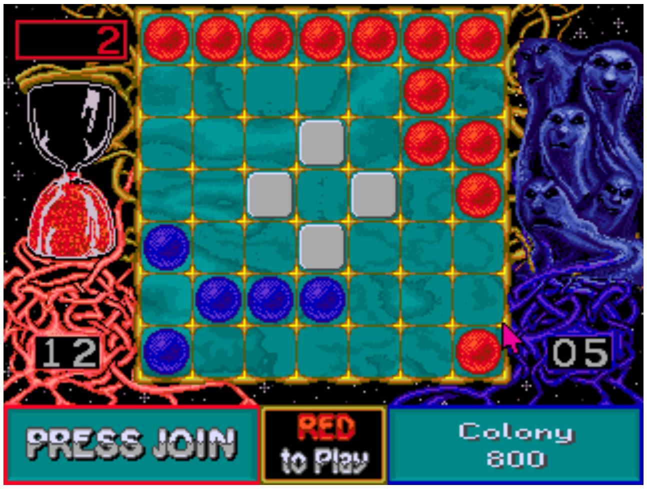
Homework 5: Animation and Sound in Ataxx

- Due Nov 26 by 8am
- Points 100
- Submitting a file upload
- File Types zip
- Available Aug 20 at 8am - Dec 2 at 8am

This assignment was locked Dec 2 at 8am.

For this assignment, you will implement **Ataxx (https://en.wikipedia.org/wiki/Ataxx)** , a strategy game where two players compete to dominate the board. The rules allow players to clone or jump their pieces across the board, aiming to convert the opponent's pieces to their own. The goal is to have the most pieces on the board by the end.



(https://en.wikipedia.org/wiki/File:Ataxx_Arcade_Gameplay_Screenshot.png)

Short History Video:

Infection, Explained in 64 Seconds (https://www.youtube.com/watch?v=4_NzTapUqqM)

(https://www.youtube.com/watch?v=4_NzTapUqqM)

Video demonstrations:

Infection (https://www.youtube.com/watch?v=DjYhX5pc4hg)

(https://www.youtube.com/watch?v=DjYhX5pc4hg)
(Commodore 64)

Attax (https://www.youtube.com/watch?v=5aQmj-9u2t4)

(https://www.youtube.com/watch?v=5aQmj-9u2t4)
(Arcade)

7-Up Spot The Video Game (https://www.youtube.com/watch?v=a8Ezmf6z3kM)

(https://www.youtube.com/watch?v=a8Ezmf6z3kM)
(NES)

Sierra Imagination Network Boogers (https://www.youtube.com/watch?v=fZhea4vhqEg)

(https://www.youtube.com/watch?v=fZhea4vhqEg)
(MS-DOS PC)

The 7th Guest –Microscope Puzzle / Infection (https://www.youtube.com/watch?v=mXQF7hF9Bwg)

(https://www.youtube.com/watch?v=mXQF7hF9Bwg)
(MS-DOS PC)

Online Implementations:

Ataxx (https://www.onlinesologames.com/ataxx) (web clone)

Ataxx (http://viiksipojat.fi/ataxx/) (another web clone)

Ataxxmax (https://www.lexaloffle.com/bbs/?pid=85540) (PICO-8 Christmas Version)

Hexxagon (https://hexxagon.com/) (hexagon board version)

How to Play:

Ataxx is a turn-based strategy board game played on a grid game board where players either jump a piece two squares or grow a new piece in any adjacent cell (8-way direction). When a piece lands next to an opponent's, all adjacent opponent pieces are converted to the player's color. The game ends when no more moves are possible, and the player with the most pieces wins.

Assignment Requirements

1. **Game Rules**

- **Map size:** 7x7 board.
- **Two teams:** Team A and Team B.
- **Each turn**, a piece may:
- **Clone:** Move to an adjacent cell (one square away) and clone the piece.
- **Jump:** Move two squares away, leaving the original position empty.
- If a piece lands next to an opponent's piece, it **converts** all adjacent opponent pieces into the player's color.
- A player's goal is to maximize their playing piece count by the end of the game such that they have more than their opponent.
- If there is a time limit enabled, the game ends if a player's time runs out and that player loses.

2. **Custom Map Features**

- **Maps** will be loaded from **levels.txt (https://gatech.instructure.com/courses/409430/files/56511895?wrap=1)** **(https://gatech.instructure.com/courses/409430/files/56511895/download?download_frd=1)** and support custom configurations.
- Maps are serialized in **JSON format** and must include:
- **Untraversable cells**: Marked as 9.
- **Traversable cells**: Marked with 0.
- **Starting positions for Team A and B pieces**: Marked as 1 for Team A and 2 for Team B.
- Example map (7x7):

```
{
    "name": "Level 2",
    "size": [7, 7],
    "board": [
        [1, 0, 0, 0, 0, 0, 2],
        [0, 0, 0, 0, 0, 0, 0],
        [0, 0, 9, 0, 9, 0, 0],
        [0, 0, 0, 0, 0, 0, 0],
        [0, 0, 9, 0, 9, 0, 0],
        [0, 0, 0, 0, 0, 0, 0],
        [2, 0, 0, 0, 0, 0, 1]
    ]
},
```

