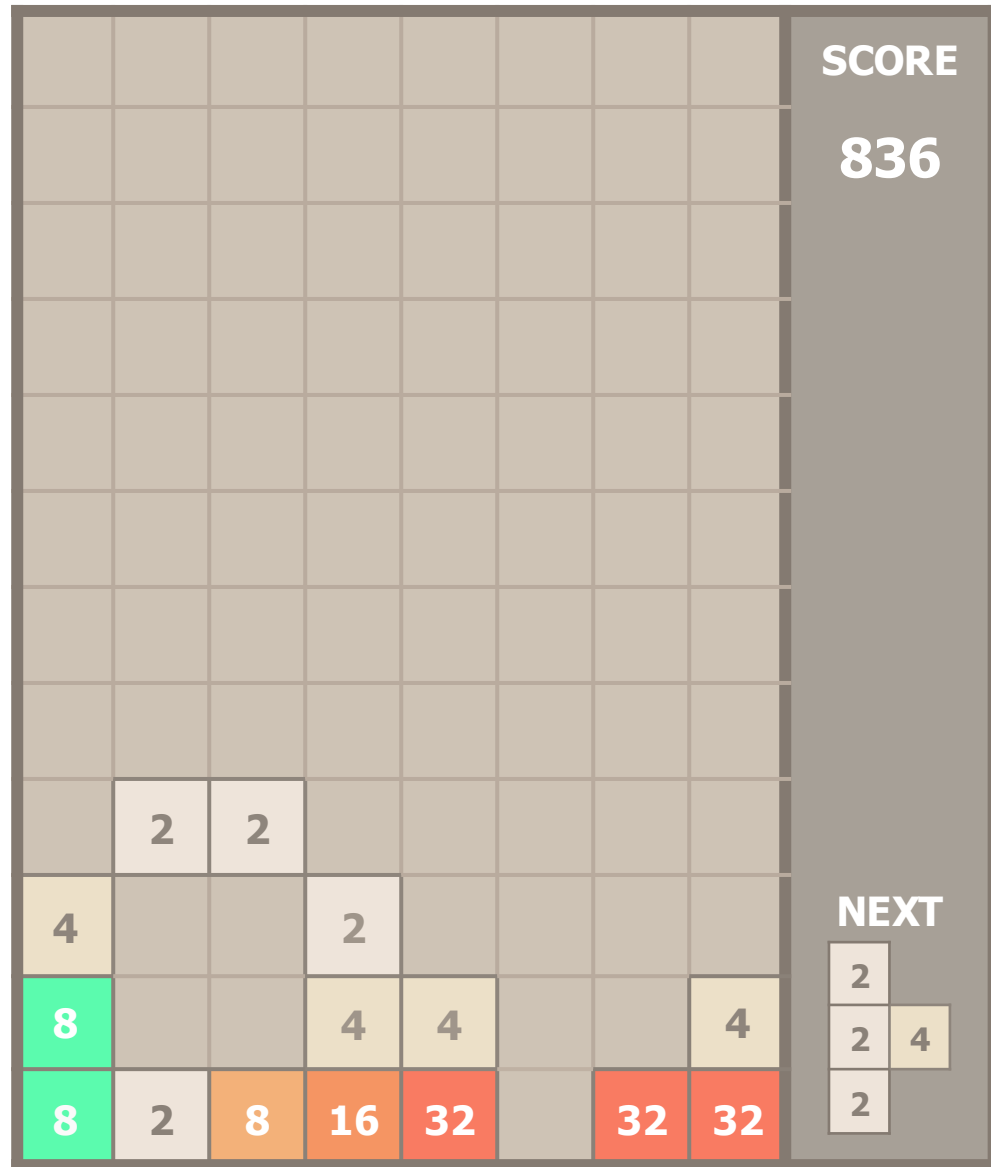
Chain Merging



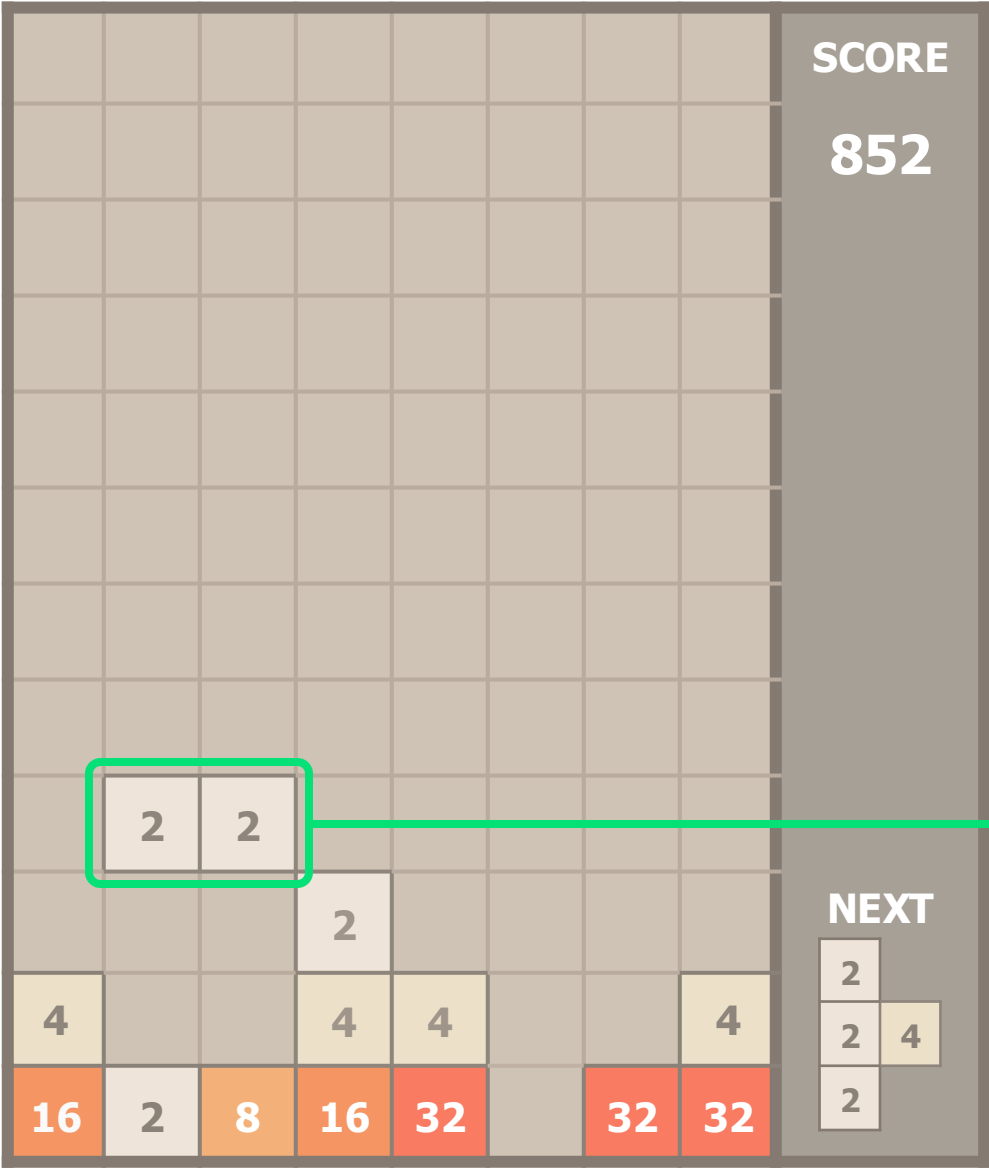
