

Overview

Build a web-based game using javascript, html, css of Whack-A-Mole. The genre of Whack-A-Mole games which is described in the following Wikipedia article:<https://en.wikipedia.org/wiki/Whac-A-Mole>.

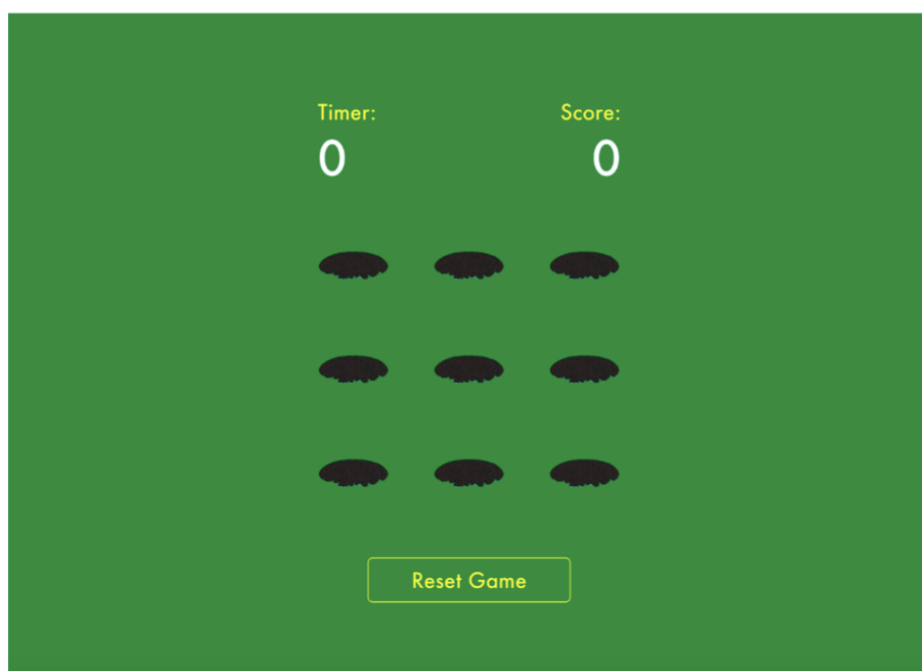
What the game looks like

The game is played in a single round. During the round, moles randomly pop out from a set of holes, and the player's goal is to "whack" the moles with a mallet that the player is holding as quickly as possible. The faster the player whacks the mole, the higher the player's score will be. In the original arcade game, the moles recede back into their holes if the player does not whack them fast enough. The following 30- second YouTube video shows the original arcade game in action: https://youtu.be/_7nPhk-fYFg

Part HTML and CSS. the game should include the following elements:

