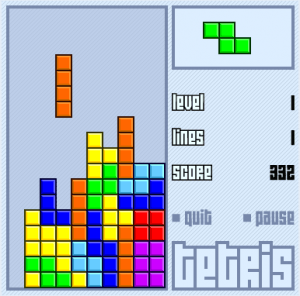
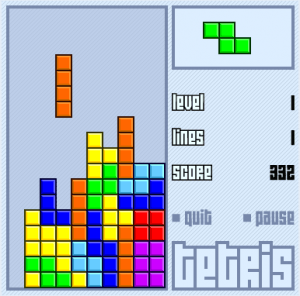
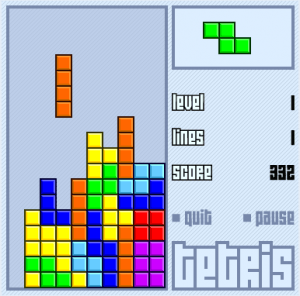
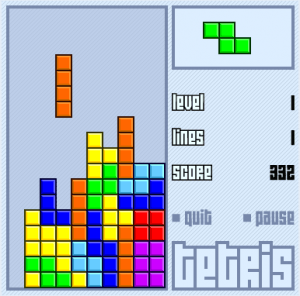
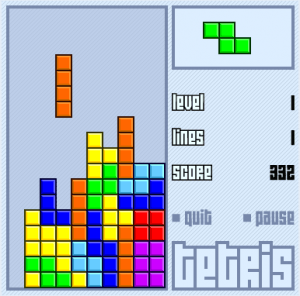
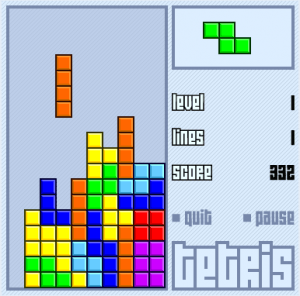
**Final Project – Tetris**

Fig. 1



You have chosen to recreate the video-game Tetris. The object of this game is to fill up horizontal rows (leaving no gaps) using the shapes the game provides while avoiding the top of the Tetris board.

Your program should start by displaying the empty Tetris board, a score section, and an area showing the next Tetris piece the game will give to the user. It should also contain a menu system that allows the user to start a new game, display the top 5 high scores, exit the game, display an About section, and provide instructions on how to play the game.

Potential screen during gameplay

**How to Play**

The user is presented with an empty Tetris board, an empty score board, an empty ‘Next Tetris piece’ section, an enabled ‘Start Game’ button, and a disabled ‘Pause’ button. The user can start the game by clicking the ‘Start’ button and has an option to pause the game using the ‘Pause’ button once the game is already started. The game then randomly gives the user one of the 7 predetermined Tetris pieces (Shown in Fig. 2) that appear at the top of the Tetris board and will fall in a straight down in real time. The user then has the capability to manipulate the landing area and orientation of the Tetris piece, as well as the speed in which the Tetris piece falls (Shown in Fig. 3) using the arrow keys. The user will be given another Tetris piece and awarded points once the piece already on the screen has stopped falling. If the user has filled a horizontal row completely, the row will be erased and points will be given. Every block placed above the row will fall down by the number of spaces cleared.