Chain Merging



Chain Merging



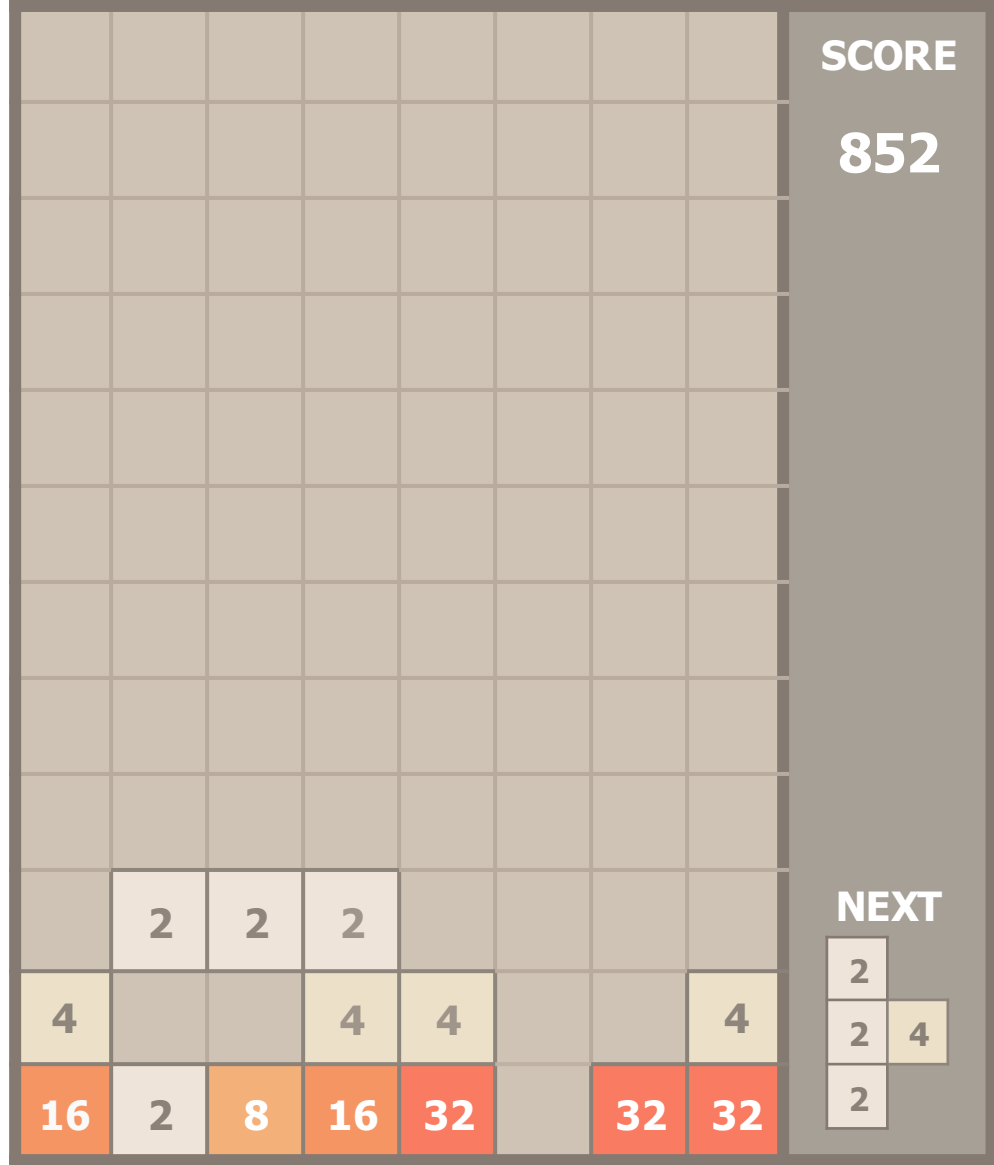
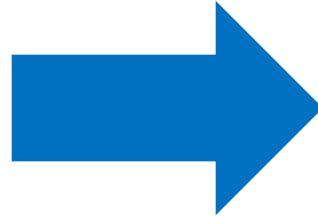
Handling Free (Unconnected) Tiles



