DIG3480c **Assignment 2**

Due By: *04/19/10 6:00pm*
Delivery Method: Make sure your assignment root folder is located and accessible at **http://sulley.dm.ucf.edu/~your_sulley_username/dig3480c/assignment2**

Overview

- PICK ONE -

Game – Write a Processing program to procedurally build an interactive multi-room adventure game.
Name – Write a Processing program to procedurally draw letters as entered by the user.
Gallery – Write a Processing program to procedurally display an interactive image gallery with thumbnails.

- PICK ONE -

Specifications

Implementation

General

- Code should be indented to show parent/child relationships.
- The server path for this assignment should be **~/public_html/dig3480c/assignment2**
- If your assignment includes JPG images, those images should be located at **~/public_html/dig3480c/assignment2/images**
- Your Processing file should be embedded in an HTML page, and that page should be located in the root directory for this assignment. The HTML file should be named **page_index.html**
- The <title> of page_index.html should be **"Assignment 2 – first_name last_name VERSION"**. *(VERSION being Game, Name, or Gallery)*

All directories and filenames should contain **no spaces or uppercase letters.**

Functional Spec

General

- Global variables may not be used in this assignment, and variables must be passed into functions as arguments.
- The user can interact with the script via mouse clicks and movements and/or keystrokes.
- The program should not stop running until the user navigates away from the webpage.

- Your program must use functions, and some of those functions **must receive arguments**.

"Game"

- Your game must consist of three "rooms", and each room must be comprised of shapes and colors generated procedurally. In each room, there should be:
    - A way to move between rooms
        - Movement can be programmed to respond to clicks or keystrokes
    - A way to interact with at least one object
        - Interaction can be programmed to respond to clicks or keystrokes
        - "Interaction", in this case, means that when you click an object, some visual change happens to that object. For example, if your game involved turning on a light switch, when you clicked on a light switch in the down position the program output could light up and change the switch down graphic to a switch up graphic.
    - At least one room with an enemy to interact with
        - Interaction can be programmed to respond to clicks or keystrokes
        - Interaction with the enemy can be as simple as attacking or fleeing, or as complex as to include a visual damage system. Be creative!

"Name"

- The user should be prompted to enter a single letter. Immediately after typing that letter, a function should be called that corresponds to the letter chosen.
- That function will show a procedurally drawn letter corresponding to the key pressed.
    - For example, if the user types the letter 'j', the function to draw 'j' should be called, and 'j' will be procedurally rendered to the screen.
- Typing another letter after the initial 'j' should result in that letter being rendered next to the existing 'j'

"Gallery"

- Part 1:
    - You must use Processing to create at least 2 procedurally generated images. Once the images are created, you should export them as JPG files.
    - The procedurally generated images should be interpretations of 2 photographic images of your choosing. I recommend choosing simple images so that you can have enough time to generate your procedural interpretations.
- Part 2:
    - Your gallery must include thumbnails of both the photographic and procedurally created images.
    - Clicking on a thumbnail should call a function that displays that image at a larger resolution inside of a placeholder area.
    - When the program first runs, no large resolution images should be displayed (only thumbnails).

Presentation

- The size of your canvas for any of the three options should be 500px by 500px.

Content

- Your HTML page will contain the tags and content that the Processing IDE exports. You don't need to edit anything other than the <title> content.

Rubric

/20 Implementation – General specs followed
/30 Functional Spec – General specs followed
/80 Functional Spec – Game/Name/Gallery specs followed
/20 Presentation specs followed

-150 if your assignment path and assignment file name are not exactly as requested in the "delivery method" section above. 'Exactly as requested' means your assignment folder and assignment file name must be named as requested, not contain spaces, and not contain capital letters.

/150 TOTAL