Create PT - Written Response Template

Assessment Overview and Performance Task Directions for Students

Video Submit one video in .mp4, .wmv, .avi, or .mov format that demonstrates the running of at least one significant feature of your program. Your video must not exceed 1 minute in length and must not exceed 30MB in size

Prompt 2a. Provide a written response or audio narration in your video that:

- identifies the programming language;
- identifies the purpose of your program; and
- explains what the video illustrates.

(Must not exceed 150 words)

This is me playing the game I made as a clone of the old lunar lander game, currently featured on seb.ly (http://moonlander.seb.ly/). The goal of this game is to be entertaining. The video is an overview of the basic functionality of the game, written in Javascript on a custom HTML page, styled by CSS. By using the left and right arrow keys, the sprite (lander) will fly to the left and right. If you press the up arrow, the lander will fly up, and play a sound that I downloaded from soundbible and edited in Audacity. I wrote a custom collision detection algorithm based on colors surrounding the player. If you hit the white ground, the player will die, and if you hit the red landing targets, you win. Using simple DOM, the restart button shows up in its beautiful css stolen from codepen (source in the code).

2b. Describe the incremental and iterative development process of your program, focusing on two distinct points in that process. Describe the difficulties and / or opportunities you encountered and how they were resolved or incorporated. In your description clearly indicate whether the development described was collaborative or independent. At least one of these points must refer to independent program development.
(Must not exceed 200 words)

I had several issues over the course of this project, one of which was the collision detection algorithms and fighting a war with security – the browser did not want to allow me to get color information on the canvas, as it was tainted by cross-origin data. The other large-scale issue I stumbled over was the infuriating (and still unsolved) issue that the game does not work in google chrome.

2c. Capture and paste a program code segment that implements an algorithm (marked with an **oval** in **section 3**) and that is fundamental for your program to achieve its intended purpose. This code segment must be an algorithm you developed individually on your own, must include two or more algorithms, and must integrate mathematical and/or logical concepts. Describe how each algorithm within your selected algorithm functions independently, as well as in combination with others, to form a new algorithm that helps to achieve the intended purpose of the program. *(Must not exceed 200 words)*

Code Segment

```
function getCrashedByColor(x,y) {
        // Desperate attempt
        x = Math.round(x);
        y = Math.round(y);
        // console.log("Checking "+x+","+y)
        var pixelData = ctx.getImageData(x , y, 5, 5).data;
        var r = pixelData[0]
        var g = pixelData[1]
        var b = pixelData[2]
        var rgb = r.toString()+g.toString()+b.toString()
        // console.log(r.toString()+g.toString()+b.toString())
        // console.log("R=" +pixelData[0] + " G=" + pixelData[1] + " B = " +
pixelData[2])
        if(r==g && g==b && r!=0) {
                return "crashed"
        } else if(rgb==2372836) {
                return "landed"
        }
}
```

Written Response

This function was called 66 times through the project (some calls were commented out because I didn't need to check collision on every pixel), but the goal was to determine if the lander was over a color, and returned crashed or landed based on what color it is over.

2d. Capture and paste a program code segment that contains an abstraction you developed individually on your own (marked with a **rectangle** in **section 3**). This abstraction must integrate mathematical and logical concepts. Explain how your abstraction helped manage the complexity of your program. *(Must not exceed 200 words) (Must not exceed 250 words)*

Code Segment

```
function drawScoreText() {
        var text = new GameText();
        text.font = "20px Roboto Mono";
        text.fillStyle = "white";

        text.x = 10;
        text.y = 20;
        if(gameStats.frameCount%2||gameStats.frameCount%3) {
                text.draw("Remaining:");
        }

        text.x = 25;
        text.y = 45;
        if(gameStats.frameCount%2||gameStats.frameCount%3) {
                text.draw("Fuel: " + health.remainingFuel + "");
        }

        text.x = 25;
        text.y = 70;
        if(gameStats.frameCount%2||gameStats.frameCount%3) {
                text.draw("Retro: " + health.remainingRetro);
        }
```

```
        text.x = 25;
        text.y = 95;
        if(gameStats.frameCount%2||gameStats.frameCount%3) {
                text.draw("Oxygen: " + health.remainingOxygen);
        }

        text.x = 600;
        text.y = 20;
        text.draw("Frames: " + gameStats.frameCount);
}
```

Written Response
This function helps immerse the player in the old-style of the game, showing a “glitchy” effect where the text flashes at seemingly random times, keeping the old-style sci-fi idea.

Export or save this document as a PDF and turn in to the AP Digital Portfolio along with your **Video** and **Program Code** (separate files).