

Bootcamp Project - Tetris

Introduction

For this exercise your goal will be to recreate the old Tetris game. In doing so you will learn many things about how a game is done, and the basics of such an exercise will serve you well as a foundation for working on more complex games in the future.

For this exercise you are going to use SFML or SDL for doing any graphics, input and sound.

Gameplay

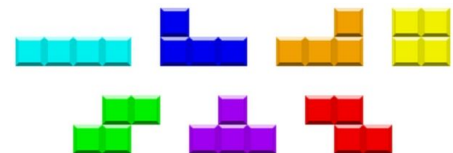
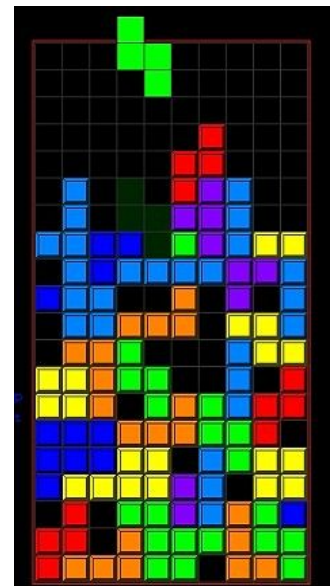
Tetris normally needs no introduction, but for the sake of completeness let's go over the rules.

Tetris consists of a grid board where pieces fall from the top of the screen and need to be aligned in order to complete rows and make them disappear.

The game board consists of a grid of 10 by 20 blocks, that is 10 blocks wide and 20 blocks high. Even though there's no direct implications on the gameplay regarding board size, that is the standard Tetris board, and we will use that size for our clone.

In Tetris the player needs to move and place down different game pieces, called Tetriminos in order to complete rows. The user can control the free fall of the pieces by moving it to the left or to the right, or also by rotating the piece by 90 degrees at a time. Additionally the user can accelerate the free fall by moving the piece downwards.

Each game piece, or tetrimino, is a group of blocks, forming an oddly shaped object and color coded so that they are easier to identify. There are 7 different tetriminos in Tetris.



Game Rules

Tetriminos will appear from the top of the screen, and start falling down towards the bottom of the screen. The speed at which they fall down usually depends on the level of the game the player is in, but they will always move one square at a time.

The player will be able to control the fall of the tetrimino by moving it, always by 1 square at a time, to the left or to the right. Additionally the player could move the piece down, to accelerate the fall.

If the downwards move cannot be completed because any of the grid positions where the tetrimino needs to move to are occupied by another block, then the piece is considered to be “colliding” and is fixed in place. Keep in mind that moving to the left or right by user interaction does not count to the colliding rule, if the player presses right, and the move cannot be completed because one of the spaces is taken, then the tetrimino simply will not move.

After the tetrimino is fixed in place the game board is checked to see if any rows are now **full**, and if they are, those rows are removed. Everything that was on top of the removed lines falls down filling in the empty space.

The player will be awarded points based on the number of removed lines (10 points per line).

Last but not least, another piece appears on the top of the game board, and the process repeats itself.

Every 10 lines the player will advance one level, which increases the fall speed of the tetriminos, making the game harder each time.

Project deliverables and rules

All your project code should be hosted in your GitHub repository. Additionally you must include a very brief design document, specifying the modules and classes you are implementing. The main code branch must always reflect what is specified in the design document. Also, it is important to add in code documentation.

Every code submission should have a detailed description of the changes that were introduced, and must compile and run without errors.

You have 30 days to complete the whole project. The code delivered after the deadline will be rejected.

Productivity Objectives

- Understand the core C++ language syntax
- Identify core object-oriented concepts
- Describe and implement the 3 types of polymorphism (Parametric polymorphism, Ad hoc polymorphism & Subtyping)
- Design and use classes effectively
- Design and use methods and reference arguments effectively
- Make use of operator overloading
- Design and use class hierarchies and virtual functions
- Write and use incomplete (abstract) classes
- Understand and use the ANSI C++ library
- Use “const” intelligently
- Make basic use of the STL
- Work with C++ I/O
- Program with the C++ try/catch exception model
- Use casting intelligently

Success criteria

Your Tetris version should compile and run without issues and abnormal terminations. The program should not present memory leaks and it is expected to make a proper use of the C++ language features and meeting the Productivity Objectives.