

CS2 Module 2 Week 4 Project: The Game of Hog



Published



Edit



UCB course project: The Game of Hog

In this project, you will develop a simulator for the dice game Hog. You will use the UC Berkeley Electrical Engineering and Computer Science online class (<http://inst.eecs.berkeley.edu/~cs61a/fa15/articles/about.html>) for reference, to walk you through the required steps, and to provide you an initial set of code to build upon. Follow links within this tutorial that reference the [Composing Programs](http://composingprograms.com/pages/11-getting-started.html) (<http://composingprograms.com/pages/11-getting-started.html>) online text, as needed to complete the project.

For the Game of Hog project instructions, please follow the link below to understand the scope of this project, retrieve the initial code, and instructions for each project requirement of **Phase I only, Problems 1 - 4**. **You may work in groups or individually, both in class and for homework this week.** Please submit completed Phase I problems in a single `hog.py` file to Canvas.

<http://inst.eecs.berkeley.edu/~cs61a/fa13/proj/hog/hog.html>
(<http://inst.eecs.berkeley.edu/~cs61a/fa13/proj/hog/hog.html>)

This project includes six files plus images, but **all of your changes will be made to the `hog.py` file**. You may not completely understand the code in the rest of the files, so please don't edit them unnecessarily - make all additions to the `hog.py` file as described in the project description online.

To run the program and to test your extensions, run the `hog_gui.py` file included in the zipped archive for this online project. You should have no need to alter the `hog_gui.py` file, but since you know some Tkinter GUI interfaces now, you could enhance the user interface of the game after you've completed the Phase I project requirements.

Game Summary:

In Hog, two players alternate turns trying to reach 100 points first. On each turn, the current player chooses some number of dice to roll, up to 10. Her turn score is the sum of the dice outcomes, unless any of the dice come up a 1, in which case the score for her turn is only 1 point (the **Pig out** rule).

To spice up the game, we will play with some special rules:

1. **Free bacon.** If a player chooses to roll zero dice (none), she scores one more than the largest digit in her opponent's score. For example, if Player 1 has 42 points, Player 0 gains $1 + \max(4, 2) = 5$ points by rolling zero dice. If Player 1 has 48 points, Player 0 gains $1 + \max(4, 8) = 9$ points.
 2. **Hog wild.** If the sum of both players' total scores is a multiple of seven (e.g., 14, 21, 35), then the current player rolls four-sided dice instead of the usual six-sided dice.
 3. **Swine swap.** If at the end of a turn one of the player's total score is exactly double the other's, then the players swap total scores. *Example 1:* Player 0 has 20 points and Player 1 has 5; it is Player 1's turn. She scores 5 more, bringing her total to 10. The players swap scores: Player 0 now has 10 points and Player 1 has 20. It is now Player 0's turn. *Example 2:* Player 0 has 90 points and Player 1 has 50; it is Player 0's turn. She scores 10 more, bringing her total to 100. The players swap scores, and Player 1 wins the game 100 to 50.
- [_ \(http://inst.eecs.berkeley.edu/~cs61a/fa13/proj/hog/hog.html\)](http://inst.eecs.berkeley.edu/~cs61a/fa13/proj/hog/hog.html)
[_ \(http://inst.eecs.berkeley.edu/~cs61a/fa13/proj/hog/hog.html\)](http://inst.eecs.berkeley.edu/~cs61a/fa13/proj/hog/hog.html)

Points 35

Submitting a file upload

Due	For	Available from	Until
Apr 24 at 10am	Everyone	Apr 18 at 12am	-

Game of Hog Program Rubric			
Criteria	Ratings		Pts
Problem 1: Implementation of roll_dice() function	5.0 pts Full Marks	0.0 pts No Marks	5.0 pts
Overall program design <i>also include comments and error handling</i>	5.0 pts Full Marks	0.0 pts No Marks	5.0 pts
Programming effort, Demo	5.0 pts Full Marks	0.0 pts No Marks	5.0 pts
Problem 2: Implementation of the take_turn function	5.0 pts Full Marks	0.0 pts No Marks	5.0 pts
Problem 3: Implementation of select_dice() function	5.0 pts Full Marks	0.0 pts No Marks	5.0 pts
Problem 4: Implementation of play() function - simulates a full game of Hog	10.0 pts Full Marks	0.0 pts No Marks	10.0 pts
			Total Points: 35.0