

Intro to Computer Science

Alice Whack-a-Mole Game Project

The purpose of this project is to build an engaging and reliable version of a Whack-A-Mole game played using Alice. Users click the mouse on suddenly appearing “moles” to win points. The game should be useable by any WHS student without any instruction other than that provided by the game itself.

You will need to use Alice Lists to complete this project. Follow this [link](#) for information on how to do this, and some starter code for Milestone 1.

The milestones for this project are listed below. They are cumulative: for example, Milestone 5 must include all the functionality of Milestones 1-4.

1	Basic Game	Start with the built-in textbook Chap09WacAMole world. Enhance it to count and display the number of moles the player successfully whacks. Use your score object from Lab 0-5. <i>Note: this is different scoring than the “rising cylinder” that keeps score in the handout – remove that!</i> Moles must appear randomly, not in a fixed pattern.
2	Timed Game	Add a timer that either displays and counts down a fixed time (like 30 seconds) allowed and stops the game when time has expired, or counts up until the player has whacked a fixed number of moles (say, 20). In the first case, the “score” is the number of moles whacked; in the second it is the time taken. Use your timer object from Lab 0-5. Display both the timer and the running count of moles whacked.
3	Custom Game	Replace the WhackAmoleBooth and mole objects with something equivalent of your own design. The basic idea should remain the same: at least 12 “moles” (zombies, grandmas, nerds, etc.) should appear randomly in 12 fixed locations one at a time for whacking. Note that you should deliver Milestones 1-2 with the standard WhackAmoleBooth and moles before converting it to this new world.
4	Multi-Game	Allow repeated plays of the game with a single Alice “Play” of the world. The player should also be able to “pause” and “resume” their game after starting it. Maintain the top score across all these plays and display it after each game, congratulating the user if they have the new high score and letting them enter a short version of their name that is displayed with the score until a higher one replaces it.
5	Multi-Mole	Make at least two different types of moles that can appear randomly in different locations: one that costs the user points (fluffy kittens?) and one that earns double points (teachers?). Note that they must not always appear in the same location – a given location can reveal a standard mole, or either of the special ones at random. They should appear relatively rarely, but each usually at least once in a typical game. Special moles should make a different sound when whacked.
6	Top Scores	The world should maintain and display list of the three best scores,

		congratulating a player if they make the list and letting them enter a short version of their name that is displayed with their score.
7	Better Whack	Animate an object (bopper, ball, bat, etc.) that actually whacks a mole when the player clicks on it successfully, and whacks the empty space when it doesn't connect. <i>This is tricky – hint: invisible objects can trigger the “When mouse is clicked on” event.</i> Make the effect visible on the mole temporarily – a blood spatter, a change in color, etc. And, make it a better sound effect than the built-in “pop” if you haven't already.
8	Levels	Add at least three “levels” of play that make it more difficult, but not impossible, and increase the score for successful whacks. For example, the moles could appear for shorter times, or not rise as high. The levels should increase automatically when the player has reached a certain score, and be selectable as well at the start of a game.
9	Hyper-moles	Add moving moles that appear in truly random locations in the playing area and carry various amounts of points based on how difficult they are to whack.

For full credit for a milestone, the submitted world must include:

- a) A comment in World.myFirstMethod listing your name(s) and a copyright notice
- b) Comments describing what each method does, and how it does it
- c) No “vestigial” code that is not actually used in the world
- d) No duplicate lines of code that could be eliminated by using a list, loop, or method. Ask if you're not sure how you can eliminate them!

Note that this Standard Project provides only for a score of 150 points for completing nine milestones and four Progress Reports. You may propose an additional two Milestones for ten points each, or a similar game with different Milestones as an Alternate Project.