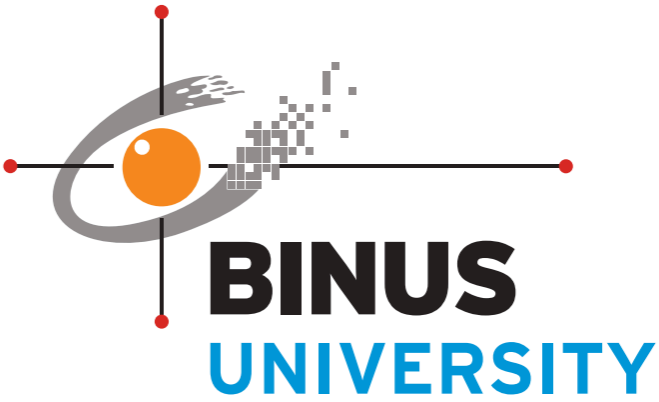
Algorithm and Programming Final Project Report



Name: Agie Winata (2902642726)

Project Title: Tetris Game

Class: L1BC

Lecturer Name: Jude Joseph Lamug Martinez MCS

Algorithm and Programming

Binus International University

Jakarta

Table of Contents

Introduction

The final project that I have submitted for this course is a classic Tetris game. Tetris is a puzzle game, first released on January 27, 1988, by Alexey Pajitnov. The game features blocks constructed in 7 distinct shapes, all of which are referred to as a tetrimonis. Throughout each round, random shapes of tetrominos would be falling from the top-middle of the board; the player's objective would be to prevent such tetrominos from reaching the top of the board. The player can do so through clearing the board by arranging the tetrominos in such a way that one horizontal line, spanning the X-axis of the board, is formed. Once this happened, the game would then clear part of the board where the horizontal lines are once formed, and all blocks on top of that line would be moved down until they reach another block or hit the bottom of the board. This logic continues until the board is filled vertically by the tetrominos, whereby a game-over screen would be displayed. The number of horizontal lines cleared is directly proportional to the points that the player obtained. Furthermore, the number of points that the players earn is also directly proportional to the speed at which the tetrominos are falling on the board.

Project Specification

The project itself is focused on modularity, as it allows for things it allows different aspects of the game to deliver smoothly to the users, whilst also maintaining readability and effectiveness for developers. This project contains seven classes, each is responsible for different aspects of the game, through the utilisation of methods and classes. The following part of the report will give a brief overview of how these classes work together to deliver a seamless experience for the user through the use of object-oriented programming. All details regarding the inner workings of the code can be seen from the documents that are inside the repository in the appendix section.