Fig. 2

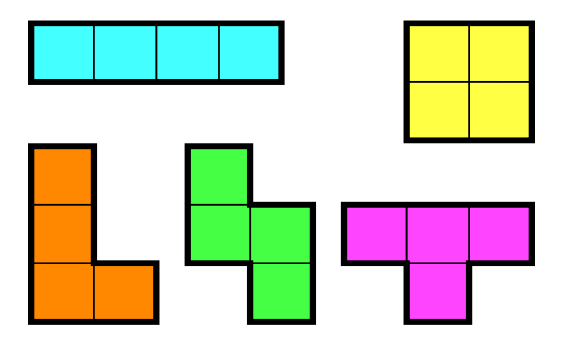
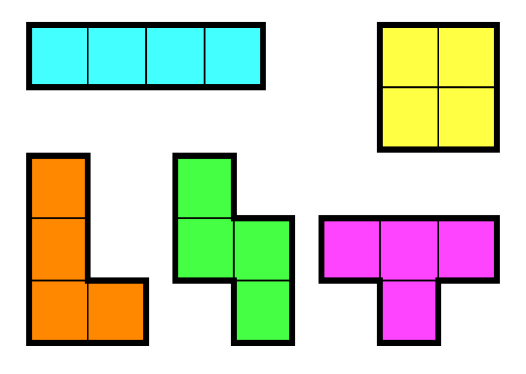
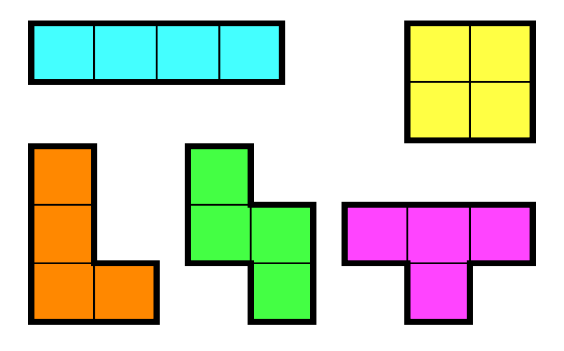
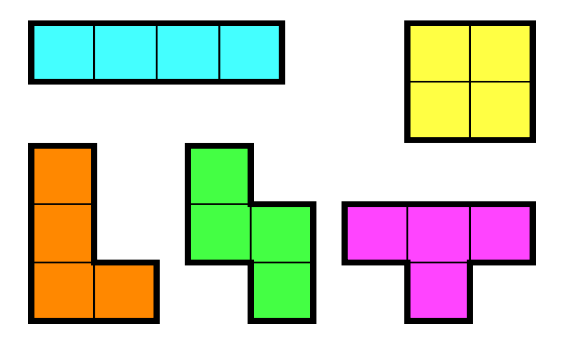
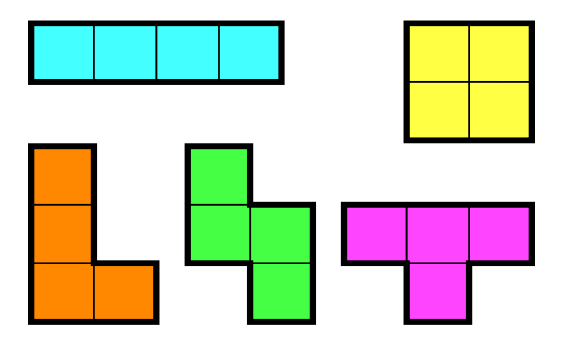
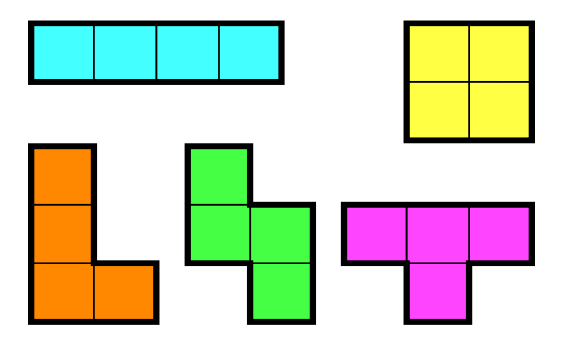
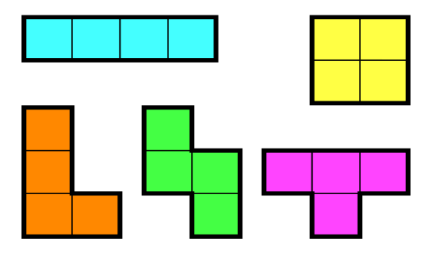
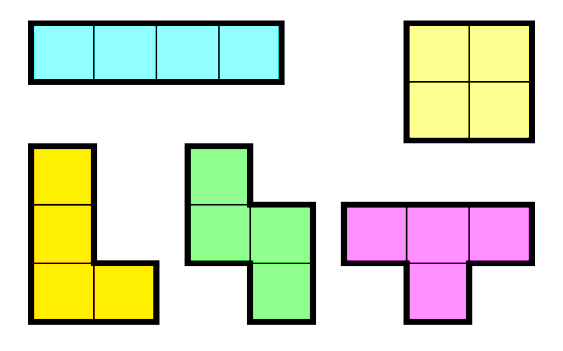
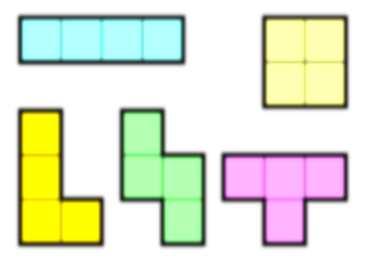


