

# Review – Final

CISC1600, Summer Session II 2012

## General Information:

- The midterm will cover each of the topics covered in Unit I and parts of Unit II.
- The final exam will be cumulative covering Units I, II, III and IV
- All test questions will be drawn from the presentations and labs.
- Below are series of questions that relate to each of the topics that we covered. If you can answer all of the questions you can consider yourself prepared for the exam.

## Part 1 – Presentation and Elementary Interface Design

### 1.1 Principles of Interface Design

- a. Define “affordances”.
- b. We discussed four basic principles for interface design in great detail. Give two general principles of interface design with examples.
- c. We discussed things to avoid when designing an interface as well. Give two rules about what should be avoided in interface design with examples.

### 1.2 XHTML and WebPages

- a. Define HTML
- b. Define Tag, Element and Attribute.
- c. List and define 3 HTML tags?
- d. We discussed in class many, many rules that XHTML has that HTML does not. What are those rules?
- e. What is wrong with the following lines of XHTML?

```
<P BGCOLOR=BLUE> This is a new paragraph.  
<B><I>This text is bold and italic</B></I> <BR>
```

- f. XML documents have “minimal components”. What is missing?

```
<html xmlns="http://www.w3.org/1999/xhtml">  
<head>  
<title>simple document</title>  
<body>  
<p>a simple paragraph</p>
```

g. What does it mean when an XHTML document has been “validated”?

*NOTE: On the midterm and final you will be shown the code for a small web-page and will be asked to find, identify AND correct 10 errors on that page.*

### 1.3 CSS (Cascading Style Sheets)

- a. Why use CSS (we discussed 5 reasons in class)?
- b. There are 3 ways to implement CSS on a site, what are they? What is the “best” way?
- c. What the difference in CSS between a class and an id?
- d. What's so useful about the <span> and <div> tags?
- e. What is syntactically wrong with the following lines of CSS

```
h1 [ font-family; Verdana, sans-serif,  
color; red,  
font-size; 20px,  
]  
p, div, h2 [color; #00DDFF, width; 80%, ]
```

- f. Create a custom class called “caution” that will make the text red with a font-size of 20px. This class needs to be able to be used by all tags.
- g. Apply the custom class you created above to word “special” in the section of text below:

The word special has been formatted using a custom class.

### 1.4 Fundamentals of website design.

- a. Regarding links and images, what is the difference between ‘absolute’ and ‘relative’ references?
- b. Knowing how to make a website is one thing, knowing how to make a good website is another. We discussed four general qualities that make for a good website, name one.

*NOTE: On the midterm and final you will be shown some code from a CSS page. It will have mistakes and you will be asked to find, identify AND correct them. You will also be required to create and possibly implement a custom class or custom id.*

## Part 2 – Interactive Programming & Graphics

### 2.1 Programming Languages:

- All programming languages must have a well defined syntax and semantics. Define syntax and semantics.
- What is a paradigm?
- Name three popular programming paradigms. Describe them.

### 2.2 The Internet, World Wide Web, Basic Concepts in Graphics

- What is a protocol? Give one example of an Internet protocol.
- What is the “internet”? How is the “Internet” different from the WWW?
- What are one advantage and one disadvantage of using client/server architecture?
- What’s the difference between “bitmap graphics” and “vector graphics”

### 2.3 Processing

- How does processing implement the imperative concepts of “sequence”, “selection” and “repetition”?
- What is special about the draw() and setup() functions in processing?
- What’s wrong with the program below? Fix the errors.

```
void setup() {
  void draw() {
    void keyPressed() {
      if ( key == 'P' || key == 'p' ) {
        point( 50, 50 )} else if ( key == 'L' || key == 'l' )
      {
        line( 10, 20, 30, 40 )
      } else {
        background( #ffffff )
      }
    }
  }
}
```

- What would the program above (now that you have fixed it) do if you pressed the ‘space’ key?

