

# *Tetris*

## Software Design Patterns

Guided by **Dr. Christophe Rodrigues**  
Created by **Gregor Jouet** and **Manasa Prakash**

*“You can either meditate for 20 minutes everyday or play Tetris. If Tetris has taught me anything, it's that errors pile up & accomplishments disappear.”*

## Index

Introduction	Page 2
Client Server Model	Page 2
Object Oriented Approach	Page 3
Game Logic	Page 5
References	Page 5

# 1. Introduction

Tetris is a tile-matching puzzle video game.

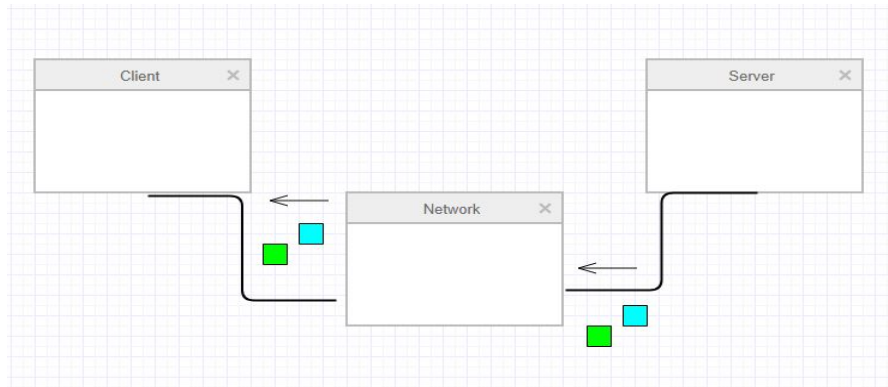


In Tetris, the primary way to score points is to clear lines by manipulating the pieces so that they fill horizontal row within the Matrix. As the pieces fall, the player's goal is to move and spin them so that they line up evenly at the bottom of the Matrix.

The objective of this project is to build Tetris as a network game by implementing a client-server model which will be used with Sockets. On the server side, multi-client management will be provided by Threads.

## 2. Client-Server Model

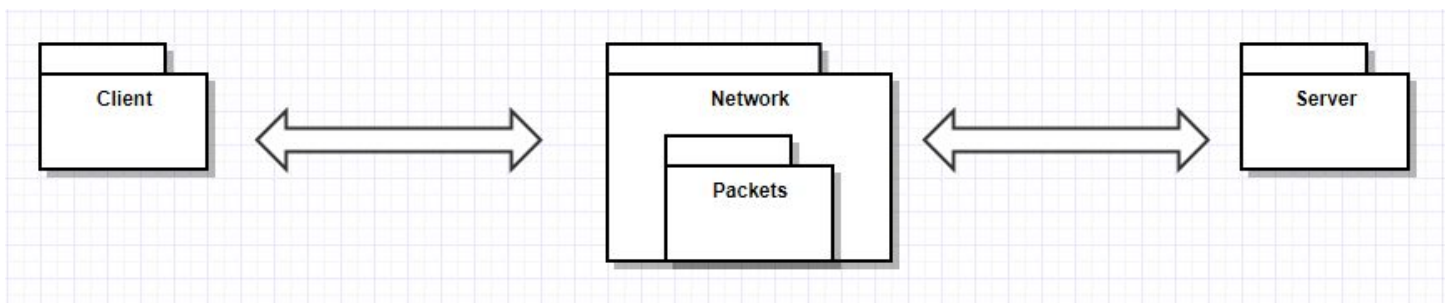
The client-server model follows a TCP communication protocol with principles of packet switching. The model hence consists of 3 parts namely the client, the server and the network that handles the communication by reading and dispatching packets, and by handling network callbacks. In packet switching messages are divided into smaller piece called PACKET. Each packet includes source and destination address information so that individual packets can be routed through the network independently.



