Phase 2: Pentominoes

By: David, Ilya, Irdi, Magdy, Max, Pablo, Zenios

TABLE OF CONTENTS

Introduction

Creating UI

The Game

Bot Implementation

The Best Sequence

Conclusion

01

Introduction

“If Tetris has taught me anything it's that errors pile up and accomplishments disappear" - Unknown

Overview of Phase 2

1) Design a UI for a Tetris-style game using the Pentomino pieces from Phase 1
   - With the ability to move/rotate pieces and a high score tracker
2) Implement a bot capable of efficiently playing our game
3) Find an optimal order of pentominoes that will lead the bot to a high score

02

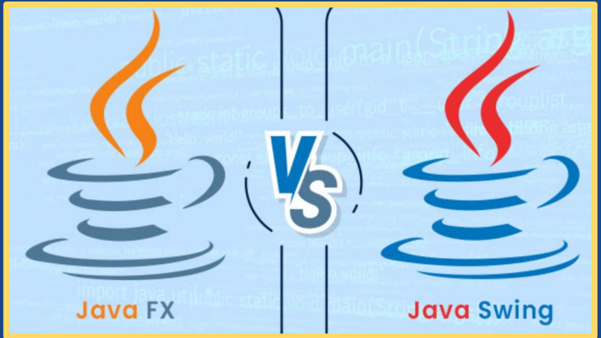
Creating UI

"The shorter way to do many things is to only do one thing at a time." - Mozart

Why Java Swing?

- We preferred Java Swing as it is more lightweight and doesn't need any extra configurations.

- Swing components provided us higher level inbuilt functionalities, allowing us to write a more organized program.



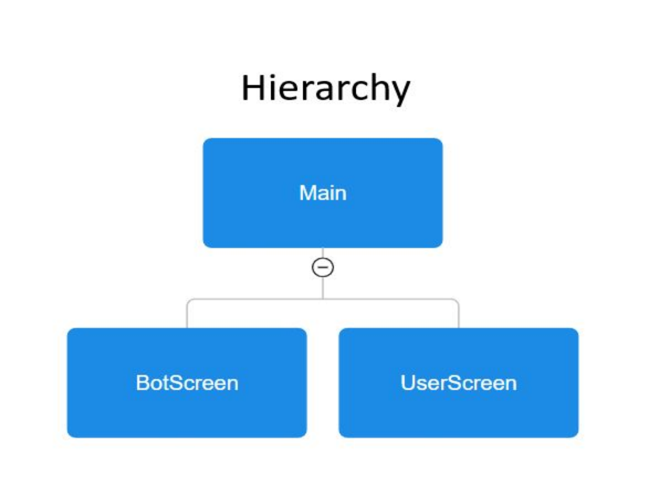
GUI implementation

- We prioritized functionality and logic rather than visual aesthetic

- Key components: a game board display, showing the score and high-score, a preview of the next pentomino

- We used a multi-frame layout

- Included double-buffering



03

The Game

"Hello, IT. Have you tried turning it off and on again?" - IT Crowd

Characteristics

- The requirements asked for a functional 5x12 tetris game

- The player must be able to rotate the pentominoes and move them

- Lines are cleared when they are full, and the pieces of this cleared line fall down by 1 block

- Points are accumulated per lines cleared

“Falling” Method

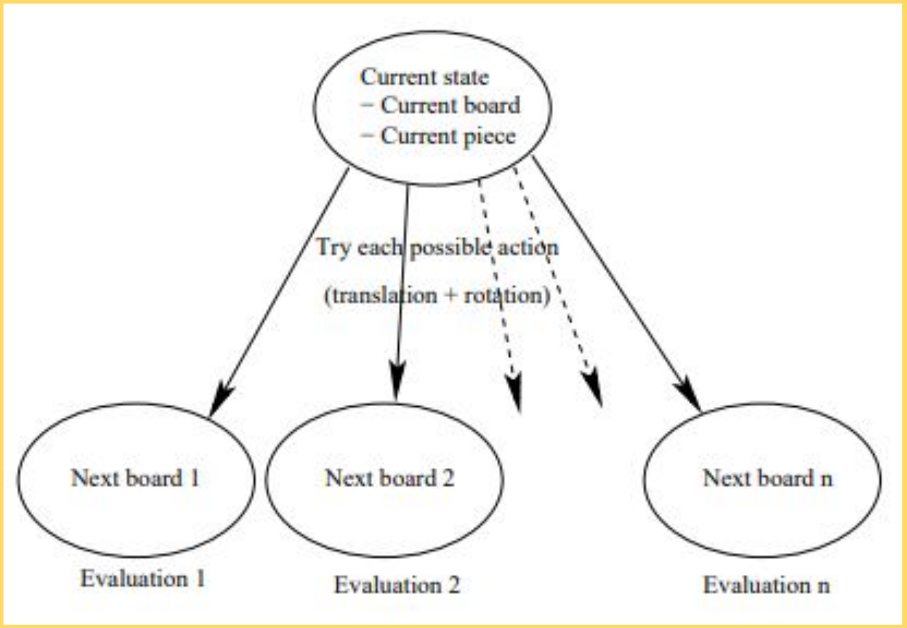
- Frame rates are used to determine the speed of the block fall
- The block drops one line per game tick (frame)
- Finding the right speed to make the game playable was tricky (11 fps)
- This sequence repeats itself so the game can keep playing

