**Tetris bot**

AI is being used more and more frequently in various sectors. Working with it can be challenging and difficult, so the team decided to create a project which will demonstrate the use of AI in a game of Tetris. The idea behind it is to make sure that the AI grows and evolves as the game progresses (klinkt gelijk dat het in 1 spel al direct evolvet, dus mss beter “… evolves the more games it plays”). Keeping this goal in mind, the team decided to use a genetic algorithm as the base for the AI.

To begin with the analysis, the creation of a Use Case Diagram seemed to be a good starting point. Here, the general demonstration of the program is shown without diving deep in the backend.**Diagram

Description automatically generated**

The following diagram demonstrates the General Sequence of the game. This diagram shows when various components of the game are being activated and under which circumstances the game can be terminated.

A picture containing calendar

Description automatically generated

Diagram

Description automatically generatedIn the General Activity Diagram, the general working pattern of the game is shown in a comprehensive way. In the AI column, the condition under which the AI evolves and grows are shown. The game is played until the AI loses or until a gene of the AI has finished its 500 moves. This results in the next gene starting to play the game. After all 50 genes in a generation have played, the AI evolves and moves on to the next, better generation.

The Class Diagram shows the various classes and methods that might be needed to code the game. This doesn’t include the UI (User Interface) or UX (User Experience) changes (changes? This doesn’t include the changes in UI or UX? Of wat doet die changes daar). The team decided to create three major classes (AI, User and Tetris). AI class contains the methods and parameters needed for the AI. Tetris contains the general methods and parameters for the game including the functions of AI. User Class contains the methods and parameters so that the user can play the game (instead of the AI).

Diagram

Description automatically generated

A more in-depth view of the AI’s Genetic Algorithm:

50 genes are generated per generation. Each gene gets the chance to play the game. The moves made by each gene get rated. Once the genes are done with their moves, they get sorted in ascending order based on their respective rating. The ones with the best ratings are selected to create children for the next generation. There is also the possibility of a child mutating their genes. Each child is then given the chance to play the move and those are rated as well. (moet move hier nie game zijn? Of duidelijk maken da elk kind 1 spel speelt waarin elke move gerate wordt) This way the AI learns and evolves.

Diagram

Description automatically generated