|  |  |  |
| --- | --- | --- |
|  | **Create PT - 2b** |  |

**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)*

|  |
| --- |
| **Scoring Guidelines** |
| **Row and Task** | **Decision Rules** |
| **Row 2 - Response 2B**Describes or outlines steps used in the incremental and iterative development process to create the entire program. | **Do NOT award a point if any one of the following is true:** * the response only includes the process for determining the program idea and does not address the development process used to create the entire program; or
* the response does not indicate iterative development;
* refinement and revision are not connected to feedback, testing, or reflection; or
* the response only describes the development at two specific points in time.
 |
| **Row 3 - Response 2B**Specifically identifies at least two program development difficulties or opportunities. **AND**Describes how the two identified difficulties or opportunities are resolved or incorporated. | Response earns the point if it identifies two opportunities, or two difficulties, or one opportunity and one difficulty AND describes how each is resolved or incorporated. **Do NOT award a point if any one of the following is true:** * only one distinct difficulty or opportunity in the process is identified and described; or
* the response does not describe how the difficulties or opportunities were resolved or incorporated.
 |

|  |  |
| --- | --- |
| **Student Response A - [**[**Video**](https://secure-media.collegeboard.org/ap/video_audio/ap18-create-sample-a-video.mp4)**] [**[**Written Response**](https://secure-media.collegeboard.org/ap/video_audio/ap18-create-sample-a-written.pdf)**]** | **Scoring Guidelines** |
| I independently developed this program starting with my original method and continuing to build off of that. During the process, as I would think of an element that I needed to add to the game, I would work on the code needed for that certain part of the game. An element that gave me some trouble was the first part my wolfMove method which was meant to have the wolf move to blue chickens and eat them. I had a few problems with this method because at first, my wolf would move, but end up halfway underground. After a lot of trial and error, I was able to make the wolf move up while moving forward and make sure that when it stopped moving it was above ground. I was also presented with opportunities to add aspects to make my game better. One of these moments was when I decided that my game was too easy to play. To make the game more fun, I looked up a tutorial on how to make a countdown timer. Once I added the timer to the game, there was more pressure, which made the game realistic.  | **Row 2** | **1** |
| **The response earned a point for this row.** The response describes the iterative process in developing the program: "starting with my original method and continuing to build off of that. During the process, as I would think of an element that I needed to add to the game, I would work on the code needed for that certain part of the game." |
| **Row 3** | **1** |
| **The response earned a point for this row.** The response identifies a **difficulty**: "the first part my wolfMove method." The response indicates that it is resolved "after a lot of trial and error ... able to make the wolf move up while moving forward and make sure that when it stopped moving it was above ground." The response identifies an **opportunity** "to add aspects to make my game better ... the game was too easy to play." The opportunity is addressed when a timer was added to the game: "there was more pressure, which made the game realistic." |
| **Student Response B - [**[**Video**](https://secure-media.collegeboard.org/ap/video_audio/ap18-create-sample-b-video.mp4)**] [**[**Written Response**](https://secure-media.collegeboard.org/ap/video_audio/ap18-create-sample-b-written.pdf)**]** | **Scoring Guidelines** |
| An incremental process​ was when I drew a flowchart detailing the flow of the program​, including how the program has three functions collect\_dict(), study(), quiz(), that collects term-definition-pairs, helps user study, and quizzes the user, respectively, and a fourth function run\_quiz() that calls the three former functions in the correct order. Another incremental process​ ​incorporated into the code is a for loop​ that prints blank lines until the counter reaches 100. An iterative process​ was ​when I decide​d to improve ​the program by​ using the lower() method to make it non-case-sensitive​ for users. Both these processes were done ​independently​. A ​difficulty encountered​ was ​when my tester found that the program​ always calculate​s the user’s quiz-scores 1-digit higher​ than it actually is. I resolved​ this by assigning ​points\_scored​ ​to 0​. Another difficulty​ was encountered when​ my tester noticed that the quiz feature was printed directly below the “study” feature and users taking​ the quiz can cheat off the answers displayed in the study section​. I resolved this by printing​ 300 blank lines ​between the “study” and “quiz” portions. These 2 difficulties were resolved by collaborating ​with a classmate. | **Row 2** | **1** |
| **The response earned a point for this row.** The response describes the incremental process used to create the program. The response lists these steps: "drew a flowchart, incorporated a for loop that prints blank lines, decided to improve the program using the lower() method." The iterative process occurred during the difficulty where testing was used to find a problem where the "program always calculates the user's quiz-score 1-digit higher than it actually is" and the response includes how it was resolved. |
