

CS 380

Homework 4

Games in Javascript

Due: Oct 25, 2020, 11:59PM

Form a group of 2 or 3 people. (If you insist, you may work solo; I will not change my grading standards.)

Send email to gtowell380@cs.brynmawr.edu with your group by Oct 7. One email per group is sufficient.

Background:

You will be programming, in Javascript, a game I call “4 is a crowd”.

The basic play is simple, remove objects from a board by tapping on a member of a group of four or more like objects. Groups are composed of objects that touch either right, left, up or down; not on diagonals. When a group is removed, objects “above” the group drop down to fill the space just cleared. Spaces above the dropping items are left empty. The minimum group size is 4.

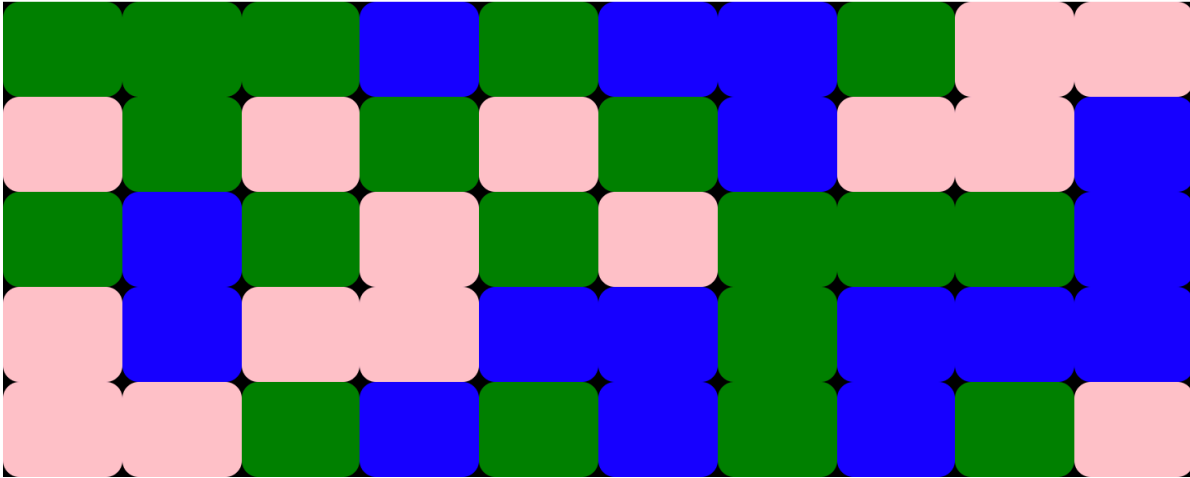
The game proceeds until all groups of 4 or more are removed from the board. At that point the board refills by randomly putting new items into the emptied spaces. The game continues with cycles of removing groups from the board and the board refilling until refilling the board does not result in any removable groups. When the board is full and there are no groups to remove, the game is over.

The game is typically played on a 10 wide by 12-15 deep grid.

For instance, here is a board 5 deep by 10 wide. Groups that can be removed are: green in top left (4), pink in top right(4), blue on right side(6) and green next to blue(5). If the user begins by removing the pink group in the upper right, then the green tile would fall down and join the green group. Thus the green group would go from 5 to 6 members. If the user next removed the group of 6 blues, then then green above the blue group would fall down and the green group would gain another member. (Alternately, if the user removed the green group, then the 2 blues above the green group would fall down and form a bridge between the six and the 3 blues in center bottom, making a group of 11 blues! (Note that removing a group and also remove items from groups, possibly making it less than 4 in size. By strategically removing groups, users can maximize the number of tiles removed from the board, and their score.

Scoring in the game is dependent on the size of the group and the number of colors in the grid. You should make up a scoring function. For example, a workable but I think quite poor scoring function is

$(\text{number in group}) * (\text{number in group}) * (\text{number of colors on board})$
so removing a group of 11 from a board with 3 colors would earn 363 points.



