

The final product would essentially be the popular puzzle game 2048 as described below:

- Tile with the number 2 or 4 generated at random location every turn
- Arrows shift tiles in that direction and combines those with the same number
- Board is filled → no more moves → GAME OVER
- 2048 attained → WIN, option to continue

Visuals

MAIN MENU:



GAME:



The game board would look somewhat similar to the [original version](#) as shown above.

Features

- Tile collision
 - Logic of shifting tiles and combining identical ones
- Colour changing tiles
- Game board
- Timer
- Scoring system + Highscore
- Music
- Restart button
- Main menu
 - Play
 - Instructions
 - Quit
 - Toggle music

User Controls

- Mouse to click buttons such as start, pause, quit
- WASD and Arrow Keys to control movement of the pieces
- Q to quit
- R to restart

End Condition

- Win: user reaches 2048, shown with win screen and given option to continue
- Lose: game board fills before the user reaches 2048, no more moves
- Can quit by pressing Q or returning to main menu and using quit button

Timeline

Estimated Date of Completion	Feature / Task	Estimated Development Time
May 28, 2023	Tile collision (logic behind it)	3 days
May 30, 2023	End calculations	2 days
Jun 1, 2023	Score conditions	2 days
Jun 5, 2023	ALPHA PROGRAM — minimal text-based version	
Jun 9, 2023	Game board + actual GUI tiles	3 days
Jun 10, 2023	Score + high score + timer + restart	1 day
Jun 11, 2023	Main menu + instructions	1 day
Jun 12, 2023	Music	1 day
Jun 14, 2023	BETA PROGRAM — essentially finished version	
Jun 16, 2023	Testing + fix loopholes	4 days
Jun 19, 2023	FINAL PRODUCT — finished version	
Jun 19, 2023	PRESENTATION	

UML Diagrams

GameFrame
-panel: GamePanel
GameFrame()

GamePanel
<u>-GAME_WIDTH: int</u> <u>-GAME_HEIGHT: int</u> -gameThread: Thread -image: Image -graphics: Graphics -tiles: Tiles -state: GameState
GamePanel() +paint(g: Graphics) : void -draw(g: Graphics) : void -drawTitle(g: Graphics) : void -drawGame(g: Graphics) : void +run(ticks: int) : void +keyTyped(e: KeyEvent) : void +keyPressed(e: KeyEvent) : void +keyReleased(e: KeyEvent) : void

Tiles
-board: Tile[][] -BOARD_SIZE: int -score: int
Tiles(size: int) +reverse() : void +rotateCCW() : void +rotateCW() : void +left() : void +right() : void +up() : void +down() : void +keyPressed(e: KeyEvent) : void +keyReleased(e: KeyEvent) : void +getScore() : int +draw(g: Graphics) : void

<<enumeration>> GameState
Menu Play GameOver Win

Tile
<u>-TILE_SIZE: int</u> <u>-TILE_SPACING: int</u> +value: int
Tile(size: int, spacing: int, v: int) +draw(g: Graphics) : void