3. **UI Elements and Gameplay Requirements**

- **Start Screen / Configuration**:
  - Game Board List: The game starts with a list of maps loaded from levels.txt which the user selects.
  - Select play mode with choice of **Player vs Player** or **Player vs Computer** (if extra credit implemented)
  - Select time mode of **Unlimited** or some number of minutes per player
- **Game Board**:
- The 7x7 board renders cells with the following states:
  - **Team A piece**: Color-coded.
  - **Team B piece**: Same size and shape, but different color from Team A.
  - **Untraversable cells**: Marked with a unique style indicating an obstacle to movement.
  - **Traversable (empty) cells**: Available for moves.
- **Interaction: Click** on a piece to select it (visually indicated), and **click on a valid destination cell** (visually indicated) to move the piece.
- **Timer Animation**:
  - If a time limit is enabled, then time is displayed for both players and decrements for the player whose turn it is. When time is exhausted for a player, the game ends.
  - Use kivy.clock.Clock to schedule update of time
- **Player Piece Counts**:
  - Display the number of pieces per player, near their timer (perhaps on opposite sides of the screen)
- **Sound Effects**:
  - Each piece movement should trigger distinct sound effects for **Team A and Team B**.
  - Use kivy.core.audio for implementing event based sound effects
  - Jumping sound is unique from growing sound.
  - Converted (captured) pieces should also have distinct sounds when converting.
  - Game end sound.
- **Animations**:
  - Pieces should **animate smoothly** when moving or cloning, using kivy.animation.Animation.
  - Visually distinct animation for jumping versus growing.
  - Additionally, an animation should play when pieces **convert** opponent pieces.
  - Consider animations such as jumping along an arc with squash and stretch, slow-in/slow-out, etc. You might consider animating the position or size of the object, or other properties.
- **Game End:**

  - Game End detected according to all moves exhausted or time runs out (if enabled).
  - If time runs out, the player's whose time runs out loses and the other wins.
  - Game end sound effect.
  - Winning player is indicated on a game end screen or dialog
  - Go back to game start screen after

4. **Extra Credit (Up to +10% Total)**

- **AI Opponent Avatar (5%):** The AI should appear on-screen with changing facial expressions and vocalized sound effects, reflecting good or bad outcomes from their perspective (similar to the arcade version of Ataxx). Also, implement an AI that for each turn **selects the move which maximizes the positive difference** in the total pieces between the two teams. If you wish, you may use more sophisticated strategies, such as minimax.
- **Level Editor (5%):** Implement a level editor which allows the user to create new 7x7 maps choosing the initial locations for playing pieces for each team as well as obstacle placement. The editor should enforce certain requirements such as a matching, non-zero number of starting pieces for each side, at least some free spaces available for each team to make a move, etc.

Resources

- **levels.txt (https://gatech.instructure.com/courses/409430/files/56511895?wrap=1) (https://gatech.instructure.com/courses/409430/files/56511895/download?download_frd=1)**
- See **https://github.gatech.edu/IMTC/CS6456_ClassResources (https://github.gatech.edu/IMTC/CS6456_ClassResources)** for kivy_config_helper.py

Grading

(Note: Grad and undergrad requirements are the same for this assignment.)

Level Selection and Game Configuration 10%

- Game start screen to select game configuration, level, etc.
- Level Select
- Game Mode Select: **Player vs Player** and **Player vs Computer** (if Player vs Computer implemented for Extra Credit)
- Time Mode Select: **Unlimited** or select number of minutes

Game Board Functionality 85%

- Board renders from selected level of levels.txt
- Team pieces, traversable, and untraversable cells are displayed correctly
- All valid moves are supported via click interaction
- A selected piece is visually indicated

- Valid moves are also visually indicated
- Clicking other than a valid move location unselects the selected piece with appropriate visual indications
- A valid move initiates animation towards new board state
- Visual animations for piece movements implemented with kivy.animation.Animation
- Visual animation for captured pieces
- Player piece counts shown on screen for both teams
- Sound effects for making moves, capturing pieces, etc., implemented with kivy.core.audio
- If limited time mode selected, show time count DOWN in MM:SS format for each player
- Game ends when all moves exhausted
- Game end sound effect played
- Game also ends if a player runs out of time (if time limit mode enabled)
- Winning player is acknowledged upon game end
- Return to game start screen after game end

Polish and Aesthetics 5%

- Layout, padding, and alignment, etc.

Extra Credit (up to 10% total)

- AI Opponent with Avatar (up to +5%)
- Level Editor (up to +5%)

Display/Device Independent Rendering (up to 10% reduction of grade if not fully implemented)

- Integrate kivy_config_tool and support various display densities through simulation
- Simulation should be turned off in your submitted code
- All layout, widgets, fonts, should scale appropriately for the given display density setting (real or simulated)
- 10 point reduction to grade for failure to support display density

Submission (up to 10% reduction of grade)

- Submit all required files in a Zip archive:
  - main.py
  - any supporting python files
  - levels.txt
  - sound files (in a sub-directory)
  - (optional) image files (in a sub-directory)
- readme.txt or readme.md (see below for specifics)

Submission

First, please clean your project of any unneeded files. Submit your source and data files in a zip file named *LAST_FIRST_HW5.zip*. The entry point of your program must be named *main.py*. For grading, the TA will run:

```
python main.py
```

Your main.py should have the kivy_config_helper code at the top of the file, initially with simulation turned off. Expect that the TA will evaluate your code with different simulation settings.

Include a readme.txt or readme.md with:

- Full name
- Email
- GT SSO account name
- Detail anything that doesn't work or you didn't complete
- Detail anything that the grader should look for in cases you decided how to implement something that didn't have specific requirements (e.g. how focus is indicated)
- Any build requirements outside of kivy