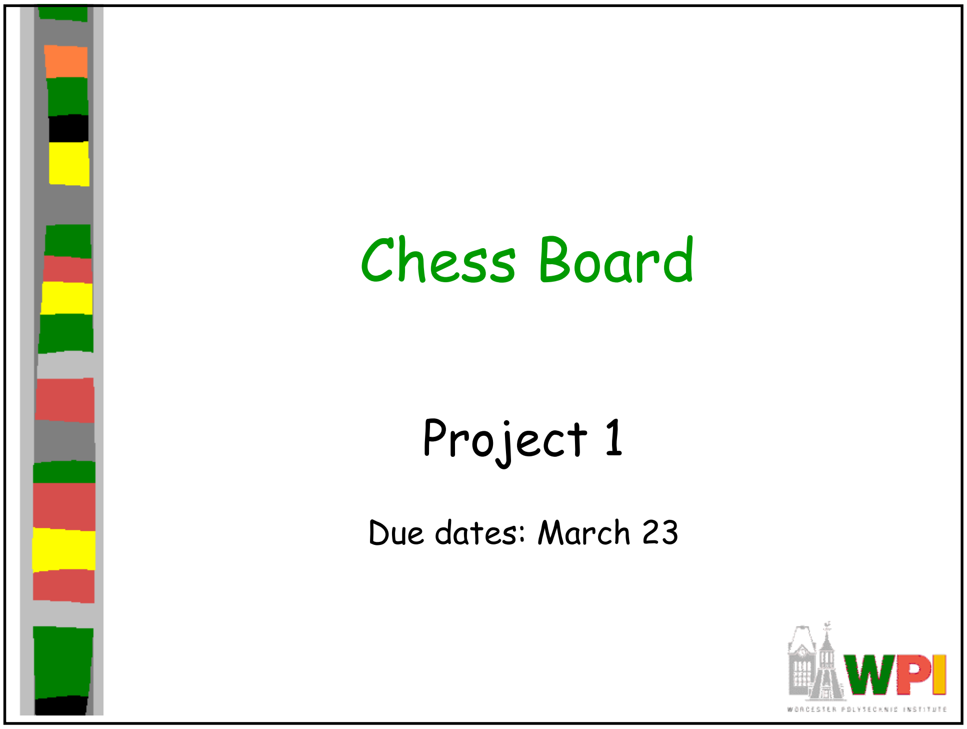
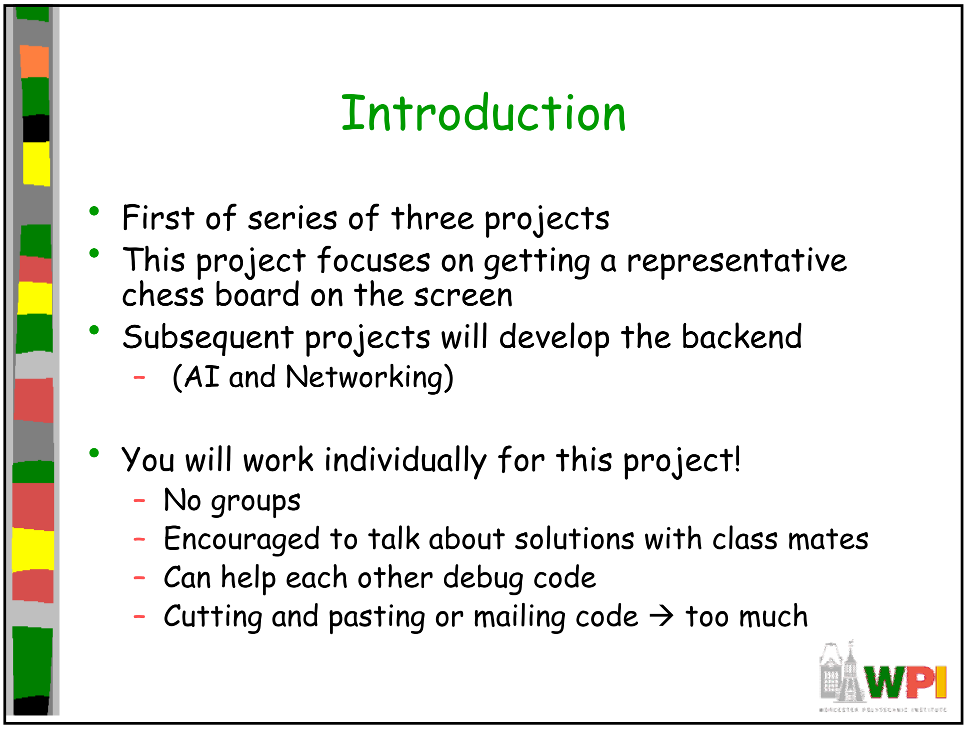


