

Project Idea

Tetris Game with AI

Description

This project is essentially a Tetris game with an AI component that is able to “learn” how to improve at the game by playing games. In Tetris, randomly selected tetriminos will appear at the top of the screen and fall downward. The user can rotate the piece or move it from side to side to position it in the desired position. When the piece is unable to move any further down, it is locked and the user can no longer change its position or orientation. When a full row is completed, that row is removed and all the blocks above that row are shifted downward. As the game progresses, the speed at which the tetriminos drop will increase. The AI component of the game will use a genetic algorithm which will continue to optimize the values of the “genes” of the AI with each game it plays. Each “gene” would be a value that determines the importance of a measurement of the state of the game such as the number of lines cleared, the number of holes, and the smoothness of the top layer. Each organism would have a set of “genes”, each with a value. After each test round, the top scoring 50% of organisms will be mixed with some random mutation to produce a new batch of organisms. As more test rounds are completed, the AI should become better and better at the game.

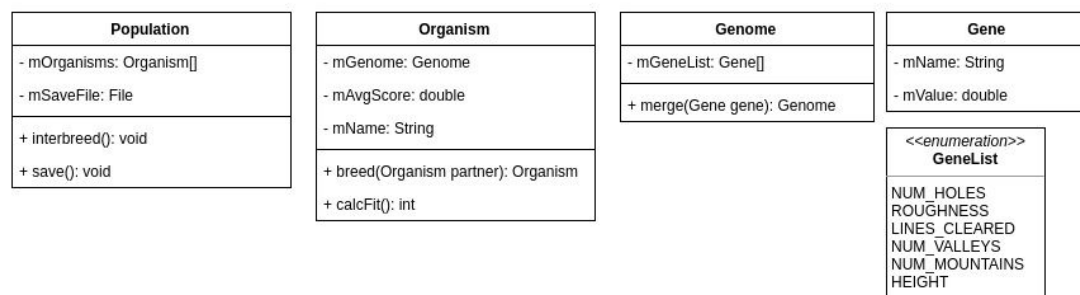
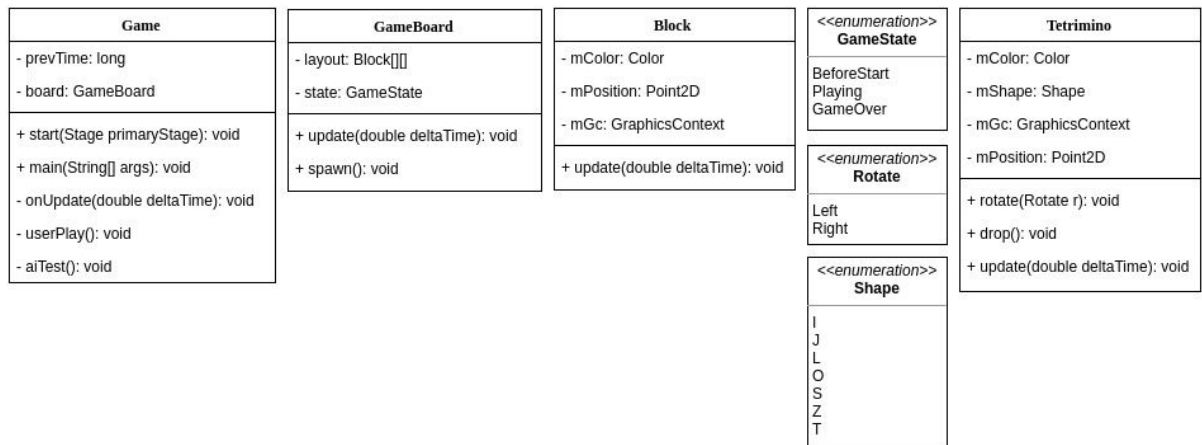
Concepts Used

Concept	Use
Algorithms to specify a problem solution and flow charts to aid in formulating them	<ul style="list-style-type: none">- Planning of the program (proposal)- Class definitions and diagram
Data representation (variable types)	<ul style="list-style-type: none">- Various data types will be used to store data in the program (double, int, String, etc.)
Branching and Conditional Expressions	<ul style="list-style-type: none">- Many branches and conditional expressions will be used (checking if a row is complete, checking if game is over, etc.)
Loops	<ul style="list-style-type: none">- Loops will be used to iterate through all the rows and check for complete rows
Input/Output	<ul style="list-style-type: none">- The values of the “genes” of the AI will be outputted to a file after every session and read from a file at the beginning of each session
Random Numbers	<ul style="list-style-type: none">- To generate the initial value of the “gene”

Culminating Project - Proposal
Michael Pu - 2017/12/21 - ICS3U

	<ul style="list-style-type: none"> - To decide on the next tetromino to appear
Methods	<ul style="list-style-type: none"> - Many methods will be used (rotating a tetromino, dropping a tetromino, running the AI session, etc.)
Classes (Object Oriented Programming)	<ul style="list-style-type: none"> - The program will follow the concepts of OOP (data encapsulation) - Each tetromino, the game, each “organism” in the AI, and many other components of the game will each be its own class
Arrays (1D and 2D)	<ul style="list-style-type: none"> - The state of the game will most likely be stored in a 2D array - The “organisms” in the AI will most likely be stored in a 1D array
Basic GUI using JavaFX	<ul style="list-style-type: none"> - The GUI interface of the game will be created using JavaFX

Planning Diagram



GUI Layout