04

Bot Implementation

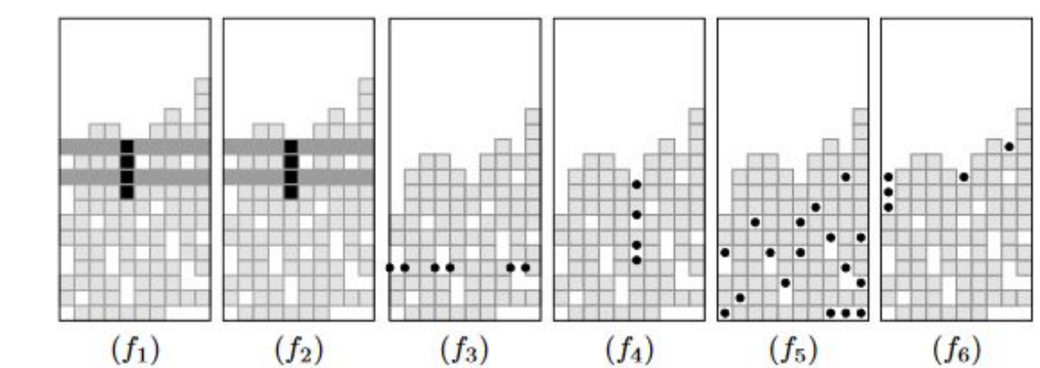
Bot Decision-Making Process

- The bot checks each possible placement of the block on the field and evaluates the state
- Using a custom evaluation formula using different weights, it calculates the evaluation
- Then it places the block with the highest evaluation



Evaluation Process

- We created 6 function features:
  - Eroded Pieces
  - Landing Height
  - Row Transition
  - Col Transition
  - Number of holes
  - Cumulative Holes

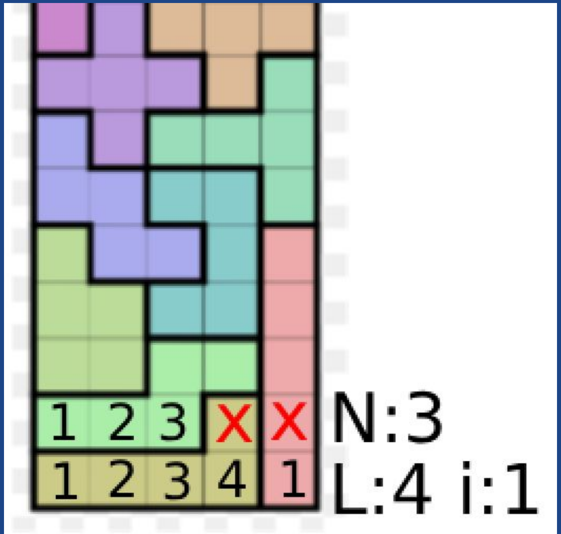


05

The Best Sequence

Heuristic Approach

- The algorithm uses a solved pentomino puzzle (using phase 1's project)
- Starts at the bottom row and goes up
- Compares the frequency that a specific block occupies in that row
- Skips the ones that are already in the sequence
- And puts the ones with the highest frequency first in the sequence
- If they have the same frequency, it checks its height and chooses the lowest one to put next in the sequence



Drawback: does not work for all puzzles (see diagram)

Sequence found: [L, V, U, 1, N, F, X, Z, W, P, T, Y]

Brute Force Approach

- First we shuffle the 12 distinct pieces
- The bot will play with those 12 pieces and records the score
- If the bot can take more pieces we feed it the same 12 pieces again
- This repeats until it loses, then it records the score
- If the bot is in a loop and never stops, this means we have found a sequence that in theory allows for an infinite score

Drawback: the bot may encounter the same 12 pieces (very unlikely)

Sequence found using the bot: [L, W, V, T, P, U, X, F, Z, N, 1, Y]

06

Conclusion

Conclusion

We got:

- Working game!
- Working bot! (and sequence)

We could have:

- Read up more on different options before starting to code
- Improve the UI
- More systematically developing the weights of the algorithm

What We Learned

- Working with game-loops (implementing Java Runnable Threads)
- Designing a UI with different sections using specific layouts in Swing
- Developing evaluation functions and implement a bot that can determine the best moves
- Splitting the code work-load on a complex project
- Time management and effective team-work

07
Bibliography

Bibliography

admin. (2022, May 3). JavaFX Vs Java Swing: Choose The Best For Web App Development. XcelTec. https://www.xceltec.com/java-development/javafx-vs-java-swing-choose-the-best-for-web-application-development/

Phon-Amnuaisuk, S. (2015). Evolving and Discovering Tetris Gameplay Strategies. Procedia Computer Science 60(1):458-467. https://doi.org/10.1016/j.procs.2015.08.167.

Cormen, T., Leiserson, C., Rivest, R., & Stein, C. (2009). Introduction to Algorithms (Third). Mit Press.

Russell, S. J., & Norvig, P. (2016). Artificial intelligence : a modern approach (3rd ed.). Pearson.

Christophe Thiery, Bruno Scherrer. Building Controllers for Tetris. International Computer Games Association Journal, 2009, 32, pp.3-11. inria-00418954