

## Space Invaders: Video Game Warmup Project

This is a warmup project for the midterm to help lock in your programming skills before tackling a larger game. You must submit an individual, unique project, but are encouraged to collaborate with other students on ideas and programming techniques as you develop your project.

Take the *Paddle* sample game from `codeheart.js` and extend it into your own *slightly* more complex arcade game. Some good choices are: *Pong*, *Space Invaders*, *Breakout*, *Tapper*, *Centipede*, *Tempest*, and *Kaboom*. These are good because, as Koster points out, these are all extremely similar to each other mechanically, but use fiction and changes of perspective to appear as different games.

You have 24/7 access to TCL216 to work on your game, except when another class is in there. TA hours will be held in TCL216 at least until spring break to give you extra support in the evenings. In addition, the Feb 26, Mar 5, and Mar 12 afternoon sessions and March 12 morning session will all move to TCL216 after a brief discussion in TCL206 to give you in-class software development time.

I will evaluate your program based on clarity of the code, evidence of a good-faith attempt to introduce an interesting mechanic, and correctness of the program based on your own stated goals for it. Because this is the first video game project, my expectations are not very high for the complexity of your games. You should build your knowledge and have fun with this assignment. The games that everyone made in the first few hours already crossed the 2.0 grade threshold and everyone will easily do well on the full project. Aesthetic considerations can lift your grade, but you will not be penalized for poor aesthetics (relative to 3.0/4.3 = satisfies the assignment) as long as the game is effectively playable. Some of the best programming tools at your disposal for achieving the code goals are:

- **Good Variable and Function Names**
  - “ball” is a good name. “b” is not (I use short names in lecture when writing on the board to save time. When reading code, we have time and need clarity)
  - Functions abstract actions, so their names should be verbs: `increaseScore()`, `computeGravity()`, `drawBricks()`
  - Objects and arrays abstract state objects, so their names should be nouns: `brickArray`, `ball`, `ammunition`, `health`
  - If you find yourself writing the same code repeatedly, it is time to add a function or a loop
  - Use “constant state” to abstract magic numbers and state names. `var MAX_HEALTH = 100, GAME_OVER_STATE = “GAME_OVER_STATE”`.
- **Comments**
  - A single line beginning with `//` or multiple lines surrounded by `/**/`
  - Explain what code does at a high level and why it is done this way, especially if it is tricky

- Don't repeat what the code already says. "adds one to score" is a useless comment. "Reward the player for destroying a brick" is a good comment for the same line of code.
- **Formatting**
  - Increase your indent level every time that you type { and decrease it when you type }...just like an outline
  - Fully parenthesize complex arithmetic expressions so that there's no ambiguity for the reader: `if (((3 + x) * 4) > 16) { ... }`

Submit your work by 4pm on March 12 using the Turnin folder on OS X from a computer in TCL216:

1. Put your project in a folder named "game1-*yourname*-00". The 00 is the revision number and you can increase it if you need to submit again later.
2. From the Finder menu bar, select the "Go" menu
3. Select "Connect to Server" from the Go menu
4. Type "afp://fuji.cs.williams.edu/" in the Server Address text box.
5. (Optionally push the "+" button to save this choice)
6. Push "Ok"
7. In the new dialog, select "**Guest**" and press "Ok". If you log in as yourself, then I will not be able to access and grade your files.
8. Select the "Courses" volume and press "Ok".
9. In Finder, open the cs107/Turnin folder.
10. Copy your project folder into Turnin. You will not be able to see it once dropped. The original should remain on your own CS account (don't delete it--you'll want to look back at your work later!)
11. If you need to update your solution, increment the revision number and re-upload.