Start the game with a “Welcome” that explains how to play and asks the user for some info: for instance: the number of colors they want to play, the grid size, their name, etc. Then show the grid and begin. Ideally, you should take the input from the user, validate it using Javascript. The Welcome and game play pages could be distinct, but you should have only a small number of page loads.

Some thoughts

(in no particular order)

- If you do not understand the game I have described above, you are free to make up a set of rules that make sense to you. So long as the rules are complete and consistent and result in a playable game, that is fine. (Alternately ask me.) Just be sure to have the rules on the “welcome” page. (The rules must make it possible for me to understand and play the game; so they not only need to make sense to you, they need to make sense to me.)
- It is easier and less confusing to players if you set the starting number of colors and the grid size. But giving users some freedom is good too.
- You might consider non-rectangular grids (or not). I have tried this but my efforts have been less than great.
- Once you have your game playable, have a friend play it (without your help). If your friend cannot figure out how to play, revise your explanation (because it is sure that I will not be able to figure it out) or the game, or both.
- It is almost impossible for a game to be too simple. It is very easy for a game to be too complex.
- The game does not have to be “good” or be fun but it must be understandable, playable and it should look nice.

- 4 is a crowd is a pretty good game for this assignment. Others are also possible. If you want, suggest your own. However, you should not simply create another implementation of a game for which there is already many. So no to “Connect 4”, no to Conway’s “The Game of Life”, no to Tic Tac Toe, etc. I will need to approve if you implement a different, known, game.
- If you are using a game of your own invention, feel free to be inspired by common games. . Again, discuss this with me.
- The grid above is ugly. You can do better.
- Hint: Consider making each box in the grid have a method that responds to the onclick event. Then when the user clicks, check to see if that box is part of a group of 4 or more.
- Consider using the “onmouseover” event so that when the user mouses over a group of 4 or more, the screen changes to show that this is a group that could be removed. (You would also need do do something on the onmouseout event.)
- Consider using a PHP SESSION to store the user’s best score. At the end of the game you can then show the best score and current score.
- Instead of using boring solid colors experiment. Possibly use images. For instance, you could put little images of each group member into each square.
- Consider occasionally increasing the number of colors. For instance, start with 3 colors, then after 2 “drops” go to 4 colors, ...
- As above, feel free to make up rules and to change the game. For instance, you could require diagonal touching rather than horizontal and/or vertical.
- Whatever game you implement, a single play of the game must take less than 10 minutes. For instance, my implementation of 4 is a crowd with the occasional increases in number of colors typically takes less than 5 minutes.
- You may create a 2 player game. However, it should be two players at the same screen (later in the semester you might do a 2-player, communicating game). When I say at the same screen, both players can see the screen at the same time, no hokey “player 1 look away now”. (In the game trade, this is a “2-player, perfect information game”. Chess is an example.) For instance, you could make a 2 player variant of “4 is a crowd” in which the two players make moves to maximize their score and minimize of that of their opponent. (Try it some day, it is actually quite difficult.) (For the sticklers out there, 2 player, 4 is a crowd is not a perfect information game as there is a random component. It is, however, a game in which both players have exactly the same information.)

What To Hand In:

1. The URL of your game

2. If you did something with PHP, the PHP source code. (PHP is not required, nor is it prohibited.)
3. Your javascript should be commented; perhaps not extensively, but there should be comments. Top level functions comments (the arguments, the return value, side effects, etc) are always useful. The comments should be sufficient for you to make sense of your code one year from now.
4. A report (probably less than a page):
 1. The names of the group members
 2. What went well or poorly
 3. Why you chose to implement in the way you did. (This would involve some discussion of options you chose to not follow.)
 4. Other comments that you consider germane.

Presetations:

Approximately half of the class will present their game (The other half will present about the next assignment.) I will look for volunteers in the Oct 14 timeframe. The presentation should focus on coding and how you used javascript to implement the game. This could get really dry and technical. Do your best to make it interesting. I will especially encourage groups to present who implement a game other than 4 is a crowd.