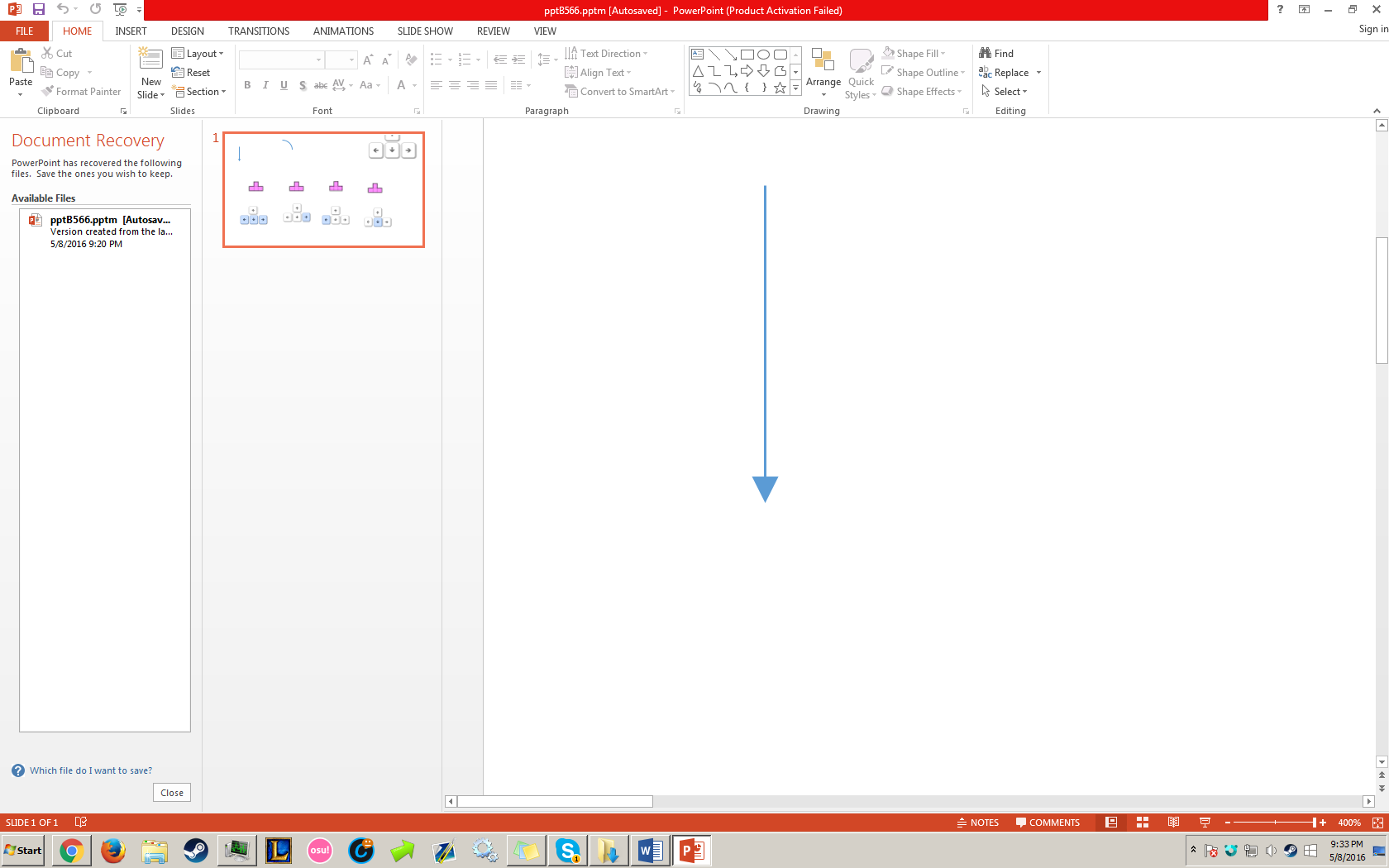
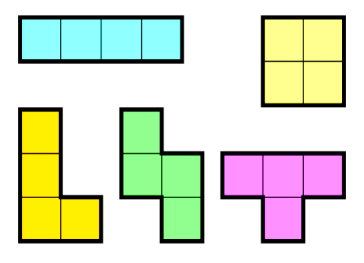
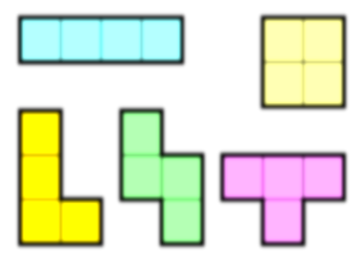
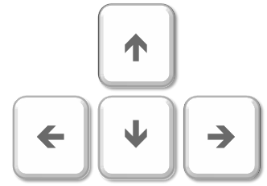
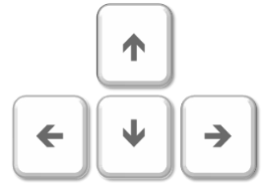
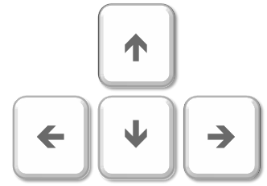
Fig. 3



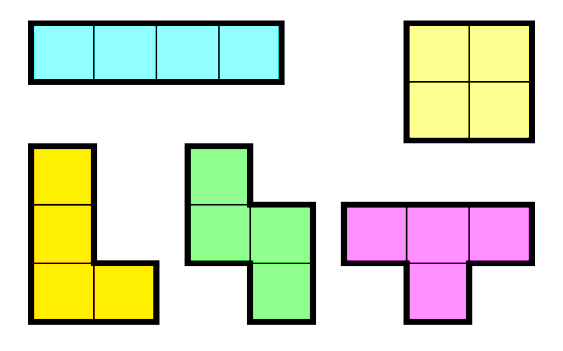
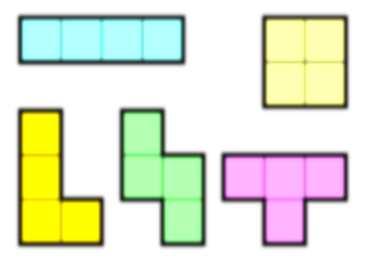
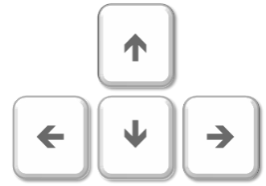
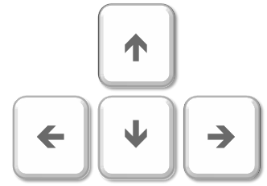
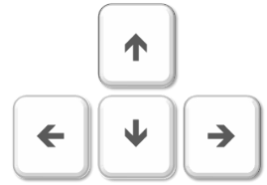
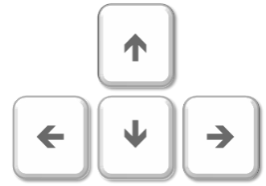
**SPACE**



