Create PT 20-21 College Board Sample F - Score: 3/6

Total score	Row 1	Row 2	Row 3	Row 4	Row 5	Row 6
Sample: 3	**1**	**1**	**0**	**0**	**1**	**0**

1. **Program Code**
   Your program must demonstrate:
   - output (tactile, visual, or textual) based on input from:
     - the user (including user actions that trigger events); or
     - a device; or
     - a file
   - use of at least one list (or other collection type) to represent a collection of data related to the program's purpose; and
   - development of at least one procedure that uses one or more parameters to accomplish the program's intended purpose, and that implements an algorithm that includes sequencing, selection, and iteration.

   Include comments or acknowledgements for any part of the submitted program code that has been written by someone other than you and/or your collaborative partner(s).

   Create a PDF file that contains all your program code (including comments).

2. **Video**
   Your video must demonstrate your program running, including:
   - input to your program; and
   - at least one aspect of the functionality of your program; and
   - output produced by your program.

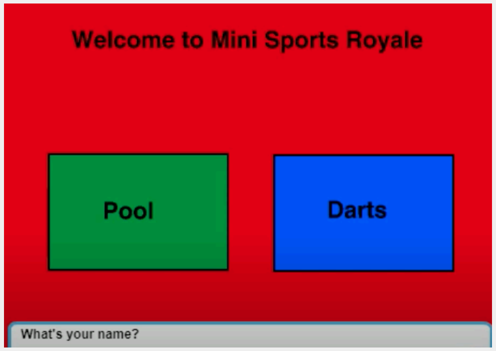
   Your video:
   - must be either .mp4, .wmv, .avi, or .mov format; and
   - must not exceed 1 minute in length; and
   - must not exceed 30 MB in file size.

   Collaboration is not allowed during the development of your video. Your video must not contain any distinguishing information about yourself. Your video must not be narrated, but text captions are encouraged.

3. **Written Responses**
   Submit one PDF file that includes your responses to each prompt below. Clearly label your responses 3a-3d in order. Your responses to all prompts combined must not exceed 750 words, exclusive of the program code. Collaboration is not allowed when answering the written responses.

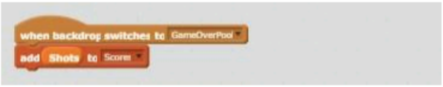
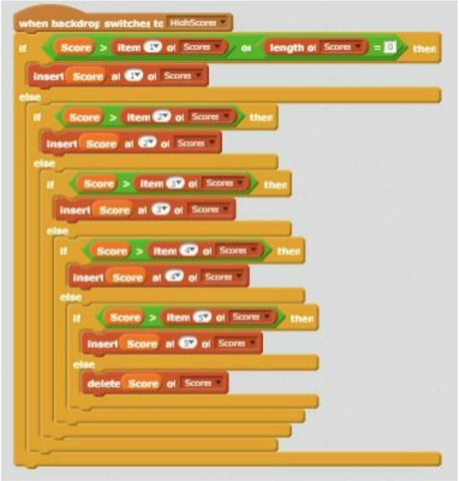
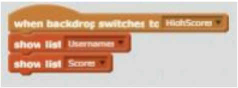
   3a. Provide a written response that:

- describes the overall purpose of the program; and
- describes what functionality the video illustrates; and
- describes the input and output shown in the video.

Student Response	**Scoring Guidelines**	
	Row and Task	**Decision Rules**
 Welcome to Mini Sports Royale Pool Darts What's your name? *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.*	**Row 1** **Video and Written Response 3a** **Program Purpose and Function** **4.A, CRD-2B** ● The video demonstrates the running of the program including: ○ input ○ program functionality ○ output AND ● The written response: ○ describes the overall purpose of the program. ○ describes what functionality of the program is demonstrated in the video ○ describes the input and output of the program demonstrated in the video.	**Consider ONLY the video and written response 3a when scoring this point.** **Do NOT award a point if the following is true:** ● the video does not show a demonstration of the program running (screenshots or storyboards are not acceptable and would not be credited.)
	The response earned the point for this row, meeting all six criteria. ● The video demonstrates the running of the program, including game play input, such as entering the user's name, selecting a game, and clicking where darts are thrown; functionality of gameplay; and output of the score for each dart thrown, as well as a high score list. This satisfies the first three criteria for the video. ● The response describes the overall purpose of creating a program that "entertains children by giving them multiple different games to play." ● The response describes the functionality demonstrated as, "The video shows the user typing in their username, throwing the dart, getting their score, and then their high score being displayed." ● The response describes the input and output shown in the video as, "One input into the program is the user's name," and, "The user's name is turned into an output when it is displayed in the high scores list."	