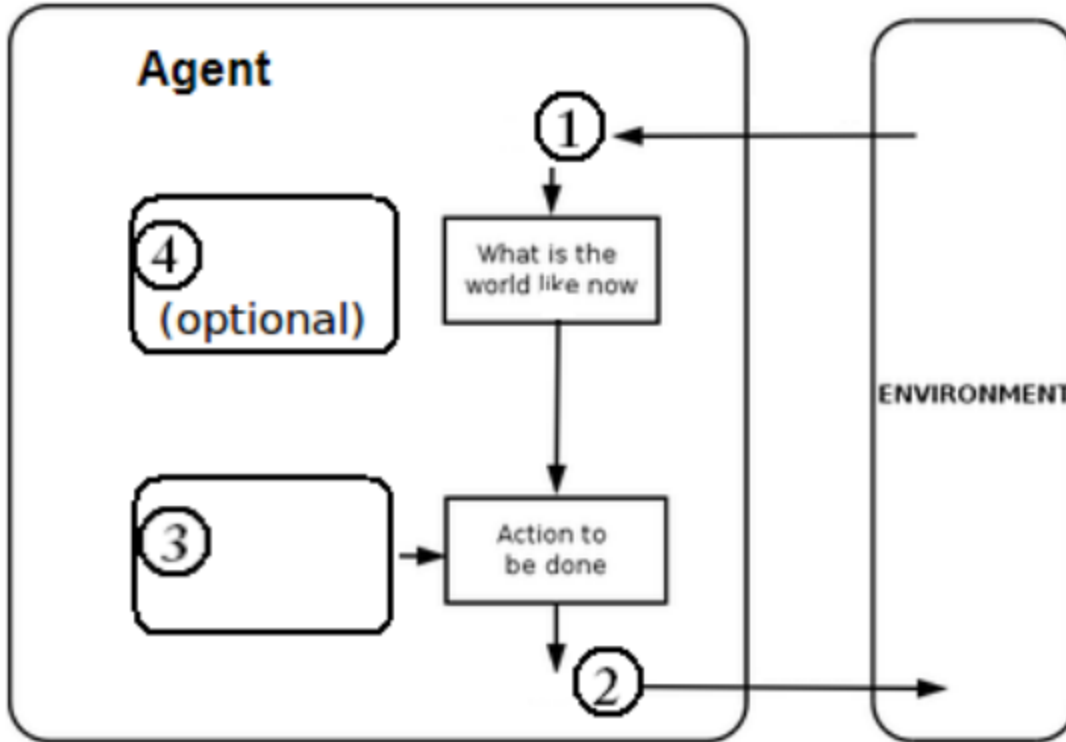
*NOTE: On the midterm you will be shown a basic program in processing and asked to describe what the output screen will look like when the program is given different input (keypressed, mouseclicked etc.). This will be one of the harder parts of the midterm.*

- Define ‘variable’.
- Define ‘event listener’.
- What’s the difference between a vector and a bitmap image?
- What is a function?
- What is an animation?

## Part 3 – Agents, Simulations and Visualizations

### 3.1 Agent Based Programming

- a. What is an agent?
- b. Look at the picture below.



For each of the numbers 1-4, provide a label and a description or example of what is happening in that part of the agent.

- c. We discussed two properties the defined “multi-agent” systems. Name and describe one.
- d. We discussed two advantages, two specific reasons to use multi-agent systems. Name and describe one of them.
- e. Simulations are complex computer models (usually) built using agents. Why bother creating a simulation?
- f. Software agents can take as input analog and digital signals. What’s the difference between the two?
- g. How does Netlogo implement the imperative concepts of “sequence”, “selection” and “repetition”?

**NOTE: On the final you will be asked about NetLogo as a language, but will not need to read or edit code in NetLogo.**

### 3.2 Visualizations:

- a. How is ‘visualization’ different from a ‘simulation’?

### 3.3 Visual Programming Languages

- a. What is a visual programming language?
- b. What is an advantage of using a visual programming language?

## Part 4 – Games Programming

### 4.1 Programming Languages

- a. How does Scratch implement the imperative concepts of “sequence”, “selection” and “repetition”?
- b. What is one of the disadvantages of Scratch as a language (what sort of things can you NOT do in Scratch)?

**NOTE: On the final you will be asked about Scratch as a language. You may also be shown one or more small Scratch scripts, in the Scratch IDE and be asked to describe what that script does, or what will be the results of running that script. You will not need to edit or correct code in Scratch.**

### 4.2 Game Design

- a. We discussed games as the quintessential example of "multimedia programming" We also described how "Game Studies" contains two competing views of how games should be classified and understood. What are those two competing views and how do they differ?
- b. How is "Game Studies" different than "Game Theory"?
- c. Describe in one or two sentences the "Theory of Natural Funativity"?
- d. When the "Theory of Natural Funativity" is applied to humans, we can identify three overlapping categories into which we can divide aspects of enjoyable gameplay. Name and give a brief description of each of these three abstract categories (or spheres of "game functionality").
- e. Along with the theory of funativity we examined concrete components or rules that can also be applied to games to make them more enjoyable. These rules, acquired from years of observation, are also known as heuristics. Briefly describe one of the heuristics (concrete rules) that we discussed in class.
- f. IF you are going to include a narrative in your game, what should be the ultimate goal of that narrative?
- g. What is MDA?
- h. What do the letters in MDA stand for (give examples)?

### 4.3 Game Mathematics

- a. Collision detection, figuring out if two objects are touching, is one of the most common problems in games. Give a brief description of OOT.
- b. We discussed two different methods for reducing the number of tests necessary when using OOT. Name and describe one.
- c. OOT is the most common method for collision detection, but it has limitations. Describe one of these limitations.

### 4.4 Game State

- a. We can define a game as a progression of "states". Define "state"; give an example.