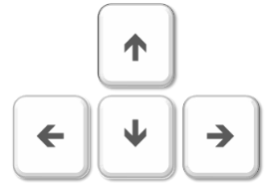
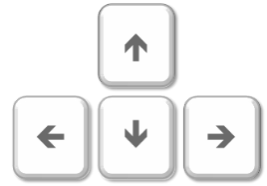
Hard Fall



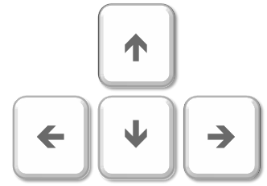
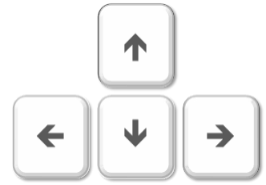
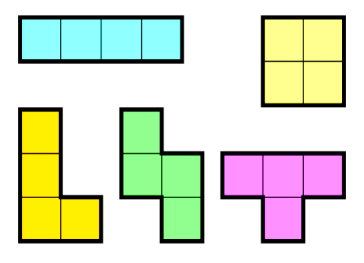
Rotate Piece



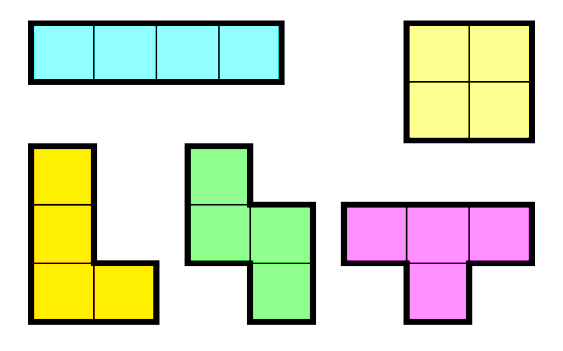
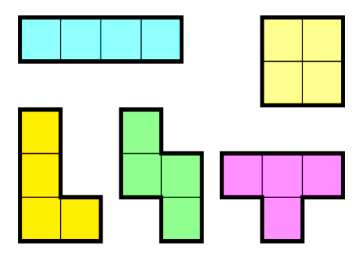
Accelerate



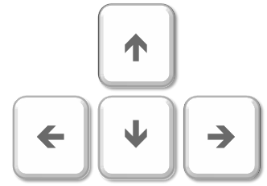
Shift Left



Shift Right



Controls



**Rules to be Implemented**

**1.** The player cannot pause the game without starting the game  
**2.** The player must strive to avoid the top of the Tetris board. If the top of the Tetris board is met with a Tetris piece, the game will be over.  
**3.** The game once beaten or lost will check to see if the score achieved is higher than any of the top 5 high scores already recorded. If so, the high score board will be refreshed to include the recently achieved high score.  
**4.** The player’s controls are limited to the left and right arrow keys for the shifting of the Tetris piece, up for the clockwise rotation, down for accelerated fall, and space for a hard fall of the Tetris piece. (This is shown in Fig. 3)  
**5.** The game is limited to 7 predetermined Tetris pieces (As seen in Fig. 2) that will be given to the user at random.  
**6.** Already placed Tetris pieces above cleared rows will not fall further than the amount of rows cleared by the user.

**Scoring**

The user will be rewarded with points if they have successfully completed a row, and have placed a Tetris piece on the board. The scoring for these actions are shown in Fig. 4.

Fig. 4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Points Given for # of Lines Cleared at *n* Levels. | | | | |
| Level | 1 Lines Cleared | 2 Lines Cleared | 3 Lines Cleared | 4 Lines Cleared |
| 0 | 50 | 100 | 300 | 1200 |
| 1 | 100 | 200 | 600 | 2400 |
| 2 | 150 | 300 | 900  **Due Date:**  **First Version: \_\_\_\_\_\_\_\_\_\_\_\_**  **Final Version: \_\_\_\_\_\_\_\_\_\_\_\_** | 3600 |
| 3 | 200 | 400 | 1200 | 4800 |
| … | … | … | … | … |
| *n* | 50 \* (*n* + 1) | 100 \* (*n* + 1) | 300 \* (*n* + 1) | 1200 \* (*n* + 1) |

Points for each block placed will be calculated using the formula *points* = 10 \* (*n* + 1), where *points* is the amount of points to be given and *n* is the current level.

**Evaluation**Two versions of the game will be handed in on distinct due dates and will contain the following.

**First Version**  
 - Completed Menu  
 - Game (row completion, fast placing, Rotation of Tetris piece)  
 - New game, exit game, and pause game options

**Due Date:**

**First Version: \_\_\_\_\_\_\_\_\_\_\_\_**

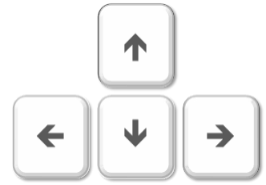
**Final Version: \_\_\_\_\_\_\_\_\_\_\_\_**