3b. Capture and paste two program code segments you developed during the administration of this task which contain a list (or other collection type) being used in your program. The first program code segment must show how data has been stored in the list. The second program code segment must show the data in the same list being processed, such as creating new data from the existing data. Then, provide a written response that:

- identifies the name of the list being processed in this response; and
- identifies what the data contained in the list is representing in your program; and
- explains how the selected list manages complexity in your program code by explaining how your program code would be written differently without using this list.

<table>
<tr><th>Student Response</th><th colspan="2">Scoring Guidelines</th></tr>
<tr><td rowspan="3">
when backdrop switches to GameOverPool
add Shots to Score

when backdrop switches to HighScore
if &lt;&lt;(Score) &gt; (item 1 of Score)&gt; or &lt;(length of Score) = 0&gt;&gt; then
  insert Score at 1 of Score
else
  if &lt;(Score) &gt; (item 2 of Score)&gt; then
    insert Score at 2 of Score
  else
    if &lt;(Score) &gt; (item 3 of Score)&gt; then
      insert Score at 3 of Score
    else
      if &lt;(Score) &gt; (item 4 of Score)&gt; then
        insert Score at 4 of Score
      else
        if &lt;(Score) &gt; (item 5 of Score)&gt; then
          insert Score at 5 of Score
        else
          delete Score of Score

when backdrop switches to HighScore
show list Usernames
show list Score

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</td><th>Row and Task</th><th>Decision Rules</th></tr>
<tr><td>**Row 2 - Response 3b**

Data Abstraction

3.B, AAP-1.C

The written response:
● includes two program segments:
○ one that shows how data has been stored in this list (or other collection type)
○ one that shows the data in this same list being used as part of fulfilling the program’s purpose.

● identifies the name of the variable representing the list being used in this response

● describes what the data contained in this list is representing in the program.</td><td>**Consider ONLY written response 3b when scoring this point.**

Requirements for program code segments:
● The written response must include two clearly distinguishable program code segments, but these segments may be disjoint code segments or two parts of a contiguous code segment.
● If the written response includes more than two code segments, use the first two code segments to determine whether or not the point is earned.

Do NOT award a point if the following is true:
● The use of the list is trivial and does not assist in fulfilling the program’s purpose.</td></tr>
<tr><td colspan="2">**The response earned the point for this row, meeting all three of the criteria.**
● The first two code segments presented are used to compute the score. The response includes program code segments showing how data have been stored in the list *Scores*, as well as how another list, *HighScores*, uses and sorts *Scores* to identify the highest-scoring player.
● The name of the list used in the response is *Scores*.
● The response explains that Scores is “a list to show our high scores called ‘scores’,” as well as “data in the scores list is an integer [sic].”</td></tr>
</table>

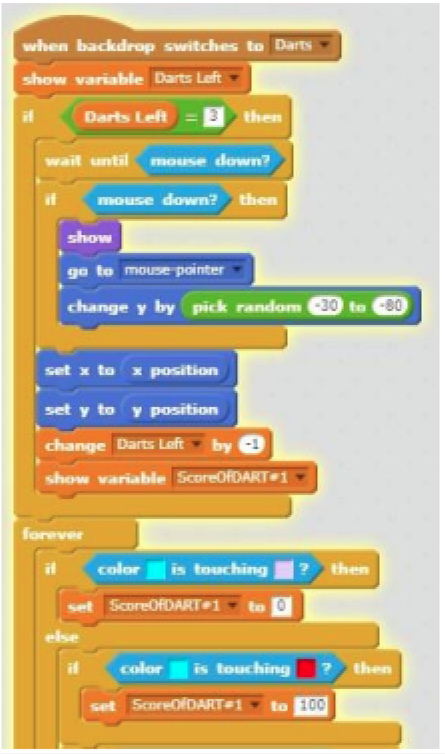
collected through the program collecting the user's results. The data in the username list is in the form of a string. The data in the scores list is an integer. The program code segment that I selected shows how the list "scores" is being used in the program. The list manages the complexity of the program 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.	**Row 3 - Response 3b** **Managing Complexity** **3.C, AAP-3.C** The written response: ● includes a program code segment that shows a list being used to manage complexity in the program. ● explains how the named, selected list manages complexity in the program code by explaining why the program code could not be written, or how it would be written differently, without using this list.	**Consider ONLY written response 3b when scoring this point.** **Responses that do not earn row 2, may still earn this row.** **Do NOT award a point if any one or more of the following is true:** ● The code segments containing the lists are not separately included in the written response section (not included at all, or the entire program is selected without explicitly identifying the code segments containing the list). ● The written response does not name the selected list (or other collection type). ● The use of the list is irrelevant or not used in the program. ● The explanation does not apply to the selected list. ● The explanation of how the list manages complexity is implausible, inaccurate, or inconsistent with the program. ● The solution without the list is implausible, inaccurate, or inconsistent with the program. ● The use of the list does not result in a program that is easier to develop, meaning alternatives presented are equally complex or potentially easier. ● The use of the list does not result in a program that is easier to maintain, meaning that future changes to the size of the list would cause significant modifications to the code.
	The response DOES NOT earn the point for this row. The response does not meet either of the criteria. ● The code segment demonstrates the use of list Scores; however, the use of the list does not manage complexity in the program. If the length of the list grows beyond 5, significant modifications would need to be made for the functionality to be maintained. ● The response states that the list "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." However, it does not explain how the list Scores manage complexity by explaining how the program would be written different without using the list. Additionally, the response does not explain precisely how the program would be more complex if a list was not used to maintain high scores. In fact, in the program code given, use of individual variables instead of a list would lead to essentially the same program.	