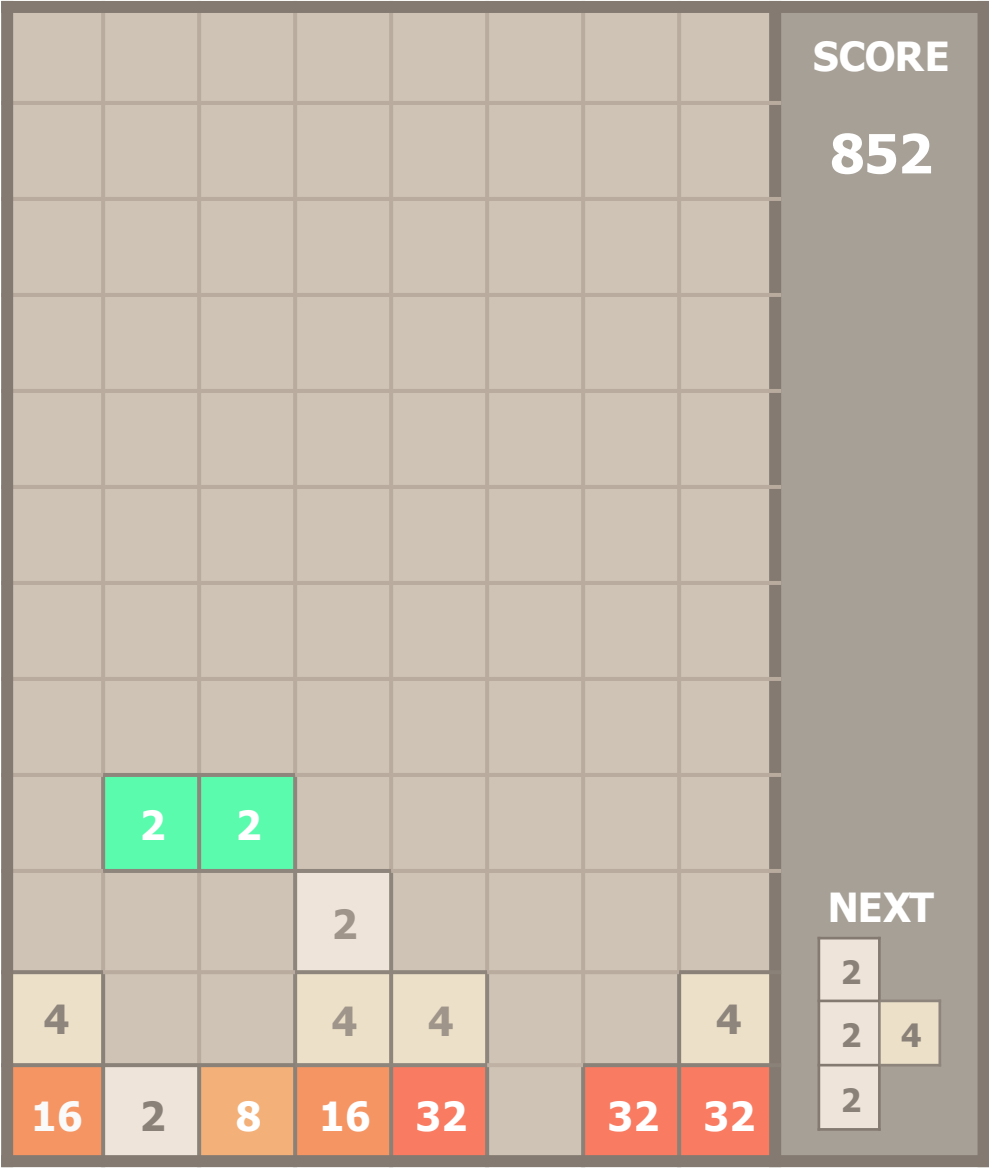
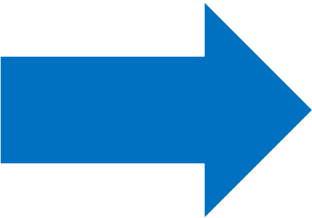
These tiles are not connected (4-connected) to the bottom of the grid or to the other tiles that are connected to the bottom. You can handle this situation by using one of the two different ways given below.