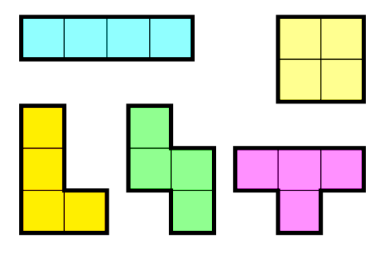
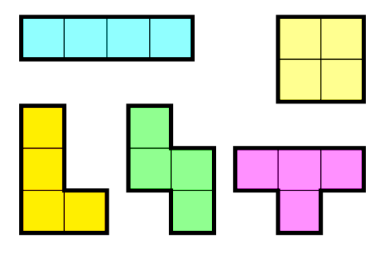
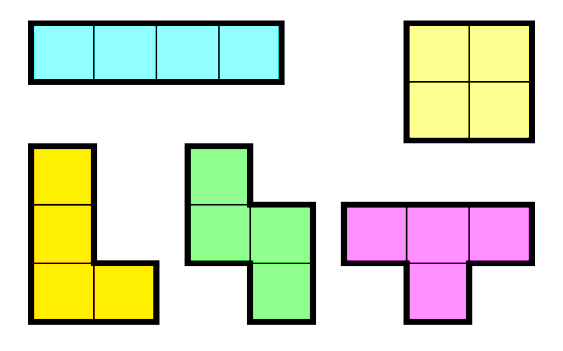
**Final Version**  
 - Previous features  
 - Scoring system  
 - High score tracking  
  
**Additional Notes**  
 - You may add any additional features that you feel are appropriate to include  
 - Project must be handed in on time to receive full marks.  
 - Hand-in design, program code, and user manual (5 – 10 pages, fully illustrated) are to be  
 handed in as well