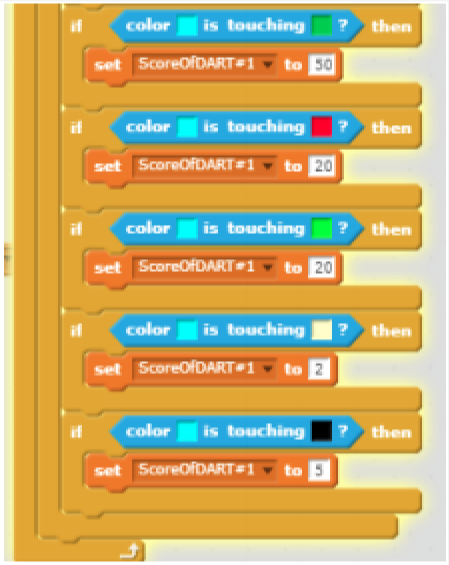
3c. Capture and paste a procedure from your program that you developed during the administration of this task which implements an algorithm used in your program. This procedure must:

- contain and use one or more parameters that have an effect on the functionality of the procedure; and
- implements an algorithm that includes sequencing, selection, and iteration.

Then, provide a written responses that:

- describes what the selected procedure does and how it contributes to the overall functionality of the program; and
- explains how the algorithm implemented in the selected procedure accomplishes its task.

Student Response	Scoring Guidelines	
	Row and Task	**Decision Rules**
 when backdrop switches to Darts ▾ show variable Darts Left ▾ if ⟨(Darts Left) = 3⟩ then wait until ⟨mouse down?⟩ if ⟨mouse down?⟩ then show go to mouse-pointer ▾ change y by (pick random -30 to -80) set x to (x position) set y to (y position) change Darts Left ▾ by -1 show variable ScoreOfDART#1 ▾ forever if ⟨color 🟦 is touching 🟪 ?⟩ then set ScoreOfDART#1 ▾ to 0 else if ⟨color 🟦 is touching 🟥 ?⟩ then set ScoreOfDART#1 ▾ to 100	**Row 4 - Response 3c** **Procedural Abstraction** **3.B, AAP-3.C** The written response: ● includes two program code segments: ○ one showing a student-developed procedure with at least one parameter that has an effect on the functionality of the procedure. ○ one showing where the student-developed procedure is being called. ● describes what the identified procedure does and how it contributes to the overall functionality of the program.	**Consider ONLY written response 3c when scoring this point.** **Requirements for program code segments:** ● The procedure must be student developed, but could be developed collaboratively with a partner. ● If multiple procedures are included, use the first procedure to determine whether the point is earned. **Do NOT award a point if any one or more of the following is true:** ● the code segment is an event handler; OR ● the code segment consisting of the procedure is not included in the written response section; OR ● the written response describes what the procedure does independently without relating it to the overall function of the program.
	The response DOES NOT earn the point for this row. The response met only one of the two criteria. ● The response does not include a student-developed procedure that uses at least one parameter. The procedure backdrop switches is a built-in event. Additionally, there is no code segment included showing a call to backdrop switches.	

