**Midterm (75 points total)**

**Last/Family Name:** Mehta

**Official First Name:** Arnav

**Student ID:** 300196365

Exam Instructions:

* Keep the exam closed at this page until instructed to begin!
* Circle the MOST TRUE answer for multiple choice questions.
* Answer written and diagram questions completely for full marks.

**1. True or False? If any statement is not true, select False as your answer. Only one answer. [1 point each; total 5]**

|  |  |
| --- | --- |
| **True** | **False** |
| x |  |
| X |  |
| X |  |
|  | x |
| x |  |

2. Fill in the blank. [5 points]

1. ( Blocking )
2. ( encoding )
3. ( recursive process )
4. ( rendering ) .
5. ( Perception )
6. [2 points] Based on direct color coding, write an appropriate hexadecimal number matching each color name.

|  |  |
| --- | --- |
| **Color name** | **Hexadecimal Color Values** |
| grey | **#ededed** |
| Magenta | **#ff00ff** |

4. The following diagram shows the duration of different mental model activities during an interactive computing task. Please write down proper mental activities for ***a, b,*** or ***c*** shown on boxes. [3 points]

|  |  |
| --- | --- |
| **a.** | User action |
| b. | Processing time |
| c. | User evaluation |

5. [4 points] Please define terminologies relate to animation techniques.

|  |  |
| --- | --- |
| Key frame | A keyframe is the starting or ending point of a frame where a change will occur |
| In-between | A Something that fills the gap between 2 main scenes |

1. Select all the right options/answers relate to an application of recursive algorithm; Just type a, b, c, d, or/and e for your answer. No need to explain. [2 points].

**Your answer:** (b), (c)

7. Which option is not related to Design implications for attention? [2 points]

**Your answer:**

**(b)**

8. Which option is not related to benefits of interface metaphors? [2 points]

**Your answer:**

**(b)**

**9. [3 points] Pixel Arithmetic Operations**: Match proper pixel arithmetic operations; enter a, b, or c into the box.

Addition: (a)

Division: (b)

Multiplication: (c)

**10. [3 points] Pixel Arithmetic Operations**: The following code example brightens pixel values. To make it complement/negative across RGB channels, modify the example code.

Your answer: Modify the code to make negative pixel effects.

rtotal = (255 – red(img.pixels[loc]));

gtotal = (255 – green(img.pixels[loc]));

btotal = (255 – blue(img.pixels[loc]));

11. To generate the output, fill out the proper code to complete pixel array, **pixels[ ] in 4** sections: **a, b, c & d**

Your answer must match the colors on the output image shown below [8 points]

|  |  |
| --- | --- |
|  | Your answer |
| a) | x+height/2 |
| b) | width-x+y |
| c) | x+y |
| d) | width/2 + y |

12. Briefly explain the difference between top-down and bottom-up processing of visual attention theory. [4 points]

1. Top-down processing

Top-down processing is the method of getting user’s attention guided by higher level cognitive processes like knowledge, goals, etc. We use this method when we want the user to interact with something using their own thought process. An example of this would be problem solving levels in video games.

1. Bottom-up processing

Using bottom-up processing, we get the user’s attention immediately using stimuli in the environment like blinking signs, bright colors or a sudden sound. In this method we do not rely on user’s knowledge but use techniques to capture attention forcefully.

13. [10 points] **Recursive graphics**: Based on the output image shown here, complete **branch**() method using recursive algorithm to approximately generate iterative pattern of shapes below. Carefully review the shape composition, orientation, position, and scale of shapes.

**// your answer**

**void branch(float h) {**

rotate(theta);

pushMatrix();

rect(-h, 0, h, h);

popMatrix();

translate(-h, 0);

branch(h \* 0.7);

}

### [12 points] Submit pde file. **Q14Even\_yourLastnameFirstnameInitial.pde** (e.g., Q14Even\_LeeKJ.pde)

15. [10 points] **2D Transformation**: Based on the output image shown here, complete **diagonalFlip**() method to generate the image mirroring diagonally across both X and Y (flipped horizontally as well as vertically) of the image shown above. You must use **pixels[ ]** operation, not allowed to use translate( ), rotate( ), scale( ) methods (no credit using any of these built-in methods).

Your answer: Design diagonalFlip () to flip pixels across horizontal direction. [10 points]

void diagonalFlip() {

source.loadPixels();

answerImage.loadPixels();

for (int x = 0; x < source.width; x++) {

for (int y = 0; y < source.height; y++) {

color c = source.get(x, y);

int newX = source.width - 1 - x;

int newY = source.height - 1 - y;

answerImage.set(newX, newY, c);

}

}

}