**Due Date:**

**First Version: May 19-20, 2016**

**Final Version: June 2-3, 2016**







**SPACE**



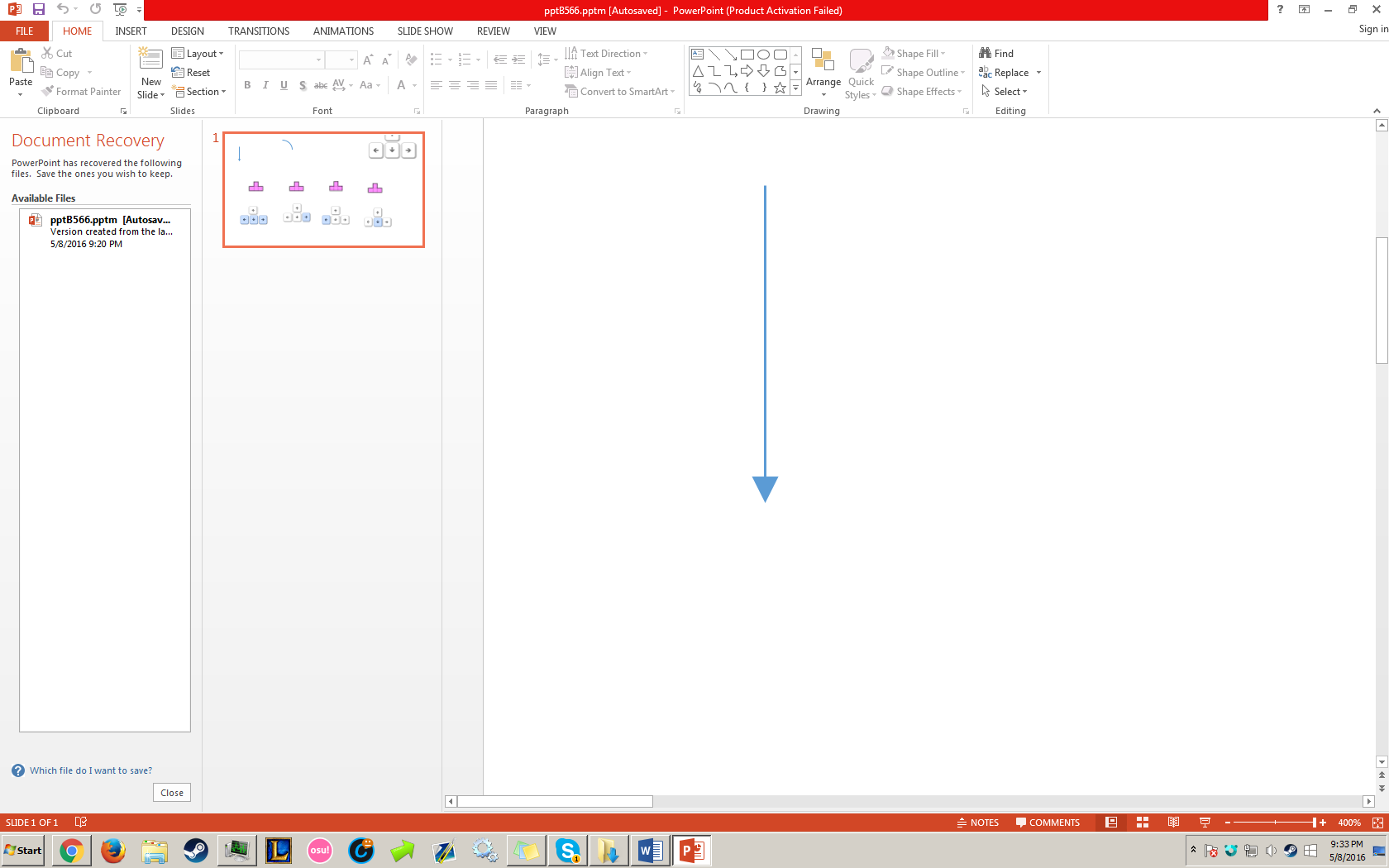
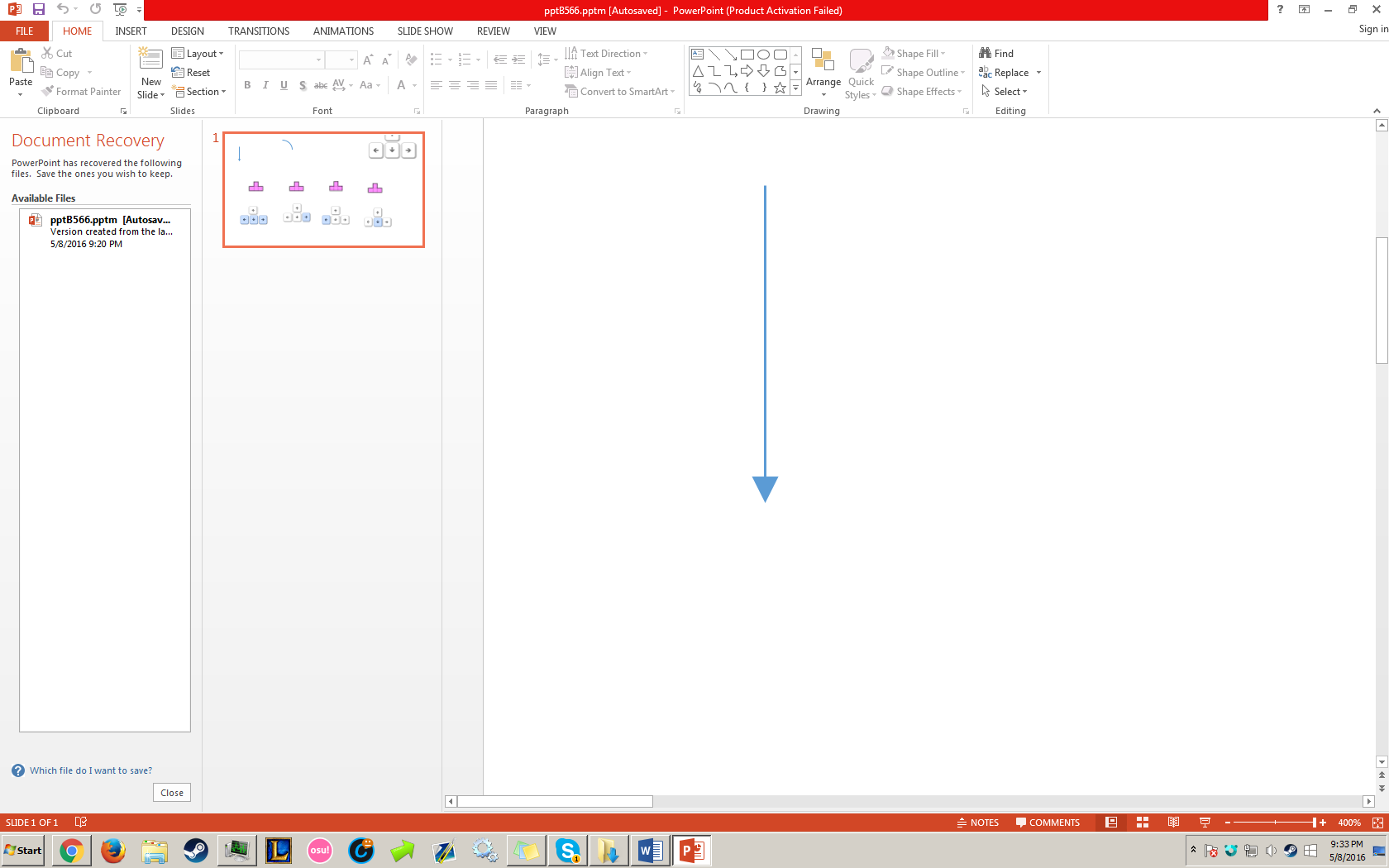
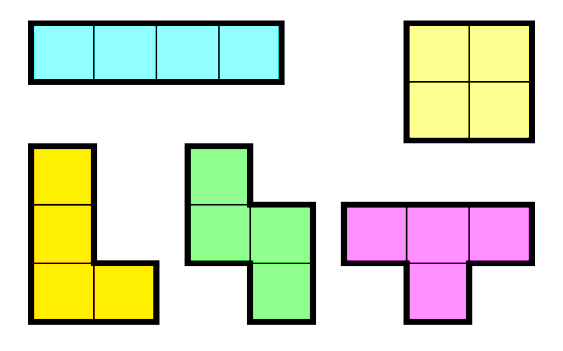
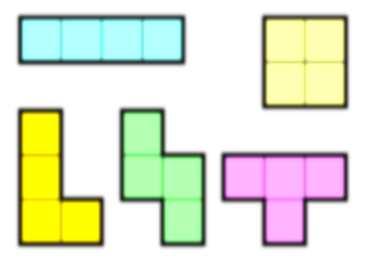
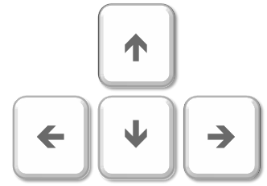
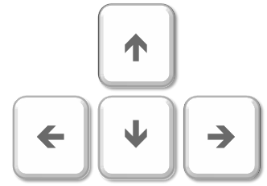
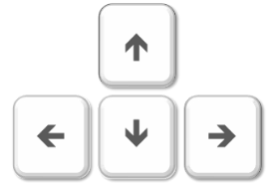
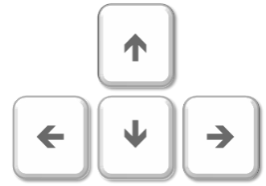
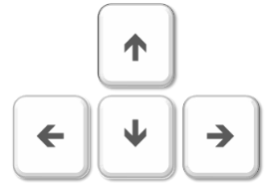
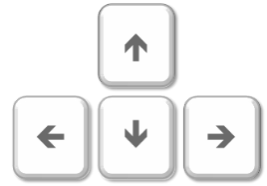
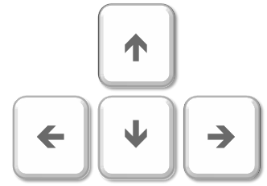
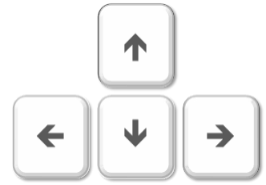
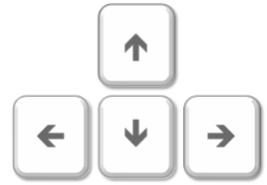
**SPACE**