| **Row 3** | **1** |
| **The response earned a point for this row.** The response describes a **difficulty** ("calculates the user's quiz-scores 1-digit higher than it actually is") and its resolution ("assigning points\_scored to 0”). The response includes a second **difficulty** ("users taking the quiz can cheat off the answers displayed in the study section") and its resolution ("printing 300 blank lines between the 'study' and 'quiz' portions”). |
| **Student Response C - [**[**Video**](https://secure-media.collegeboard.org/ap/video_audio/ap18-create-sample-d-video.wmv)**] [**[**Written Response**](https://secure-media.collegeboard.org/ap/video_audio/ap18-create-sample-c-written.pdf)**]** | **Scoring Guidelines** |
|  I completed the program independently. I started out by setting up the background and arranging the cards before developing the Card class. After looking at the pixel size of the cards, I determined the coordinates I needed to use. I tested the program multiple times to make sure I got the coordinates and placement of the cards right with each version of the positioning algorithm. I was able to visually confirm the accuracy of the program by running it to check if it worked. Then, I worked on getting them to flip. Lastly, I developed the algorithm that tests if the card flipped had an x behind it and made the Gameover class to end the game. At first, I wanted to make the program work so that whenever I clicked the card, the card would flip. However, I had trouble implementing Greenfoot’s mouseClicked() method. In the act() method of the Card class, I called the turnCard() method I wrote, which would flip the card. Although I used the mouseClicked() method properly, it did not flip the card; the image of the card did not change as intended. To solve this issue, I replaced the mouseClicked() call with a call to Greenfoot’s isKeyDown() method in the same location to flip the card based on pressing its order value (1-9). Each card is instantiated with its order value, so after assigning that parameter to a private instance variable, I could use that variable as a parameter in the isKeyDown() call. I also needed an easier way to restart the game, so I developed a way to press r in order to reset the background when the Gameover screen pops up. With an if statement checking if the user presses “r” in the act() method of the Gameover class, pressing “r” would reset the cards.  | **Row 2** | **1** |
| **The response earned a point for this row.** The response describes the incremental process to develop the program: "setting up the background and arranging the cards before developing the Card class ... determined the coordinates ... tested the program multiple times ... confirm the accuracy of the program by running it to check if it worked ... getting them [cards] to flip." |
| **Row 3** | **1** |
| **The response earned a point for this row.** The response identifies a **difficulty** ("trouble implementing Greenfoot's mouseClicked() method") with a resolution ("replaced [it] with a call to Greenfoot's isKeyDown() method in the same location"). The response identifies a second **difficulty** ("needed an easier way to restart the game") and its resolution ("developed a way to press r in order to reset the background when the Gameover screen pops up"). |
| **Student Response D - [**[**Artifact**](https://secure-media.collegeboard.org/ap/video_audio/ap18-explore-sample-d-artifact.mp4)**] [**[**Written Response**](https://secure-media.collegeboard.org/ap/video_audio/ap18-explore-sample-d-written.pdf)**]** | **Scoring Guidelines** |
| The bulk of my project was created independently, but I had a minimal amount of help from a partner. I started off by planning out how to make the bowling pins get knocked out as the bowling ball went forward to hit them. While doing this, I encountered a huge problem in my code. Every time the bowling ball touched the bowling pins, they didn’t disappear although the code specifically stated that they should disappear once they were in contact with one another. My friend helped me with this by suggesting that I should make a code for the bowling pins so that when the bowling pins broadcasted messages and the bowling ball received them, the program would run smoothly and the bowling pins would disappear every time the bowling ball received their individual broadcast message. Another issue I faced was when all the pins were down, the host wouldn’t say “Great job!” I resolved this problem by broadcasting a message when the last pin fell down so that when the host received the message, he would say “Great job!” | **Row 2** | **0** |
| **The response DID NOT earn a point for this row.** The response does not describe the overall development process of the entire program. The response focuses almost entirely on the two difficulties. |
| **Row 3** | **1** |
| **The response earned a point for this row.** The response earned the point for this row. The response describes two difficulties. The first **difficulty** is that "every time the bowling ball touched the bowling pins, they didn't disappear." This is resolved by making the code for the bowling pins broadcast messages for the bowling ball to receive. The second **difficulty** is that "when all the pins were down, the host wouldn't say ‘Great job!’" This is resolved by "broadcasting a message when the last pin fell down." |