# Chess Board

## Project 1

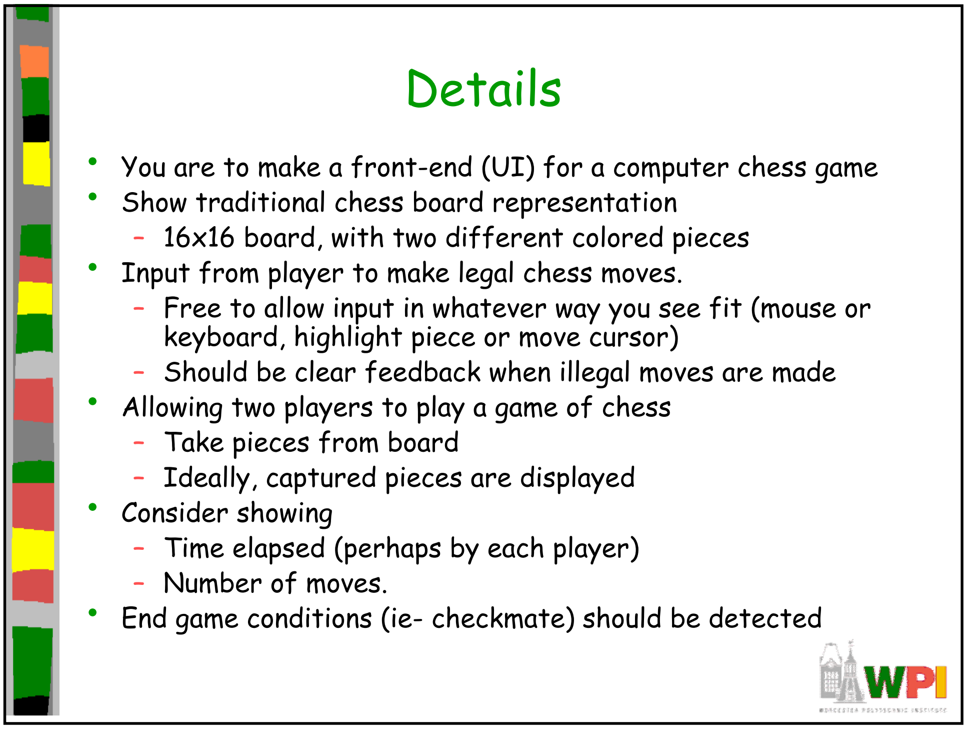
Due dates: March 23



## Introduction

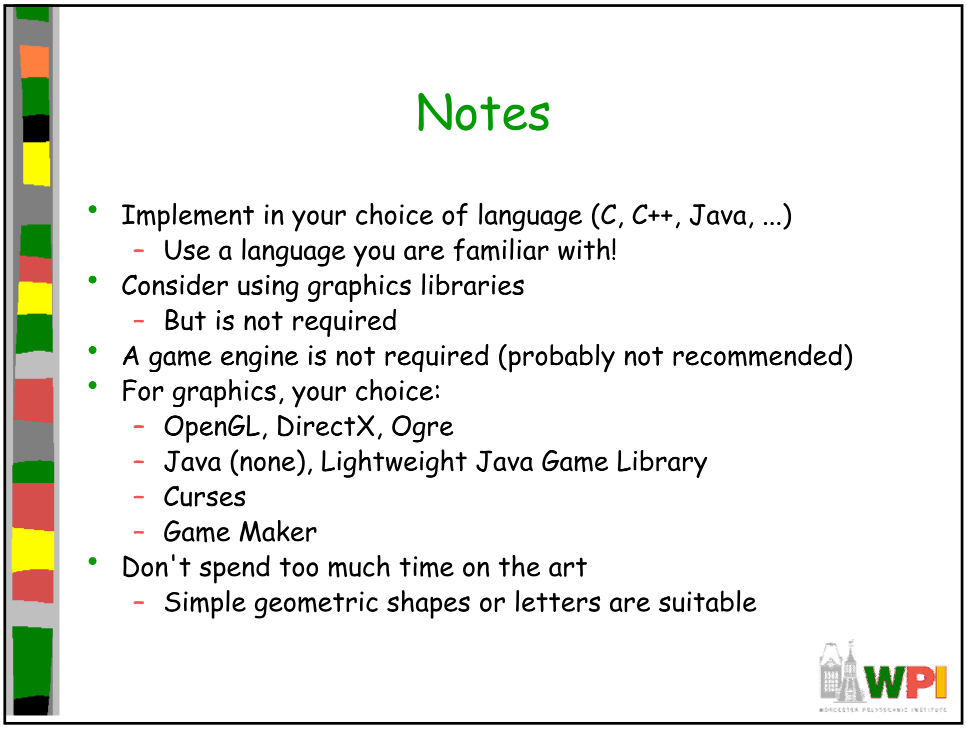
- First of series of three projects
- This project focuses on getting a representative chess board on the screen
- Subsequent projects will develop the backend
  - (AI and Networking)