1. by moving each free tile down until it becomes connected (4-connected) with another tile that is connected to the bottom of the grid or it reaches the bottom (**harder to implement**)
2. by deleting each free tile and adding its number to the score (**easier to implement**)

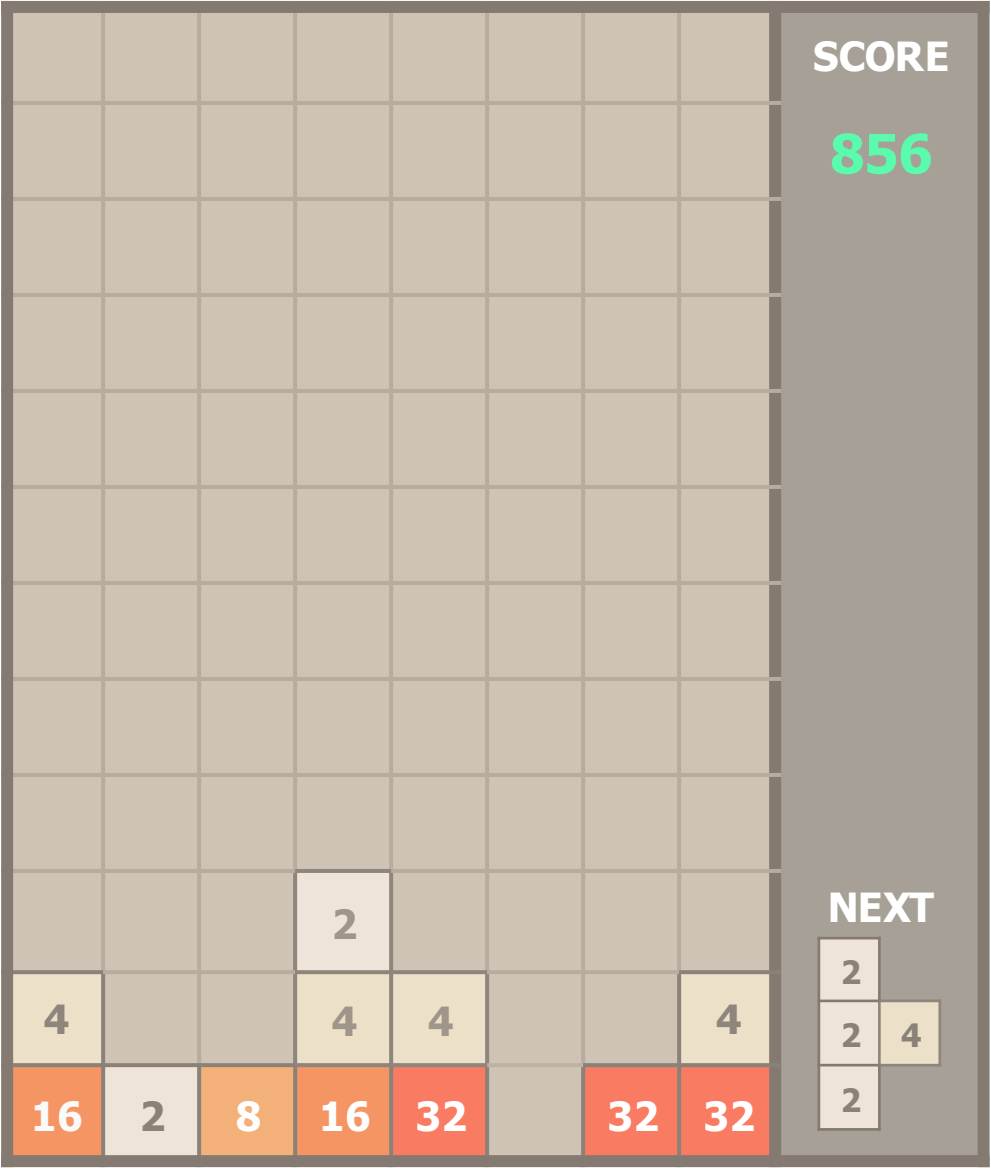
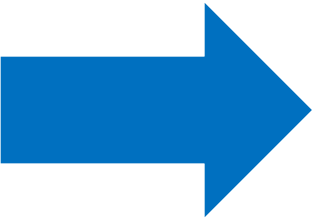
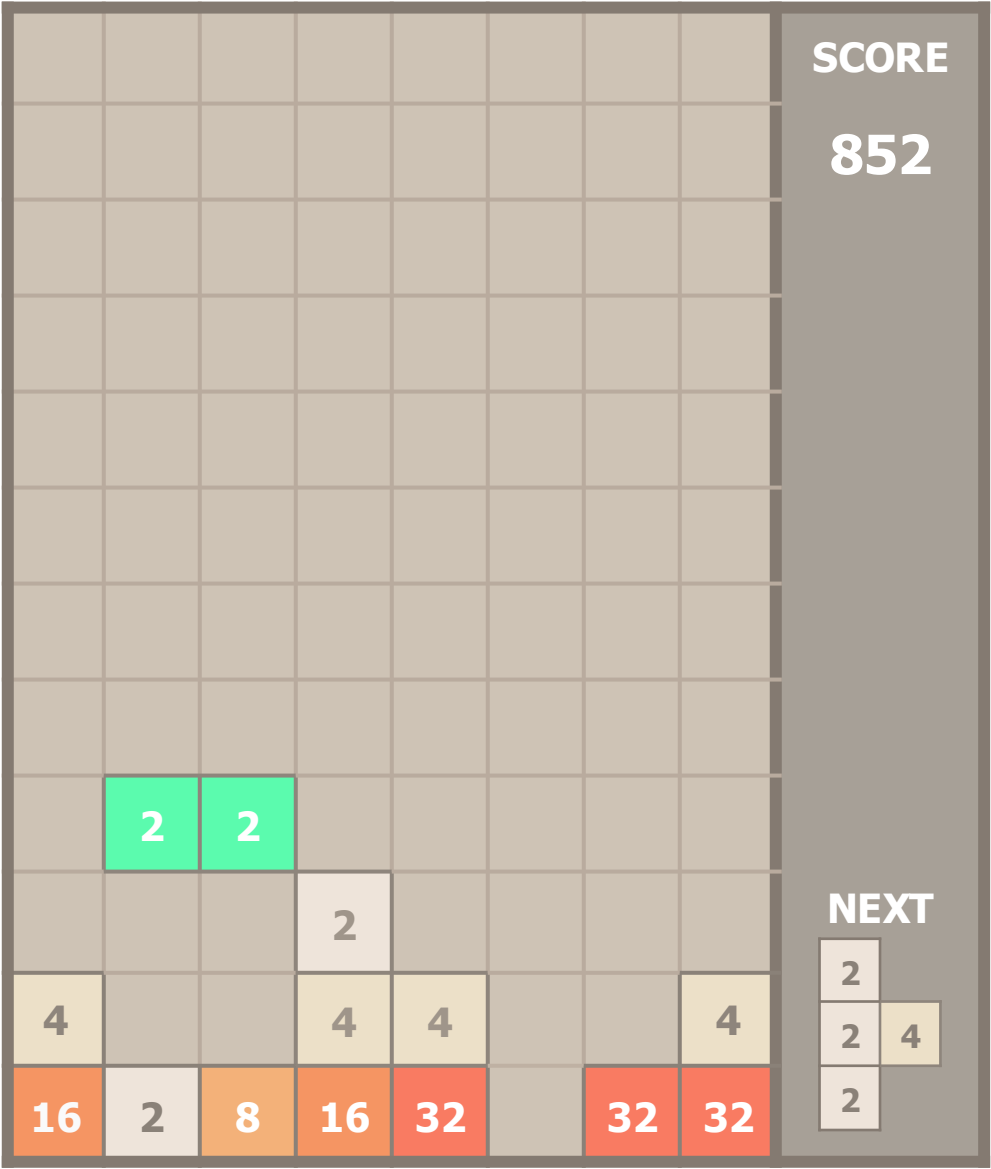
1 – Moving Each Free Tile Down Until It Becomes 4-Connected to the Bottom