**SPACE**



**SPACE**

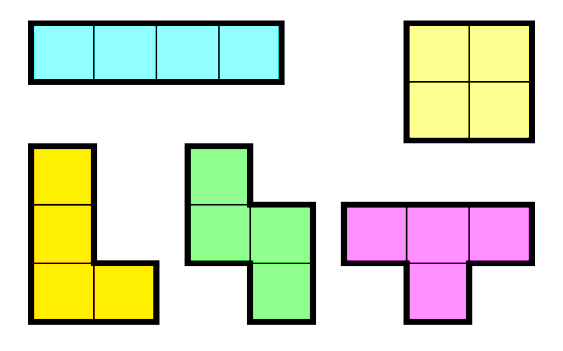
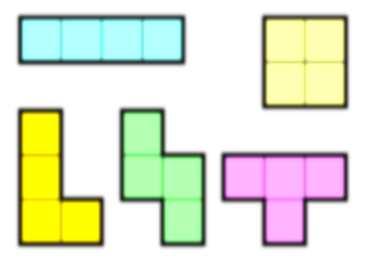
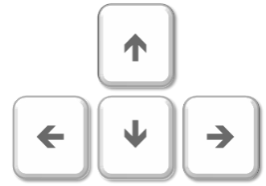
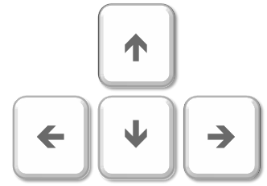
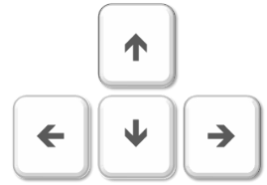
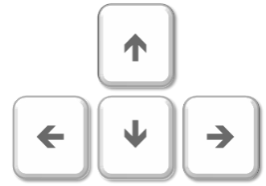


Controls

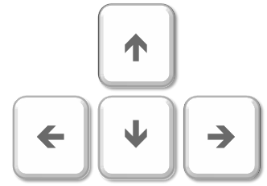
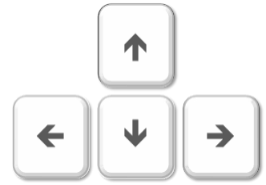
Rotate Right

Accelerate

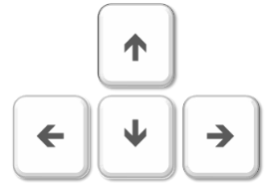
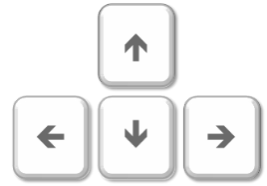
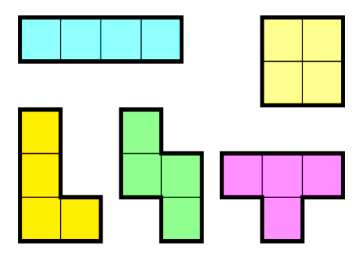
Rotate Left



