Approach A requirements (outlined by project commissioner)

- Standard Tetris game functionality: given random pieces, use keyboard to move and rotate
- Row clearing functionality
- Score keeping functionality: current score, high score table
- The same shape should not appear 3 times in a row
- Requirements on processing and memory consumption: able to work on a 1 GHz CPU and take up < 200MB of memory

Approach A requirements (broken down in more detail by software engineer)

Functional requirements

1. Game space: separate window, showing a 10x20 grid with square pieces able to occupy spaces in the grid
2. Game functionality: tetromino pieces occupy spaces, and able to be moved and rotated by keyboard inputs
3. Game loop: tetromino pieces fall, new pieces generated randomly when a piece is solidified
4. Collision detection: pieces can't pass through each other, pieces stop and solidify when they collide with something below them
5. Game over: when the spawned in tetromino is colliding with a solidified piece
6. Row clearing functionality: when a row is full it is cleared, the score counter is incremented, and all solidified pieces move down by 1
7. Score keeping and high score table
8. Randomiser: the same piece is not generated 3 times consecutively

Non-functional (quality) requirements

9. Able to work on a 1GHz CPU (i.e. run smoothly and efficiently)
10. Take up no more than 200 MB of memory