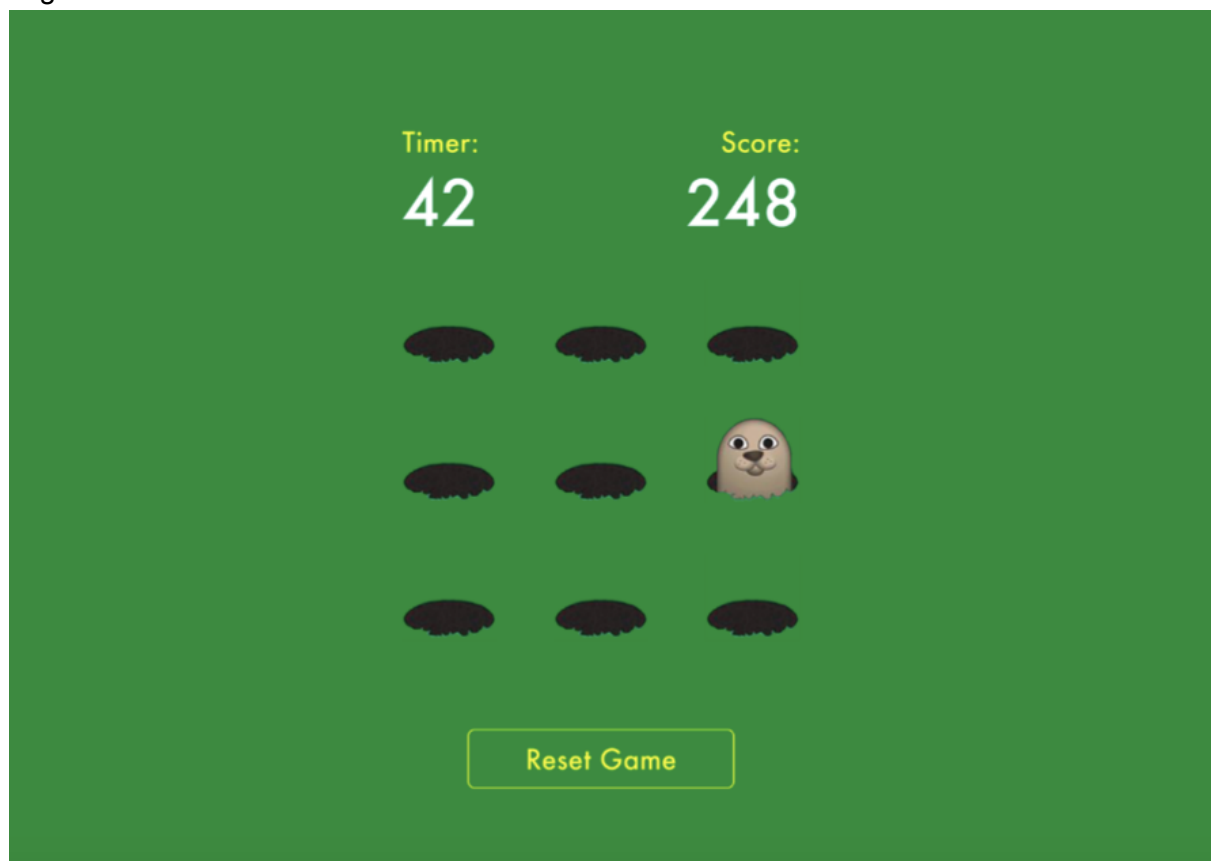
- A 3x3 grid of holes that the moles will pop out from with spacing between the holes. The amount of spacing between the holes is up to you and can be large if you prefer. Please use the hole.png image to represent an empty hole.
- A green background with RGB values of (1, 137, 63) and a hexadecimal value of #01893f behind the grid of holes. The background does not need to apply to the entire page.
- A game timer display
- A score display
- A "Reset Game" button that will start a new round of the game. See below for the function of this.

Below is a sample mock-up to give you an idea. The font pictured below is Futura, if you can't locate it use other font. It is fine for you to use a standard-looking "Reset Game" button — without the special styling shown below.



Please make sure all of the function/behavior listed below can run and work as a game:

1. The “Reset Game” button should set the displayed score to zero, set the timer to some number of seconds that you decide (around 60 seconds preferred), and start counting down the timer.
2. When the timer counts down to zero, the game should freeze, leaving the “Reset Game” button as the only functional element in the game. You should visually indicate that the game is over in some way.
3. During the course of a round (while the timer is counting down), moles should appear pseudo-randomly, and at pseudo-random intervals. a player should not be able to predict when a mole is going to appear, or which hole the mole will appear in. To make the mole appear at a particular hole, you should replace the hole.png image for that hole with the mole.png image. You do not need to animate the mole’s appearance in any way. Once a mole has appeared, it can stay that way for the rest of the round. It should be possible for more than one mole to be present in the playing field.
4. Once a mole has appeared, it should disappear after a random amount of time. A maximum of just a few seconds should be enough, and the minimum should be at least half a second. To make the mole disappear, you can simply revert the appropriate mole.png image to the hole.png image. As with the mole’s appearance, you do not need to animate the mole’s disappearance in any way. Once the mole has disappeared, another mole should be able to randomly appear from the same hole. The image below shows how the game might look mid-round.



5. The moles should be clickable, and clicking the moles should count as “whacking” them with the mallet. This means that when the player clicks a mole, it will immediately disappear (becoming a hole.png image), and its hole should be “live” again, making it possible for another mole to randomly appear there. Empty holes should never respond to click events, and only the “Reset Game” button should respond to click events after the timer reaches zero.

6. Clicking a mole should also boost the player’s score. You may increase the score by either a fixed amount or, optionally, by a varying amount that is greater when the player clicks the mole more quickly.

7. The “Reset Game” button should reset all moles to empty holes as well.