

Practical Work #4

Problem: Tetris-like game

Emrin Angelov
Xavier Betis

Objectives

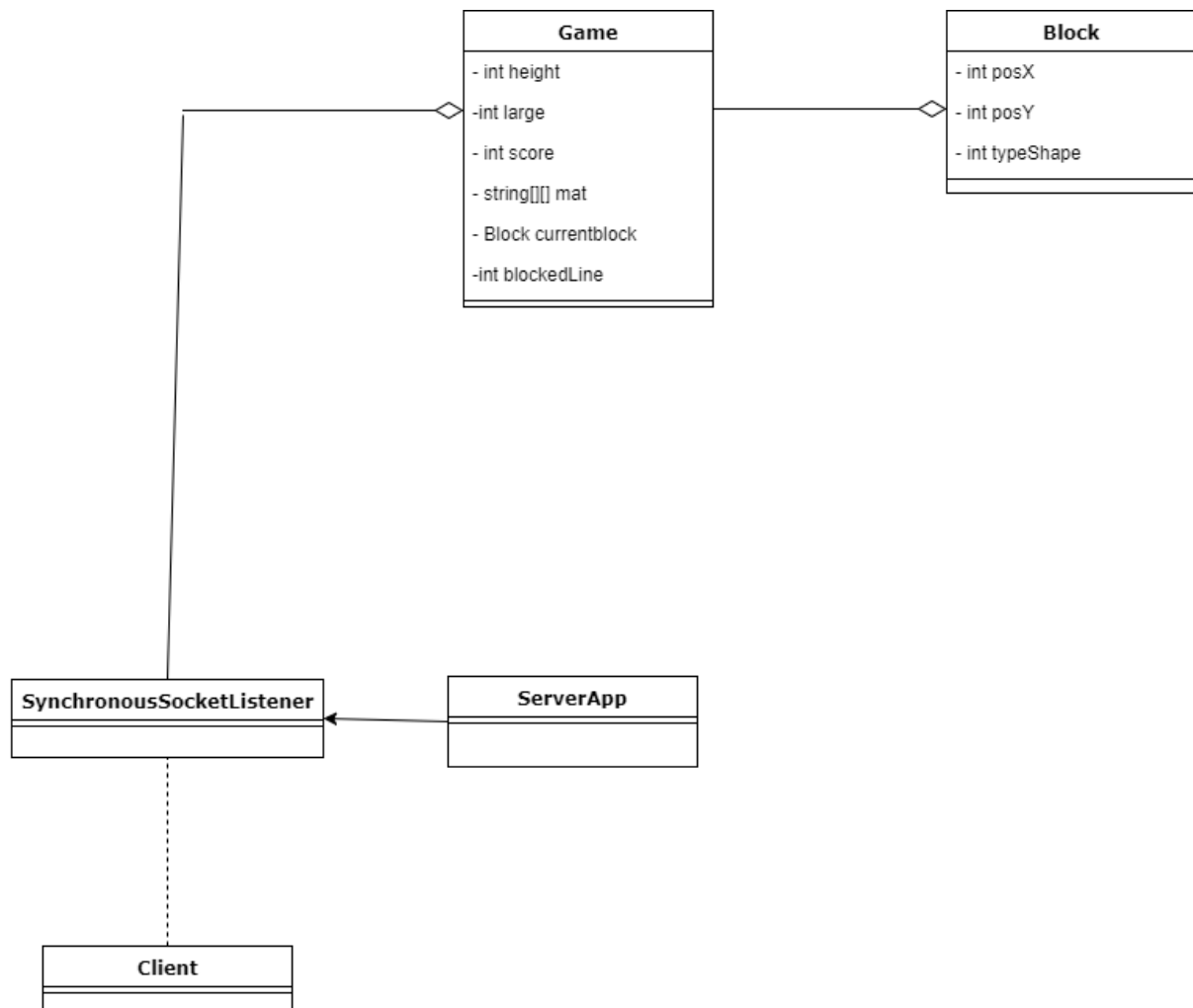
The aim of the problem is to achieve a small network game like Tetris. To achieve this, a client-server model will be used where the network layer will be provided by Sockets. On the server side, multi-client management will be provided by Threads.

Protocol and connection

There are two ways to launch both .exe files, the simple way, just launching the .exe file will trigger default values, the other way is launching from the console by giving the required arguments. More detail on this can be found in the ServerApp.cs or Client.cs files. Once launched, the server creates a listening socket in the StartListening() method of SynchronousSocketListener.cs. After a client is launched it will establish connection with the server, which will launch the game after 3 client connections have been established. The number of necessary clients is arbitrary as it was not specified, and can easily be modified. Socket-wise, the listener keeps listening and for each new client it creates a new socket. At 3 sockets, new threads are established and started. These threads are instances of Game, managed by each client.

Acceleration of time during gameplay has also been implemented. Default values are 1500ms at start and -10ms after each update.

Object design



The Tetris is playing in the socket listener, each player create an instance of a game. All the functionalities of the game, all the mechanics and the environment of the Tetris is in the class game. This is this class which will allow all the operation during the game. This class include a block class which correspond to the block falling in the matrices include in the class game. Each block has a position and a particular shape which is take in consideration in the game class.