

Design Rational

Reasoning:

The custom Tetromino piece that I created for my project contains a total of 10 blocks and is in the formation of an “E” like the letter in Tetris. The reason why I designed it in this way, was not just because it fits with name of the game, but also the interesting interaction it has with both other tetromino pieces, and the game board. Because this custom piece is large compared to the rest, it comes to no surprise that misplacing it or having a tough placement will cause immense punishment for the player. However, this piece still has its advantages. Because of the gaps inside of the tetromino, pieces like “L”, “J”, “T”, and “I” fits into it no problem. With the right pieces you can clear 3 lines faster than your average Tetris pieces. Also, this piece can be used when lines that where cleared leave behind small block. The “I” tetromino would fail to fill in all the gaps, but the custom one would not only fill one space, but two at the same time.

Gameplay:

The custom gameplay experience that I added was a bonus to any line cleared with the “E” tetromino and a 2-min timer to achieve as much points as possible within a small-time frame. This gameplay highlights the pieces potential as this not only allow players to gain more points per custom block used in a short time frame but also allows players to have a dominant strategy when using this piece. The player could play it safe and turn it horizontal (1500 points maximum) or make it harder and gain even more points by turning it vertical (2500 points maximum, almost double). To put it into comparison, an average line clear is only 100 points. This means that a player could fill 14 lines and spend more time only to have less points that a player who had one “E” tetromino cleared in a shorter time.

This game mode makes the custom piece the most valuable and desired piece in the entire game, but it requires skill to use it effectively due to its interesting arrangement of blocks.

Challenges:

While implementing the piece, there were a couple challenges I came across. The first one being the rotation of the piece. Because of the sheer size of the tetromino, when rotated the blocks would push through the border and become soft locked. Crashing the game and making it unplayable. To solve this problem I implemented a code where it would look at all the child objects in the parent one and check each and every one could rotate without going past the border. If it couldn't, rotate it back. Another problem was the line clearing. While the "I" Tetromino could clear four lines, the "E" could clear five, but because of the gap sometimes only two (top and bottom) lines would be cleared. This would cause the blocks to not fall and instead hover in the air. To solve this problem, I added a line of code where instead of checking every row if it can move down, it would also check every column to find every block on top of it to move down with it. Finally, the last problem I found implementing was the value of the piece. I found that if more than two of the same custom pieces was on the board, then the value drops significantly and almost causes the player to lose immediately because of the sheer size of the piece. To fix this problem, I reduced the probability of spawning that custom piece. Instead of 1/7 chance of spawning, it became 1/14 probability. The main reason for this was to reduce the amount of "E" tetromino pieces possible and to deal with how unity "random.range" works, causing the same piece to spawn multiple times.