Game Components

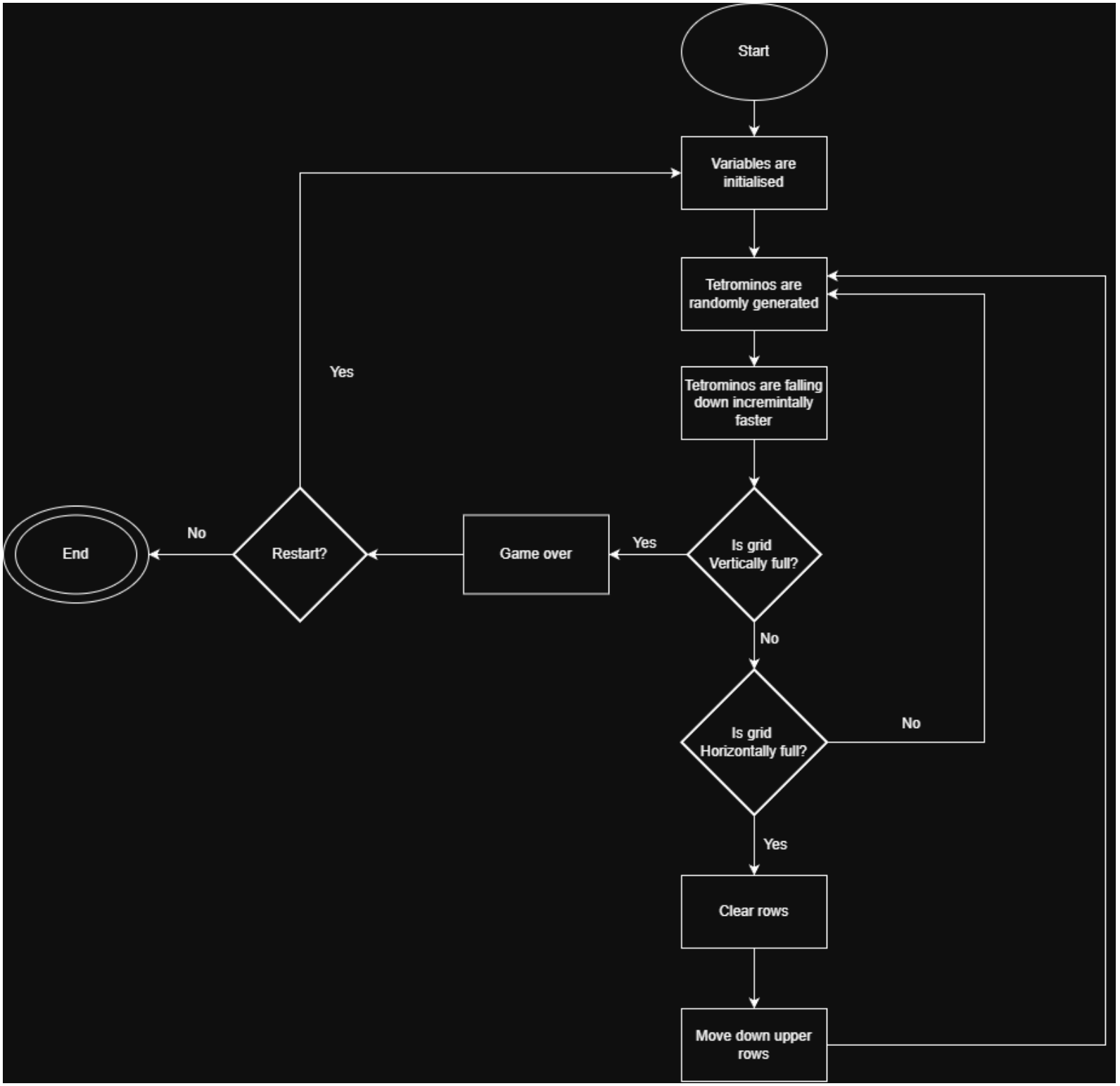
The main game loop of the project is responsible for processing the user's input, updating the game canvas with objects created from different classes, and increasing the difficulty for users. The grid itself is composed of a 20x10 two-dimensional list of zeros. The grid class itself is responsible for many of the game aspects, such as checking the position of the tetrominos, cleaning completed rows, and handling the falling blocks as a result of that clearing effectively. Furthermore, the grid is also responsible for updating itself by drawing already locked tetrominos into itself, so that players know which spaces are available for new tetrominos.

The game class is responsible for overseeing all the logic that this game has. It is responsible for initialising the current tetrominos and the one after that at random. Not only that, but this class is also responsible for the collision handling feature of the game, where it checks if a block placement or rotation is valid and reverses it if it is not. Furthermore, this class is also responsible for handling score updates, where the number of lines cleared at an instance is directly proportional to the score earned by the player. Last but not least, this class is also responsible for resetting all important variables of the game once a "game-over" event is triggered.

Tetrominos

The tetrominos themselves have some functions built into them. First and foremost, these tetrominos are initialised by a dictionary containing several lists, where each list dictates the coordinates of the individual blocks that make up the tetrominos. The keys to this list would serve as a means to rotate the tetrimonos. Not only that, but these tetrominos also have an ID variable that they have inherited from their parent class, where each ID corresponds to a different colour and shape of the tetrominos. These tetrominos would also move left, right, and down according to the user input, whilst also naturally moving down incrementally faster due to the game's main logic.

Use-Case Diagram



Depedencies

This project requires several dependencies to run, one of which is PyGame, which is not outdated, as it is responsible for creating the canvas and generating all the blocks used in this game. Another dependency that needs to be installed is an IDE that can read and run Python, as this entire project is written in Python. Lastly, Python needs to be installed so that the machine knows how to interpret and run the logic of the code written.

References

Progamming With Nick. (2023, May 4). *Creating Tetris in Python with pygame - Beginner Tutorial (OOP)*. Www.youtube.com. https://www.youtube.com/watch?v=nF_crEtmpBo

Wikipedia Contributors. (2019, October 12). *Tetris*. Wikipedia; Wikimedia Foundation. https://en.wikipedia.org/wiki/Tetris

Appendix

- Github Link: Here
- Poster: Here
- Tetris gameplay: Here