- You will work individually for this project!
  - No groups
  - Encouraged to talk about solutions with class mates
  - Can help each other debug code
  - Cutting and pasting or mailing code → too much



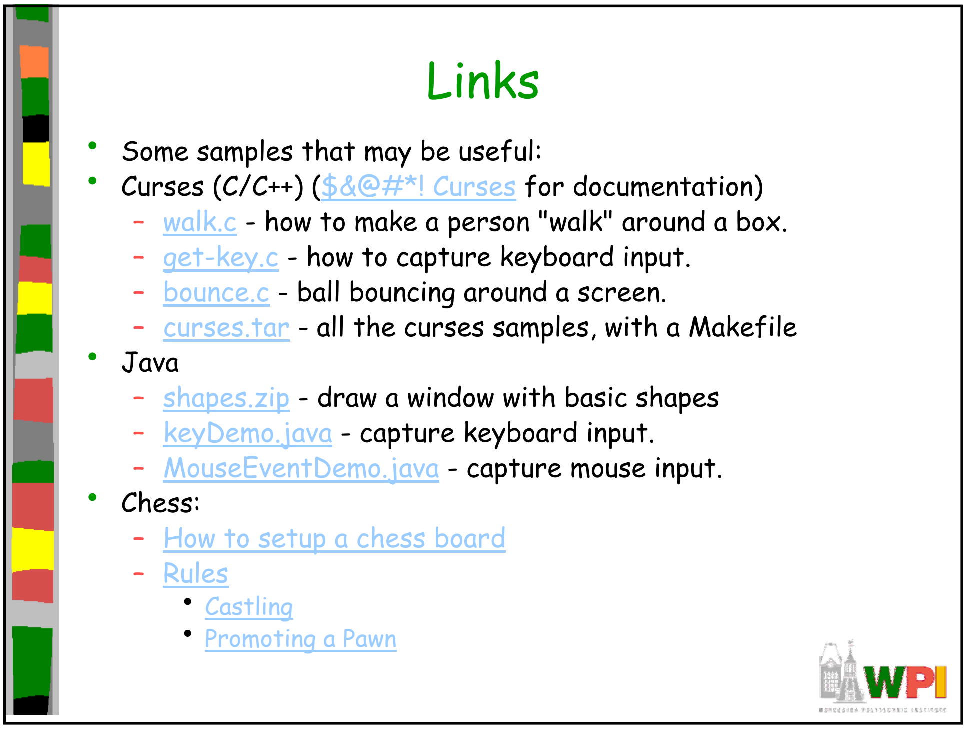
## Details

- You are to make a front-end (UI) for a computer chess game
- Show traditional chess board representation
  - 16x16 board, with two different colored pieces
- Input from player to make legal chess moves.
  - Free to allow input in whatever way you see fit (mouse or keyboard, highlight piece or move cursor)
  - Should be clear feedback when illegal moves are made
- Allowing two players to play a game of chess
  - Take pieces from board
  - Ideally, captured pieces are displayed
- Consider showing
  - Time elapsed (perhaps by each player)
  - Number of moves.
- End game conditions (ie- checkmate) should be detected