### 3. Object Oriented Approach

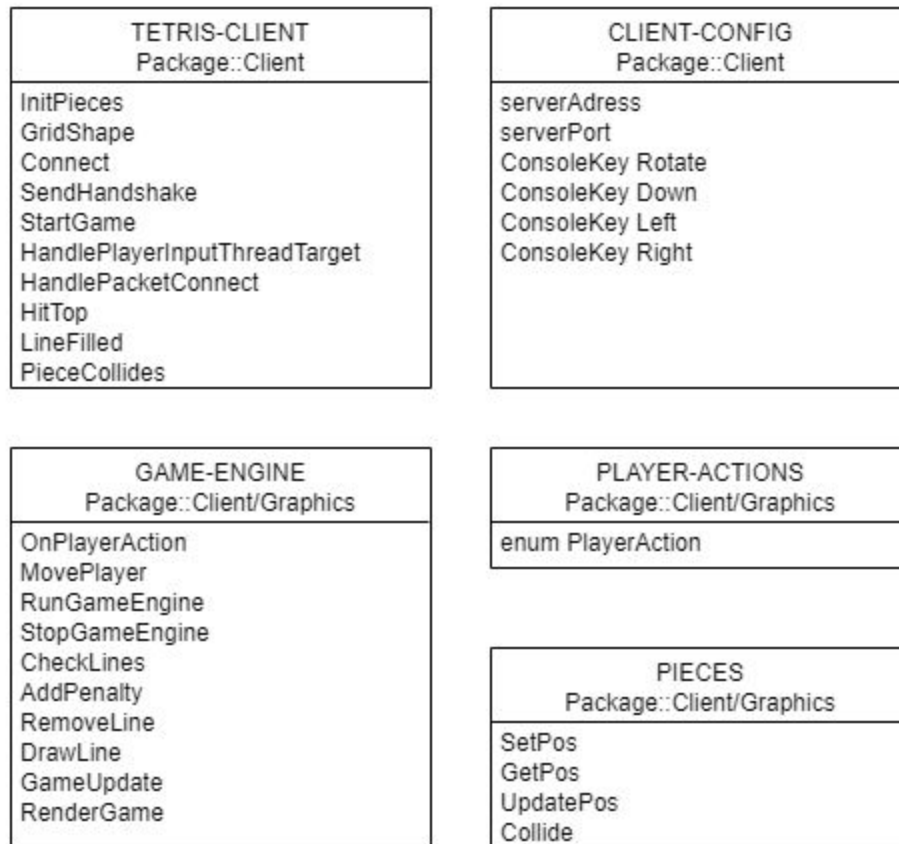
The program consists of 3 packages which are the client, the server and the network. The network package consists of another subpackage which is named Packets and manages the reading and writing of packages by considering primitive data types as binary values in a specific encoding. The structure of the classes in these packages and the relationships between them is detailed in the diagram below.