 if < color 🟦 is touching 🟩 ? > then set ScoreOfDART#1 ▾ to 50 if < color 🟦 is touching 🟥 ? > then set ScoreOfDART#1 ▾ to 20 if < color 🟦 is touching 🟩 ? > then set ScoreOfDART#1 ▾ to 20 if < color 🟦 is touching ⬜ ? > then set ScoreOfDART#1 ▾ to 2 if < color 🟦 is touching ⬛ ? > then set ScoreOfDART#1 ▾ to 5 *The algorithm shown above throws a dart at the dart board at a random strength/height drop. Then, the algorithm scores the dart bases 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 valueis 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*	● The response does describe what the event does: it "throws a dart at the dart board at a random strength/height drop. Then, the algorithm scores the dart bases [sic] on what color the tip of the dart is touching."	
	Row 5 - Response 3c **Algorithm Implementation** **2.B, AAP-2.H, AAP-2.K** The written response: ● includes a student-developed algorithm that includes: ○ sequencing ○ selection ○ iteration ● explains in detailed steps how the identified algorithm works in enough detail that someone else could recreate it.	**Consider ONLY written response 3c when scoring this point.** **Responses that do not earn row 4 may still earn this row.** **Requirements for program code segments:** ● The algorithm being described can utilize existing language functionality or library calls. ● An algorithm that contains selection and iteration, also contains sequencing. ● An algorithm containing sequencing, selection, and iteration that is not contained in a procedure can earn this point. ● Use the first code segment, as well as any included code for procedures called within this first code segment, to determine whether the point is earned. ● If this code segment calls other student-developed procedures, the procedures called from within the main procedure can be considered when evaluating whether the elements of sequencing, selection, and iteration are present as long as the code for the called procedures is included. **Do NOT award a point if any one or more of the following is true:** ● The response only describes what the selected algorithm does without explaining how it does it. ● The description of the algorithm does not match the included program code. ● The code segment consisting of the selected algorithm is not included in the written response. ● The algorithm is not explicitly identified (i.e., the entire program is selected as an algorithm without explicitly identifying the code segment containing the algorithm). ● The use of either the selection or the iteration is trivial and does not affect the outcome of the program.

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.	**The response earned the point for this row, meeting both criteria.** ● The response includes a program code segment of a student-developed algorithm that includes sequencing, selection (if...then), and iteration (forever loop). Because the forever loop is used purposefully to record where the dart is touching, it can be used to satisfy this requirement. ● The response explains in detail how the algorithm works, including that it “throws a dart at the dart board at a random strength/height drop. Then, the algorithm scores the dart bases [sic] 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 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.”

3d. Provide a written response that:

- describes two calls to the selected procedure identified in written response 3c. Each call must pass different arguments that cause a different segment of code in the algorithm to execute; and
- describes what condition(s) is being tested by each call to the procedure; and
- identifies the result of each call.

Student Response	Scoring Guidelines	
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	**Row and Task**	**Decision Rules**
	Row 6 - Response 3d **Testing** **4.C, CRD-2.J** The written response: ● describe two calls to the selected procedure identified in written response 3c. Each call must pass a different argument(s) that causes a different segment of code in the algorithm to execute. ● describes the condition(s) being tested by each call to the procedure. ● identifies the result of each call.	**Consider ONLY written response 3d when scoring this point.** **Responses that do not earn row 4 may still earn this row.** **Do NOT award a point if any one or more of the following is true:** ● A procedure is not identified in written response 3c or the procedure does not have a parameter. ● The written response for 3d does not apply to the procedure in 3c. ● The two calls cause the same segment of code in the algorithm to execute even if the result is different. ● The response describes conditions being tested that are implausible, inaccurate, or inconsistent with the program. ● The identified results of either call are

<table>
<tr>
<td rowspan="2"><i>it is closest to. I fixed this by adding an if, else statement to the forever loop that assigns the background color a score.</i></td>
<td></td>
<td>implausible, inaccurate, or inconsistent with the program.</td>
</tr>
<tr>
<td colspan="2"><b>The response DOES NOT earn the point for this row. The response does not meet any of the criteria.</b>
● The response does not describe two calls to the backdrop switches procedure using different arguments. Instead, the response describes two paths in the program code based on user input during execution of the code rather than two calls to the procedure from another part of the program code that lead to different behaviors.</td>
</tr>
</table>