Create — Applications from Ideas
Written Response Submission Template

Submission Requirements

2. Written Responses

Submit one PDF document in which you respond directly to each prompt. Clearly label your responses **2a – 2d in order. Your response to all prompts combined must not exceed 750 words, exclusive of the Program Code.**

Program Purpose and Development

2a. Identify the programming language and identify the purpose of your program. Explain your video using one of the following:

- A written summary

- of what the video illustrates OR

- An audio narration in your video. If you choose this option, your response to the written summary should read, "The explanation is located in the video."

(Approximately 150 words)

Insert response for 2a in the text box below.

My program is a called "Chemistry 4 Kids". It is a computer game that allows kids to complete three different chemistry experiments. The first experiment is a slime-making lab, the second is a chemical reaction lab, and the third is an exercise in observing pH. The purpose of the program is to condense complex chemistry principles into something children can understand and have fun with. I developed my program using the Snap! program. Through the use of different blocks and Sprites, I was able to easily create my program in a manner that I believe is easier to visualize than traditional JavaScript or HTML.

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; the second could refer to either collaborative or independent program development. *(Approximately 200 words)*

Insert response for 2b in the text box below.

The development of my program was not entirely independent, though I only collaborated with my programming partner to share ideas and talk out complex problems we were facing. The coding of my program was entirely written by myself. Most of the difficulties I faced centered around the many different Sprites I had to use in the Snap! program. Because my program included three different experiments, I faced the problem of Sprites switching costumes depending on the experiment chosen by the user. I solved this problem using broadcasts. When an experiment was chosen, a different broadcast message would be sent telling the program which experiment to begin. I used control blocks to tell the Sprites that different broadcasts would have them switch into the necessary costume for a particular experiment. Another issue I faced was during the development of my algorithm that started the program and directed the program to start the chosen experiment. I solved this problem by creating a variable for the answer given by the user. I then used various if statements so that the program could determine that if the variable was equal to the number of a certain experiment, that experiment should begin.

2c. Capture and paste an image or images of your program code segment that implements the most complex algorithm you wrote. (marked with a **color border** below)


Your algorithm should integrate several mathematical and logical concepts. Describe the mathematical and logical concepts used to develop the algorithm. Explain the complexity of the algorithm and how it functions in the program. *(Approximately 200 words)*

Insert text response for 2c in the plain box below.

I chose to use this algorithm because it is the start to my program. This algorithm uses user input and if statements to help determine which experiment the program should begin. The program asks, through the Scientist sprite, for the user to input the number of the experiment they would like to complete. The program then sets the user's answer to a variable named answer. Then, three different if statements are used to figure out which experiment to start. When the program determines which experiment to start, it broadcasts out a message saying "begin experiment (whichever one the user chose)". These broadcasted messages will help determine which stage costume to use, which Sprites need to be shown, and which costumes the Sprites should be wearing.

2d. Capture and paste an image or images of the program code segment that contains an abstraction you developed (marked with a matching **blue color border** below)
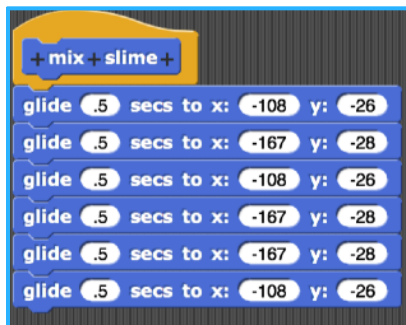
Your abstraction should integrate mathematical and logical concepts.
Explain how your abstraction helped manage the complexity of your program.
(Approximately 200 words)
Insert text response for 2d in the plain box below.

This abstraction was used for the spoon Sprite during Experiment 1, or the slime making experiment. Specifically, the abstraction is used for the mixing of the slime ingredients by the spoon Sprite. An abstraction is used to condense the movement of the spoon Sprite in the bowl. Instead of using six separate glide blocks to tell the spoon to move back and forth within the bowl, I created my own block to condense the code. The "mix slime" block includes six different glide blocks that tell the spoon Sprite to move back and forth so it looks as though the spoon is mixing the ingredients in the bowl to make the slime. After the mix slime block is utilized, the Scientist tells the user that the slime is ready and then a "finish slime" message is broadcasted so that the program knows to move forward with the experiment.