#### Package Diagram

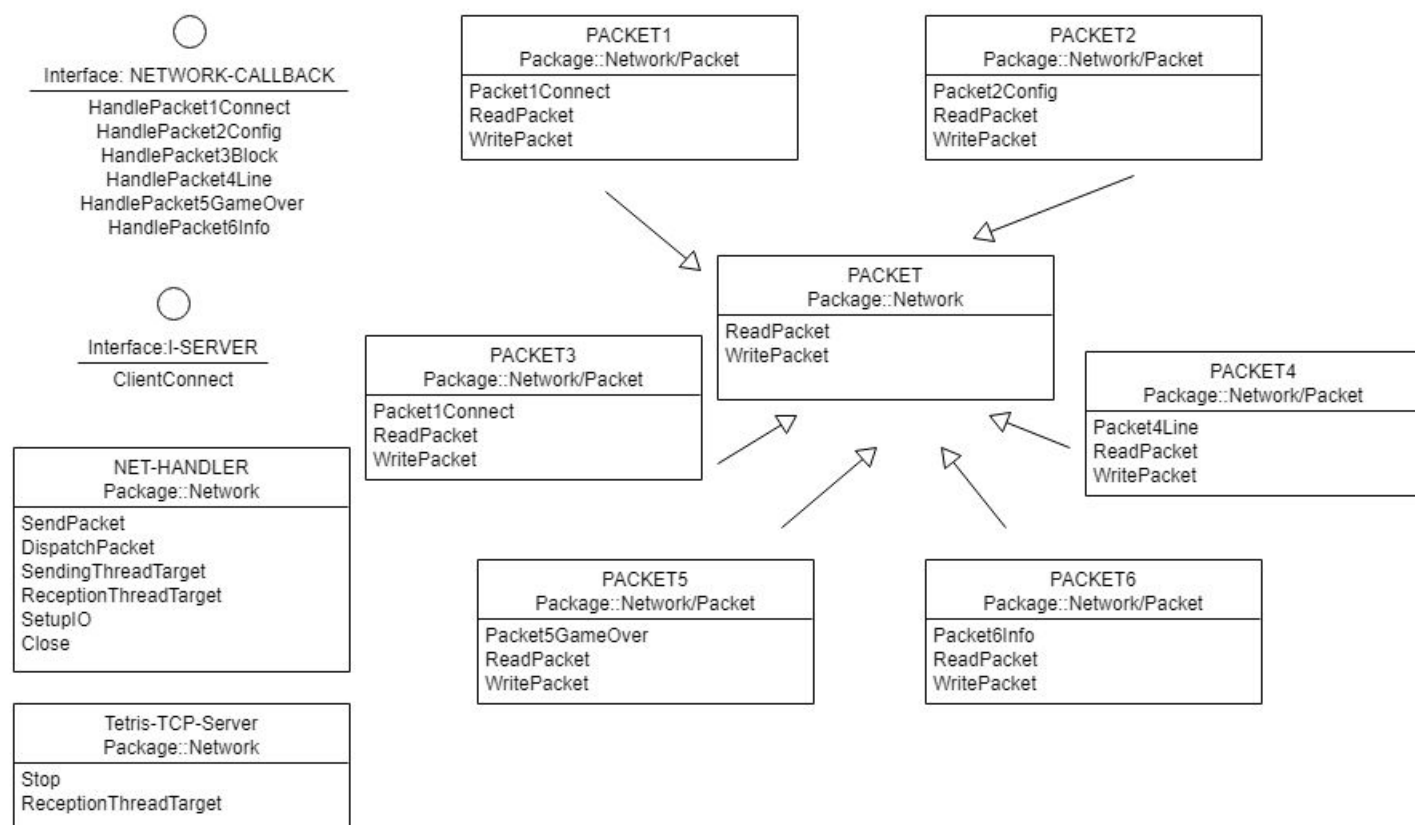


#### Class Diagrams

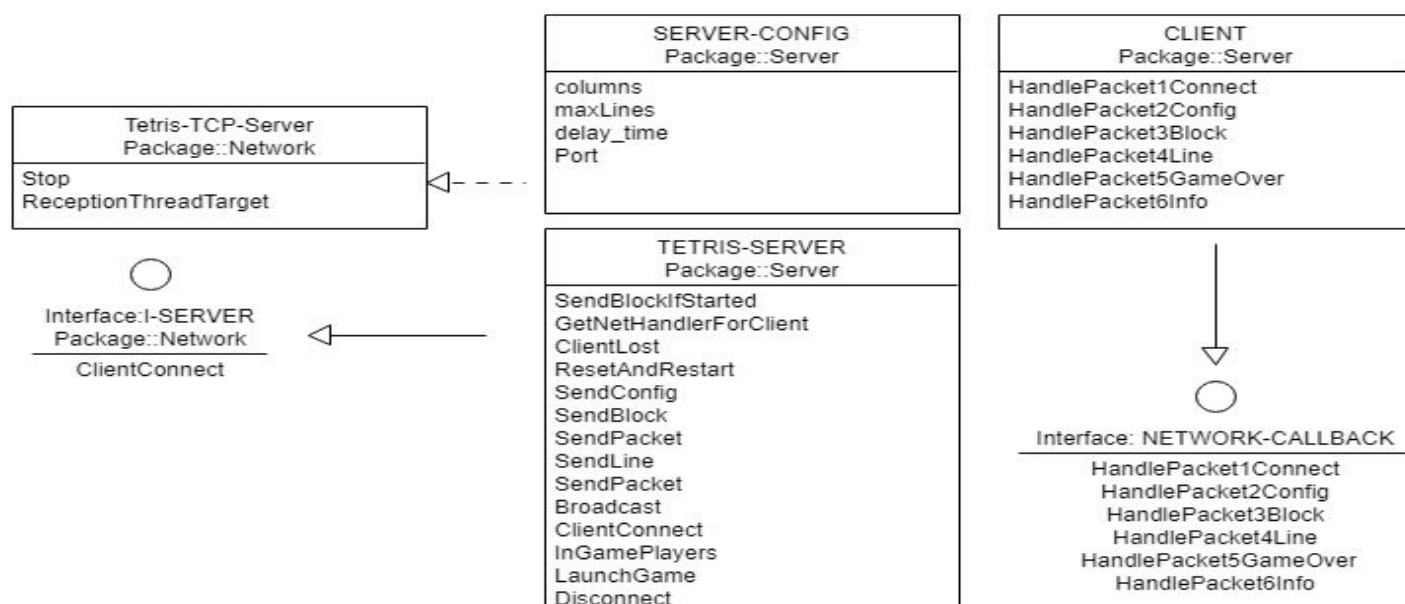
##### Client



## Network



## Server



## 4. Game Logic

### Winning

When a player (client) creates N horizontal lines without gaps by placing a block; for the player, the N lines get destroyed, and any block above the deleted line will fall and for every other player, N penalty filled horizontal lines at the bottom of their screens are added. These penalty lines cannot be deleted; the blocks of such penalty lines will be displayed using star chars ('\*').

When a player has a block positioned on its top line, he/she has lost. The socket communication is closed, the message "Game over" is displayed, and the client program stops.

## Gravity

Traditional versions of *Tetris* move the stacks of blocks down by a distance exactly equal to the height of the cleared rows below them. Here, we manage the gravity using `delay_speed` in order to ensure uniformity among all clients.

## 5. References

- [Project specification document](#)
- [Sockets | Microsoft Docs](#)
- [Packet Switching | Wikipedia](#)