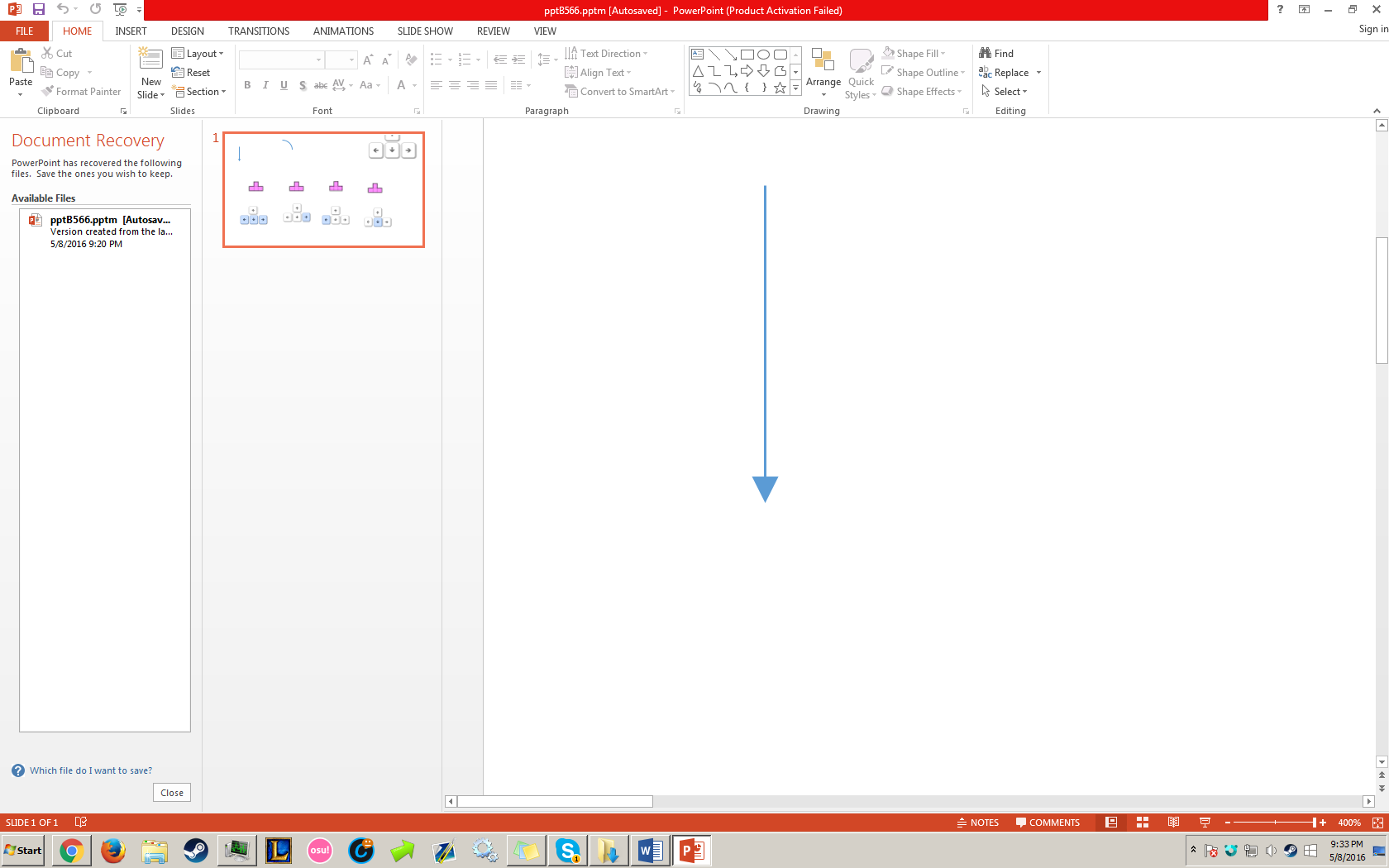
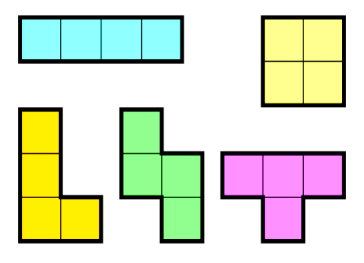
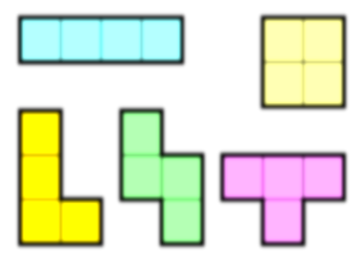
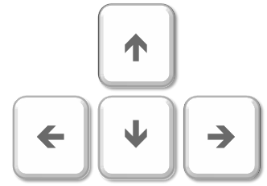
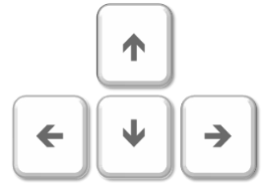
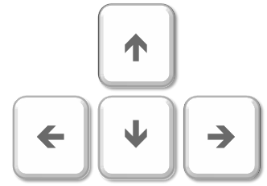
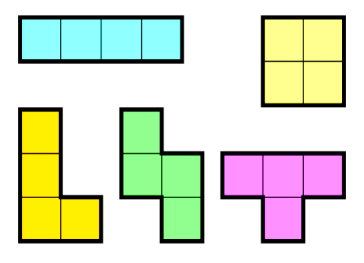
Accelerate



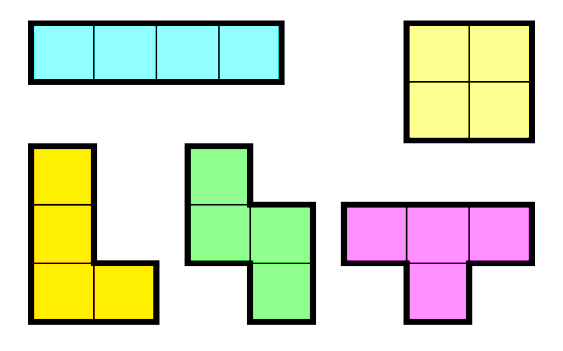
Shift Right



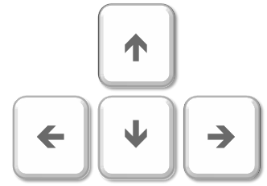
Shift Left



Rotate Piece



Controls



**SPACE**