1 INTRODUCTION

Tetris is a popular block game. The main components of the game are variously shaped tiles, which the player must stack on top of each other so as to create as few gaps as possible between the tiles. Once a line is completely filled with tiles, it is cleared away so as to make space for new tiles. The tiles drop from the top of the screen and lands at the bottom. The objective is to keep the game going for as long as possible without filling up the entire screen. In other words, the player must try to clear as many lines as possible. Points are awarded for lines cleared.

1.1 TILES

There are six different possible shapes of tiles. Tiles can be I-shaped, J-shaped, L-shaped, O-shaped, S-shaped, T-shaped, or Z-shaped.

![Tile shapes](Google Images)

Figure 1.1: Tile shapes [Google Images]

1.2 HOW TO PLAY

The game is played with a single player at a time. Players get individual scores, and the winner is the player with the higher score.
The game begins with a blank game board. The tiles descend from the top of the screen, one at a time. The player will be able to move the tiles horizontally across the screen. The player will also be able to rotate the tiles so that the best orientation can be found. Once a single line is covered with tiles, that line will be cleared from the screen. There is limited space on the game board, so the player must try to create as few gaps as possible between tiles.

The game is over when the game board is completely filled, so that no more tiles can be added. At the end of the game, the cumulative score is based on how many lines the player was able to clear.

![Game demonstration](Google Images)

Figure 1.2: Game demonstration [Google Images]

## 2 Implementation

I will have different classes for the game board and the tiles. I will create the different-shaped tiles using multi-dimensional arrays. The tiles will be moved across the screen and rotated by using a translate function. The different blocks will be stored in different classes and accessed as needed. The game board will be a separate class. I will create methods to initialize the game board, and store the different blocks that the player receives in the course of the game. I will check for possible moves and clear lines when a line is completed.

I will also have a method to keep track of the player's score. The score will be updated every time a new line is cleared. I will also need a method to randomize the order of the different shapes of tiles that the player will receive.

The player will be able to move and rotate pieces and quit the game using different keys, so the game must allow keyboard input.

### 2.1 Possible Enhancements

Apart from the basic requirements of the game, I will try to add some features to make the game more interesting.
One possible enhancement can be increasing the level of difficulty of the game as time progresses. This can be done by increasing the speed at which the tiles descend to the bottom of the game board. The player will need to make quicker moves to place the tiles. Another way to enhance the difficulty is to make the tiles automatically descend to the bottom once the player has had about 10 seconds to maneuver the tile and change its orientation.

After the player has reached a certain score, there can also be a bonus round where the tiles descend slower, and the player has more control.

Another enhancement can be imposing a time limit. The game will be over after the player has had about 10 minutes. The objective of the game would be to attain the highest possible score before the time is over.

2.2 Difficulties

I assume that one difficulty will be learning how to rotate the tiles or change their orientation. It might also be difficult to enhance the game by speeding the movement of the tiles after a certain amount of time has passed. An overall challenge will be to keep track of the state of the game and the scores as lines are cleared and new tiles are introduced.