| **Student Response E - [**[**Video**](https://secure-media.collegeboard.org/ap/video_audio/ap18-create-sample-e-video.mp4)**] [**[**Written Response**](https://secure-media.collegeboard.org/ap/video_audio/ap18-create-sample-e-written.pdf)**]** | **Scoring Guidelines** |
| There were many problems that arose while coding the program. One of the early problems encountered was deciding how I should set up my study guide. For example, I could have chosen to do flashcards along with doing multiple choice. However, I felt that the flash cards would be more effective and efficient way of creating this app. Furthermore, when making this study guide app, I felt that there needed to be something else that could have made the study guide more useful for the reader. Originally, there were just flashcards, but I felt there was something else that could be done. So I included another button that allowed the user to type in the definition as the word was being given. This was a major development addition as it is more effective for the user to write the information than by just looking at cards. This is also more effective for memorization.  | **Row 2** | **0** |
| **The response DID NOT earn a point for this row.** The response does not describe the incremental or iterative process used in developing the entire program. The response focuses on two decisions that were made in determining what would be in the program. |
| **Row 3** | **0** |
| **The response DID NOT earn a point for this row.** The response identifies an **opportunity** as adding functionality to allow users to enter the "definition as the word was being given." This is resolved by including "another button." The **difficulty** identified is the decision to use flashcards over multiple choice, which is a **design choice,** not a program development difficulty. |
| **Student Response F - [**[**Video**](https://secure-media.collegeboard.org/ap/video_audio/ap18-create-sample-f-video.mp4)**] [**[**Written Response**](https://secure-media.collegeboard.org/ap/video_audio/ap18-create-sample-f-written.pdf)**]** | **Scoring Guidelines** |
| There were several problems that were presented in the code. One problem was setting the timer to countdown 30 seconds and stopping the time when the pause button was clicked, then resuming the time again. This was an independent development. I had to find out how to resume the game, such that the time that was paused did not restart again after clicking the resume button. To solve this, I made a function called Time and called it in the event handler of the resume button, which in result would change just the time in intervals of 1000 milliseconds, but not move back to 30 seconds. Another problem, also an independent development, was updating the high score the user received every time the score was higher. The code would update the high score, but to the score the user received, regardless if it was higher or not. I decided to add in an if statement, such that if the score the user received was greater than the score from other games, it would update the high score value to that score. | **Row 2** | **0** |
| **The response DID NOT earn a point for this row.** The response does not describe the incremental and iterative design process for the entire program. |
| **Row 3** | **1** |
| **The response earned a point for this row.** The response identifies two difficulties and how they are resolved. The first **difficulty** is "setting the timer to countdown 30 seconds and stopping the time when the pause button was clicked." Its resolution is "a function called Time and called it in the event handler of the resume button, which in result would change just the time in intervals of 1000 milliseconds, but not move back to 30 seconds." The second **difficulty** is "updating the high score the user received every time the score was higher." Its resolution is "to add in an if statement, such that if the score the user received was greater than the score from other games, it would update the high score value to that score." |
| **Student Response G - [**[**Video**](https://secure-media.collegeboard.org/ap/video_audio/ap18-create-sample-g-video.mp4)**] [**[**Written Response**](https://secure-media.collegeboard.org/ap/video_audio/ap18-create-sample-g-written.pdf)**]** | **Scoring Guidelines** |
| One difficulty I faced was figuiring out how to get the Snake to grow. To solve this problem I made the snake create a clone every time it got an apple. When there are new clones, it waits a while then deletes some of the clones so the snake isn't too long. This was done independently. Another problem I had was channging the speed of the snake. To solve this I had to add a certain amount of steps to the original speed of the snake every time it got an apple. This was done collaboratively. | **Row 2** | **0** |
| **The response DID NOT earn a point for this row.** The response does not describe the overall development of the entire program. |
| **Row 3** | **1** |
| **The response earned a point for this row.** The response identifies two difficulties during the program development. The first **difficulty** is "figuring out how to get the Snake to grow" and is resolved by making "the snake create a clone every time it got an apple ... it waits a while then deletes some of the clones so the snake isn't too long." The second **difficulty** is "changing the speed of the snake." This is resolved by "add a certain amount of steps to the original speed of the snake every time it got an apple." |
