Imprimir

PSI Group C Wiki

Group members

Alexander Pazos Lago

Clara Benéitez Trillo

Stefan Bajić

Laura González Lemos

Project explanation

Hex, also called Nash, is a two player abstract strategy board game in which players attempt to connect opposite sides of a rhombus-shaped board made of hexagonal cells.

A turn in Hex consists of placing a piece of your color on a hexagon. The first player's goal is to form an unbroken chain of hexes of his color that connects the top to the bottom while the second player tries to form an unbroken chain connecting the left and right sides.

MiniMax Algorithm

Minimax is a kind of backtracking algorithm that is used in decision making and game theory to find the optimal move for a player, assuming that your opponent also plays optimally. It is widely used in two player turn-based games such as Tic-Tac-Toe, Backgammon, Mancala, Chess, etc.

A simple java program to find maximum score that maximizing player can get.

Minimax Algorithm with Alpha-Beta pruning

Alpha-Beta pruning is not actually a new algorithm, but rather an optimization technique for the minimax algorithm. It reduces the computation time by a huge factor. This allows us to search much faster and even go into deeper levels in the game tree. It cuts off branches in the game tree which need not be searched because there already exists a better move available.

Java program to demonstrate working of Alpha-Beta Pruning

Evaluation Function

A node tree is constructed at this point. The game state is evaluated when a terminal node is reached, either because a player has won or because the maximum depth has been reached.

It takes into account:

1. To optimize AI speed, the depth progressively increases during the game.

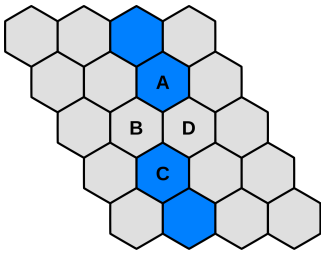
It's a better state for the Maximizer Player if:

2. He wins, and the state is better with fewer moves needed for victory.

3. Central cells are prioritized over border cells.

4. He has a set of cells (we refer to it as an island) oriented from top to bottom.

5. He has two or more cells that are guaranteed to be connected in the future (this is called “bridge”).



7. Opponent tries to prevent the Player from connecting cells and he stops the opponent from doing it.

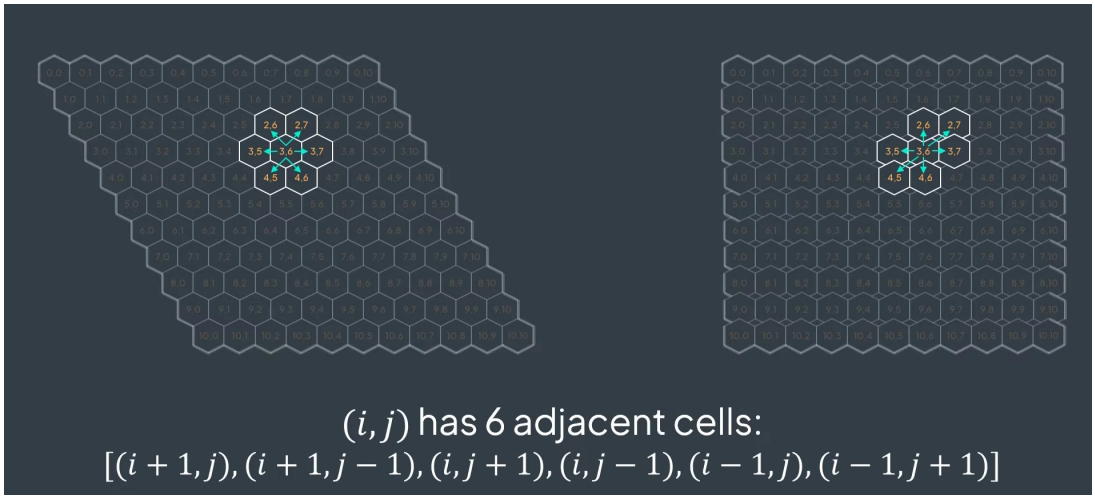
It’s a worse state for the Maximizer Player if:

8. Opponent successfully prevents cell connection.

9. Opponent successfully prevents the Player from having a "cell bridge."

Game Core

The hexagonal board can be a square matrix like this:



Problems

- During the implementation of the AI, it was challenging to figure out how to retain a copy of the current game state needed for each level of the Minimax algorithm, primarily because Hashmaps were used to detect the game winner.
- We had to limit the depth level of the AI algorithm because smartphones have more resource constraints.

State of the art

Collection of similar works or apps with appropriate references.
Differences between the existing apps and yours.

- Hex: A Connection Game https://play.google.com/store/apps/details?id=com.game.hex

The main difference between this application and ours is that the board is 11x11. Probably has more levels of depth.

- Hex.html: https://www.lutanho.net/play/hex.html

The main difference between this project and ours is that this is not a mobile application. Moreover, it implements a neural network.

Project planning

It must indicate tasks, participants, the range of weeks for each task and the participants assigned to each task.
To give a graphical overview of the tasks and their temporal sequencing is appropriate to use Gantt charts with tools like http://www.ganttproject.biz/

Resources used

Resources needed for their development and use, like tutorials, external libraries, etc.

- Hex GUI Template: https://sourceforge.net/projects/benzene/files/hexgui/hexgui-0.9.0.zip/download
- Benzene: https://benzene.sourceforge.net/
- Disjoint-set structure to detect a winner in Hex game:

- Tic Tac Toe AI with Minimax Algorithm:

- Android Chess 007: Drawing 8x8 chessboard in Kotlin:

- Minimax and Alpha-Beta Pruning Algorithms and Psuedocodes:

Some online resources and courses:

- Android Developers http://developer.android.com/index.html
- Android Developer NanoDegree https://www.udacity.com/course/android-developer-nanodegree--nd801
- Programming Mobile Applications for Android Handheld Systems: Part 1 https://www.coursera.org/learn/android-programming
- Programming Mobile Applications for Android Handheld Systems: Part 2 https://www.coursera.org/learn/android-programming-2
- Android programming course: learn how to create your own applications http://www.sgoliver.net/blog/curso-de-programacion-android/

Class concepts

Application of concepts covered in class in the development of the project.

In the Game Theory topic we studied the Max-min strategy which helped us to understand the Minimax algorithm.

Mocks

https://ninjamock.com/s/DW66WLx

Individual work

From each of the members of the group, a list with: date, time taken and work done.
The information in this page will not be used to grade the students, just to estimate the amount of work done for this course.

Alex: Core Implementation + GUI Implementation, 35 hours approx.

Clara: Core Implementation + Intelligence Implementation, 35 hours approx.

Laura: Intelligence Implementation, 20 hours approx.

Stefan: GUI Implementation, 12 hours approx.

References

List of references used to develop the project.

- Hex Game: https://en.wikipedia.org/wiki/Hex_(board_game)
- Hex Bot: https://devpost.com/software/hex-bot
- Hexy: http://vanshel.com/Hexy/