

AP Computer Science Principles: 2020 Create Task
Student Sample F

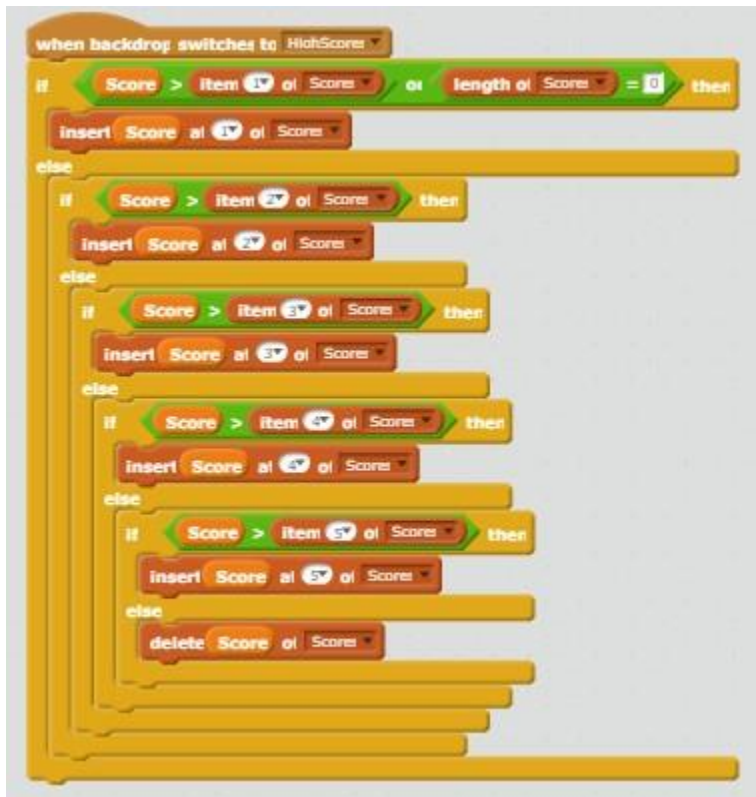
3a)

The video shows the user typing in their username, throwing the dart, getting their score, and then their high score being displayed. One input into the program is the user's name. Another input is the coordinates of the mouse click. The user's name is turned into an output when it is displayed in the high scores list. The input of the coordinates of the mouse is shown as output in the end as a score. The program entertains children by giving them multiple different games to play.

3b)



```
when backdrop switches to GameOverPool
  add Shots to Score
```



```
when backdrop switches to HighScores
  if Score > item 1 of Scores or length of Scores = 0 then
    insert Score at 1 of Scores
  else
    if Score > item 2 of Scores then
      insert Score at 2 of Scores
    else
      if Score > item 3 of Scores then
        insert Score at 3 of Scores
      else
        if Score > item 4 of Scores then
          insert Score at 4 of Scores
        else
          if Score > item 5 of Scores then
            insert Score at 5 of Scores
          else
            delete Score of Scores
```



```
when backdrop switches to HighScores
  show list Usernames
  show list Scores
```

In our program, we made a list to show our high scores called "scores" and usernames called "usernames" at the end of the program. The program collects the username based on what the user types in as their name in the beginning and the scores are collected through the program collecting the user's results. The data in the username list is in the form of a string. The data in the scores list is an

CSP 2020 Sample F

integer. The program code segment that I selected shows how the list “scores” is being used in the program. The list manages the complexity of the program by reducing the amount of lines needed. It reduces the amount of lines because it makes it so we don’t have to display the scores over and over again each time the game is played. Instead, we can just add it to the list so we can put the scores in order and print them all at one time.

3c)



The algorithm shown above throws a dart at the dart board at a random strength/height drop. Then, the algorithm scores the dart based on what color the tip of the dart is touching. It does this by first, showing you how many darts you have left. Then, it only lets you throw the dart if there are 3 darts left because it is the first dart. When the mouse is clicked, the dart goes to where the pointer is and then its y value is

CSP 2020 Sample F

changed randomly. One is taken away from darts left and then the program runs a loop that assigns the dart different scores based on which color the tip of the dart is touching. The algorithm uses sequencing inside of the first if statement when it has to first, wait, then throw, then take away a dart. The algorithm uses selection by having a lot of if statements. These if statements tell the program what to do if the dart is touching certain colors. The algorithm uses iteration by using a forever loop to score the darts.

3d)

One test case I ran was clicking mouse to throw all three of the darts. By doing this, I was testing the ability of the code to throw and score the darts. I expected the dart to be thrown and for its y value to be changed randomly. I also expected the dart to be scored correctly base on what color the tip is touching. When I tested this, I found that it worked correctly. Another test case that I ran was that I threw all of the darts off of the board. By doing this I was testing how the program would score a dart if the tip of the dart is touching a color that has no score to go along with it. I was expecting the dart to have a score of zero, but when I tested it, the program would assign it the score of whatever color it is closest to. I fixed this by adding an if, else statement to the forever loop that assigns the background color a score.