

## Fill-in the dots game

Due: November 17th 2005

### Clarifications

- click in white space makes menu disappear
- dots have to appear randomly with a minimum distance of 20 pixels, they are not allowed to overlap
- if a dot is really close to the edge of the board the pie-menu can appear partially outside the board
- if a dot is color filled already clicking on it does not pop up the menu
- the timer runs from the start until all dots are filled
- the score-times should be displayed after the user has finished
- the list of times should be ordered fastest to slowest with the one of the current trial highlighted
- if the pie-menu pops up the cursor stays over the dot and the menu appears on the left
- for extra credit either move the cursor into the center of the pie-menu or pop-up the menu over the dot and make it semi-transparent

### Background

Menus are a basic way to interact with an interface, in this project you will build a fill in the dots game that requires the user to fill dots with the color specified by the margin of the dot using a pie menu that allows a choice of 4 colors.

### Pie Menus

Pie menus are circular menus that allow the user to make a selection by moving the cursor in the desired direction, in the case of the project there are four. After the cursor is moved in one direction for a specific amount of the diameter (1/5 for the project case) the option in that direction gets highlighted and a click on it selects that option. The pie menu for this project should have a diameter of 40 pixels and be evenly divided in four sections for the four colors, the colors should be centered around the main compass directions (north, west, south, east). Once a selection is made, the menu disappears and the selected value is applied (if correct).



Figure 1: An illustration of how the pie menu works, first step the menu pops up, then the selection is highlighted, then it disappears and the selection is applied.

### Interface Requirements

When the application is started the user is presented with a  $400 \times 400$  pixel grid that is initially empty and on press of a start button gets filled with 10 dots each with a diameter of 10 pixels and a margin size of 2 pixels that are randomly distributed and each have one of the 4 colors. The user then is required to click on each of the dots which invokes the pie menu and select the right color according to the margin of the dot. The menu should appear on top of the dot to color, thus having a semi-transparent menu keeps the target visible. Also, if the user clicks not in a dot, the pie menu should not appear. If the right color is selected the dot is filled and the user can move on to the next dot. On selecting the wrong color the user gets a beep indicating that the selection was wrong. The pie menu disappears in any case, so that in the error case the user has to click on the dot again to invoke the menu. The system should keep track of the time needed to complete the task and display it as feedback throughout the task. Furthermore the system keeps track of the last 5 scores from the current session. You do not have to write results to a file, if the session is started it records the time for completion for five trials (five users) and then uses a first-in first-out principle to handle more results. The interface should have a help function that when called provides simple instructions of how to use the game.

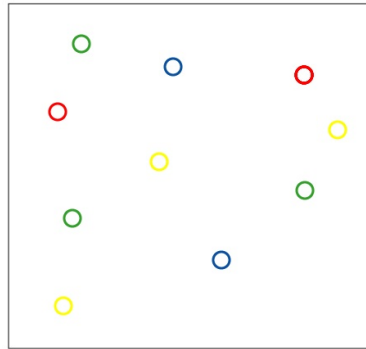


Figure 2: An illustration of how the board can look like with 10 randomly placed dots with randomly chosen colors.

- Game board 400 x 400 pixels
- Pie menu for color selecting with four colors (red, green, blue, yellow)
- Displays 10 dots randomly on press of start with different colors
- A start button, the finish is detected automatically
- A timer that gives feedback about how much time the player needed
- A score list of the last four times
- Help which contains a description of the game

#### Project Guidelines

- You are free to design the interface to your liking based on the principles you learned in class. Use a background and title for the interface. Necessary restrictions such as all dots need to be filled before finishing should be enforced. The filling of the dots requires no specific order as long as all of them are filled.
- Create a single document (1 page, the page should have a title) and it should be no larger than 800x600 and the file size should be less than 800KB.
- The focus is on the interface, and the grading criterion is centered on your interface design and the interactivity.
- This is an individual project.
- You should only use Flash for this assignment.

#### Grading Criteria

The grading for this assignment will be based on three criteria:

1. **Completeness:** Did you do everything that was listed in the requirements? Did you email your assignment with the correct subject? 40%
2. **Graphic Design:** Did you make good use of concepts such as alignment, group, consistency, etc? 15%
3. **Aesthetics:** Did you create an appealing interface by using appropriate fonts, colors, images, etc? 15%
4. **Coding Style:** Did you create a project that is clearly structured and readable for somebody else? (Use of layers, naming of components and behaviors, good use of the library in Flash) Is your ActionScript well structured and commented? 30%

#### Submission Instructions

Submit your assignments electronically, both the .fla file ('<lastname>-<first-initial>-Flash3.fla') and a .swf movie ('<lastname>-<first-initial>-Flash3.swf') of your interface zipped into one zip file named '<lastname>-<first-initial>-Flash3.zip'.

(Refer to Appendix 1 for information on how to export a .fla files as .swf and detailed instructions on naming your file).

Once you have created your zip file, email it as an attachment to the TA (geapi-p3@cs.umd.edu). The subject to the email should be: CMSC434-`<lastname>`-`<first-initial>`-Flash3

The email must be received by **9:30 am on November 10th 2005**.

Appendix I: Exporting a .fla file as .swf

In the menu select the item 'file' and under 'file' select 'export', under 'export' select 'export movie'. You can use the default settings for the properties of the .swf file, please name it according to the instructions below.

(Replace '`<first-initial>`' with your first initial and '`<lastname>`' with your last name. So if your name was Max Schreiber, you should submit a file named schreiber-m-Flash3.zip).