2 - Deleting Each Free Tile and Adding Its Number to the Score



2 - Deleting Each Free Tile and Adding Its Number to the Score



About the Project

Project 2

- This a group project.
 - Each group will consist of **3 students**.
 - **GitHub usage is essential.**
- **Python 3** will be used as the programming language for this project.
- **You have to use the given base code as a starting point for this project.**
- In addition to the **stdraw**, the **color** and the **picture** modules **provided within the base code**, you can use only the basic Python modules such as:
 - **math**, **numpy**, **random**, **copy**, **enum** and **os**
- **Do not use any advanced module** such as **pygame** in your solution.

Project 2 – Suggested Calendar

- **March 24 Week:**
 - Assignment of the project along with the base code
 - Run the base code by using a Python 3 IDE
 - Understand the base code and how to use the stddraw module in Python
- **March 31 Week:**
 - Design and implement Tetris (without the 2048 part)
- **April 7 Week:**
 - Design and implement Tetris 2048
- **April 14 Week:**
 - Design and implement Tetris 2048

Submission Deadline and Deliverables

- **Submission Deadline:** Sunday April 20, 2025, at 23:59
- **Deliverables:**
 - **Commented Python codes** with an additional text file that contains **the link for your project on GitHub**
 - **A report** in which you present your work with the following contents
 - A cover page
 - An abstract
 - A description of the project
 - A description of your solution
 - Clearly explain your code design (Use UML diagrams etc.)
 - Show and explain the classes, the methods, the code blocks you added/modified, and the data structures you used
 - Include example screenshots from your program (with explanations) that show how your program works
 - **A clear statement about the division of the responsibilities within the team**
 - A list of your achievements (What did you learn during the project?)
 - A list of the references (If you have any)
 - **A video presentation (with a demo)** in which you summarize and showcase your work on this project
 - **Each team member must explain the parts that have been developed by him/her**
 - **A self-evaluation rubric** in which you evaluate your project

Project 2 – Evaluation Rubric

| Tetris (40 + 10) | | | | | | | | | | |
|------------------|--------------------------------|---------------------------|-------------------|--------|-----------|-----------------|------------------|--------------------------------|------------------|------------|
| 1 | 5 | 1 | 1 | 10 | 5 | Bonus: 10 | 3 | 1 | 10 | 3 |
| Game Grid | Tetrominoes (Different Shapes) | Move Down (Button & Auto) | Move Left & Right | Rotate | Hard Drop | Show Next Piece | New Random Piece | Update Grid (Landed Tetromino) | Clear Full Lines | Game Cycle |

| 2048 (40 + 10) | | | | | | |
|----------------|-------------|------------------|--------------------------------|---------------|---------------|---------------------|
| 5 | 5 | 5 | 5 | 10 | 10 | Bonus: 10 |
| Tile Numbers | Tile Colors | New Random Piece | Update Grid (Landed Tetromino) | Merging Tiles | Chain Merging | Handling Free Tiles |

| Other Coding Items (20 + 10) | | | |
|---------------------------------|--------------|--------------------------|-----------------------|
| Score | Game Over | Menu | Bonus |
| 7 | 3 | 10 | 10 |
| Computing and showing the score | Win and Loss | Pause, Restart, GUI etc. | Adding Extra Features |

| | | | |
|--------------|--------------|------|--------|
| 80% | 10% | | 10% |
| Coding Total | Video | | Report |
| | 5% | 5% | |
| | Presentation | Demo | |

Questions?