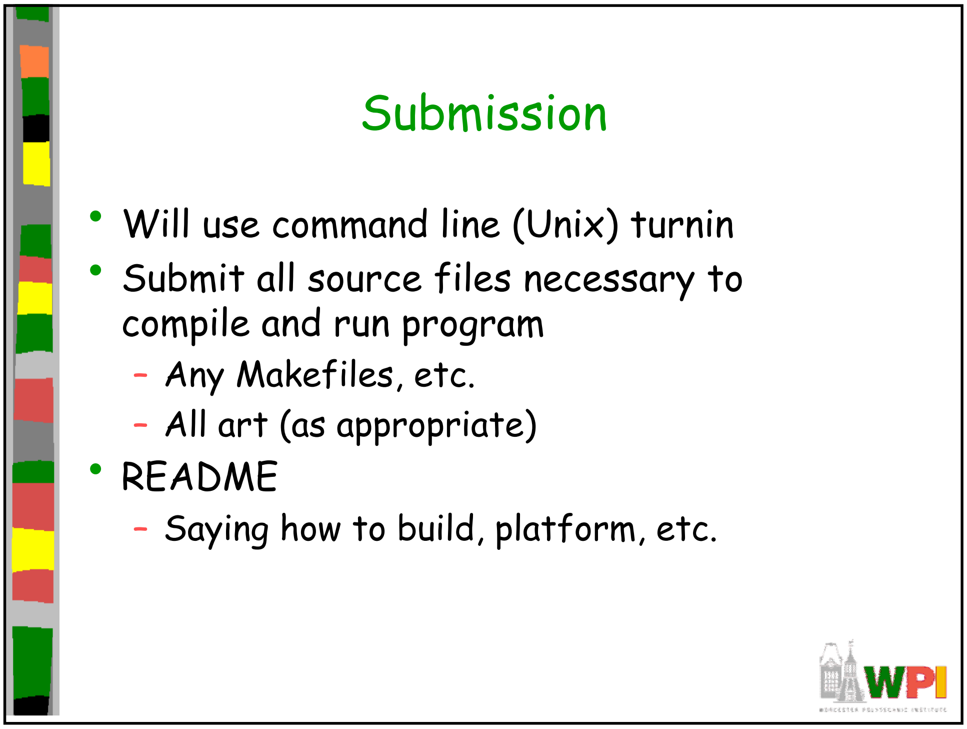
## Notes

- Implement in your choice of language (C, C++, Java, ...)
  - Use a language you are familiar with!
- Consider using graphics libraries
  - But is not required
- A game engine is not required (probably not recommended)
- For graphics, your choice:
  - OpenGL, DirectX, Ogre
  - Java (none), Lightweight Java Game Library
  - Curses
  - Game Maker
- Don't spend too much time on the art
  - Simple geometric shapes or letters are suitable



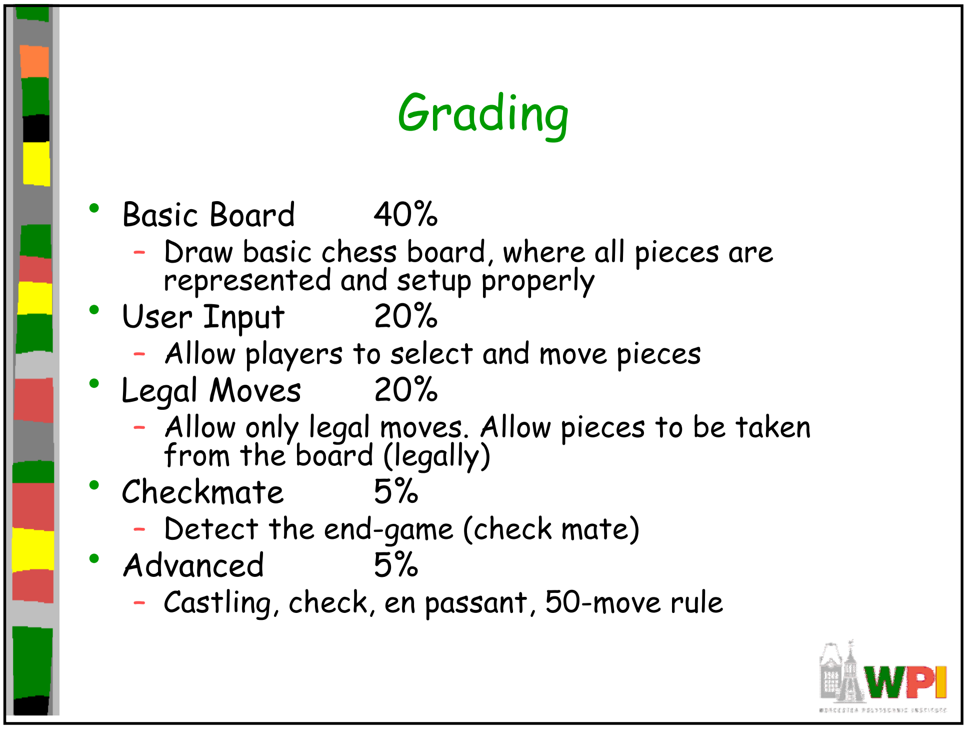
# Links

- Some samples that may be useful:
- Curses (C/C++) ($&@#*! Curses for documentation)
  - walk.c - how to make a person "walk" around a box.
  - get-key.c - how to capture keyboard input.
  - bounce.c - ball bouncing around a screen.
  - curses.tar - all the curses samples, with a Makefile
- Java
  - shapes.zip - draw a window with basic shapes
  - keyDemo.java - capture keyboard input.
  - MouseEventDemo.java - capture mouse input.
- Chess:
  - How to setup a chess board
  - Rules
    - Castling
    - Promoting a Pawn



# Submission

- Will use command line (Unix) turnin
- Submit all source files necessary to compile and run program
  - Any Makefiles, etc.
  - All art (as appropriate)
- README
  - Saying how to build, platform, etc.



## Grading

- Basic Board 40%
  - Draw basic chess board, where all pieces are represented and setup properly
- User Input 20%
  - Allow players to select and move pieces
- Legal Moves 20%
  - Allow only legal moves. Allow pieces to be taken from the board (legally)
- Checkmate 5%
  - Detect the end-game (check mate)
- Advanced 5%
  - Castling, check, en passant, 50-move rule