| **Student Response H - [**[**Video**](https://secure-media.collegeboard.org/ap/video_audio/ap18-create-sample-h-video.mp4)**] [**[**Written Response**](https://secure-media.collegeboard.org/ap/video_audio/ap18-create-sample-h-written.pdf)**]** | **Scoring Guidelines** |
| The main goal of the app I created is to have people learn new words in different languages. The quiz works as follows: If the correct answer is clicked, then the next question appears; if the wrong one is clicked, then the quiz restarts. The main structure of the app is a series of onEvent functions to lead you from one question to the next; this series of repeated onEvent functions exemplifies repetition, a key element in learning a new language. To make the app more interactive, I decided to add in user input on three of the twelve questions. I had trouble with having the user input box cleared after the check button is clicked. I kept running the app, but came back to the same problem. It was after a couple times that I realized that I needed to add in a setText function to clear the input box every time the check button is clicked. This app was entirely an independent project, with no outside collaboration.  | **Row 2** | **0** |
| **The response DID NOT earn a point for this row.** The response does not describe the incremental and iterative development process for the program. Instead, the response describes how the program functions and states the main components of the program code. |
| **Row 3** | **0** |
| **The response DID NOT earn a point for this row.** The response does not identify two difficulties and/or opportunities and how they are resolved or handled. The response only identifies one **difficulty**, clearing the user input box. |
| **Student Response I - [**[**Video**](https://secure-media.collegeboard.org/ap/video_audio/ap18-create-sample-i-video.mp4)**] [**[**Written Response**](https://secure-media.collegeboard.org/ap/video_audio/ap18-create-sample-i-written.pdf)**]** | **Scoring Guidelines** |
| Whilst developing the code for the program, two main problems came about. One main problem was to learn how to get the app to display in the countdown slot in the screen the values chosen by the user preliminary set by me on the dropdown text. This was an independent development. At first, my idea was to implement many numbers from where to choose from, but that would had resulted in an exponentially longer code to map to the screen and only giving the option of one-digit values. Therefore, I decided to implement a dropdown textbox with predetermined values as to now only map the value of it to the screen, resulting in a successful simplification of my code. Second problem was the most essential to fix, how to subtract one progressively. This was also resolved independently. For this I implemented a function that would repeat itself every second, which would take countdown and subtract one from its value until it reached zero. But to keep it from going into negative numbers, I implemented a function which would reset the value of countdown and set it to zero. Which took less functions and is more cleaner than implementing a for-loop. | **Row 2** | **0** |
| **The response DID NOT earn a point for this row.** The response does not describe the overall interactive or iterative development of the entire program. |
| **Row 3** | **0** |
| **The response DID NOT earn a point for this row.** The response states that "One main problem was to learn how to get the app to display in the countdown slot..." However, this is not a problem or opportunity during the program development process, **it's a design issue.** The second difficulty and its resolution are given. |
| **Student Response J - [**[**Video**](https://secure-media.collegeboard.org/ap/video_audio/ap18-create-sample-j-video.mp4)**] [**[**Written Response**](https://secure-media.collegeboard.org/ap/video_audio/ap18-create-sample-j-written.pdf)**]** | **Scoring Guidelines** |
| In our app " Sports Trivia", it included player's faces and team logos for the participant to guess from right to wrong. We have a title screen which shows which sport you want to be questioned on. The game consists of 2 questions for each sport that has to be answered. So basically whatever sport you want you can choose it from the welcome screen and it'll bring you to the desired sport you want to be questioned on. It was hard deciding which particular questions my partner and i would come up with because we would always have debates on which question is the hardest or easiest. Also, it was hard figuring what genre our game would be based on, there are plenty of different ideas for a trivia game. Developing the game was a hardship, figuring out our code and transitions for each screen was always a debate, there was a lot of testing and bugs along the way but we came to a final conclusion for all these tough situations. | **Row 2** | **0** |
| **The response DID NOT earn a point for this row.** The response does not describe the incremental and iterative development of the entire program. |
| **Row 3** | **0** |
| **The response DID NOT earn a point for this row.** The response does not identify two difficulties and/or opportunities related to the program development process and how these programming issues were resolved or handled. The issues identified have to do with "what particular questions my partner and I would come up with" and "what genre our game would be based on." |