**Block.cpp**

1. Block constructor now receives only side, it generates a random shape using getRandomShape function, need to decide about "hasReachedEnd" how we deal with it in order to consider this block reached the end, maybe update this field and then make it static
2. Fixed buildBlockPoints in the past in built weird shapes, now it works
3. Added a "canMove" Block function to check if the block can move (it doesn't og out of boundaries) we need to add a check with collision with other blocks – maybe if the char on the next position is not ' ' (whitespace) = collision with other blocks?, it may also work for the boundaries as boundaries are different than ' ' (white space)
4. Changed moveBlock to return bool instead of void so that if the block didn't move it will indicate the game that it's "stuck" and in case it's a drop movement and it's stuck it means the shape reached the end and now we need to make it static
5. Changed how the function works so it will firstly erase the symbols (leave a whitespace) from the screen and then draw the new symbols. Seems like we need to add an if statement indicating if this is the top boundary, it won't delete it, (gotoxy of this point, check if it's not a '-')
6. Added makeBlockStatic (we need to do it)
7. Added getRandomShape
8. Added isReachedEnd (we need to do it, I'm not sure it's the right approach because we're already checking if it's the end on "canMove")

**Game.cpp**

1. Created a new class of game so that the main will only have the startGame and be clean.
2. Creates the board and initializes with constructors.
3. Draws the boards (maybe we need to consider drawboard to draw both boards because I'm not sure someone will want to draw only one board..)
4. The bools are for the check if the block moved or not
5. The random seed generator is for the getRandomShape
6. Creates blocks according to left or right block (helps with the gotoxy location)
7. Moves the blocks with the while loop..
8. In case the block couldn't move down (DROP), need to make the block as static and then generate a new block.

**GameConfig.h**

1. Added more eKeys

**Main.cpp**

1. Made it more aestethic using game.cpp

**Point.cpp**

1. Added a set function (I'm not sure if I used it eventually)
2. Fixed the draw to have this->x instead of x, so we won't confuse it
3. Changed setMoveDirection, instead of bool function it sets the move direction and gets the next point (it also returns the next point as to where it should move)
4. Changed the move function to use setMoveDirectionAndGetNext and then move this point (canMove already checked if it's out of the boundaries)

**Board.cpp**

1. Yesterday I made the board as a static board on the screen and that's not the correct approach as the board should be a two dimensions array (I knew that but I was probably too tired to think about it) so I replaced it and built a two dimension array.
2. The constructor first initializes the board, and drawBoard just draws the board according to the array
3. About the pause and resume – I think when the player presses ESC (pause) what it should do – gotoxy of the starting location of left board, update the current state of the board into the two dimensional array and same goes for right board – gotoxy using the RIVAL\_POS and then update the current state of the board to the right board. The resume will do the opposite of the pause. Using the print board for left and then gotoxy rival and printboard for right. We just need to consider how we keep the position of the current shape that drops down.
4. The drawScore function is the function from yesterday, I'm not sure it should be like that but I kept it for now.

**Just a few more words..**

1. It draws the borders and the shape correctly
2. It moves the shapes correctly to the sides
3. It rotates the shapes correctly (except for 'L' and 'J' – need to understand why)
4. Currently it doesn't stop the shape from moving out of boundaries to left or right
5. It stops at the bottom border, this is the part where it should make it static and create new block.