Name(s)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Period \_\_\_\_\_\_ Date \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |
| --- | --- | --- |
|  | **App Development Planning Guide** |  |

## 

## Project Description

For this project you will work with a partner. Together you will create an app that teaches your classmates about any topic you both find interesting. Along the way you’ll learn how to use many of the features of App Lab as well as skills that will help you when building more apps throughout this class.

**You will submit**

* Your final app
* This completed project-planning guide

**App Requirements**

* Uses at least three screens
* Includes examples of images, audio, and text
* A clear and easy to navigate user interface
* Clearly communicates information about your topic
* Code is cleanly written and free of errors

**Steps**

* Collaborate with your partner to pick a topic you are both interested in
* Interview classmates to identify what they already know about the topic
* Design your app’s user interface using this planning guide
* Design and program your app in App Lab
* Collect feedback from your classmates and update your app
* Share your final app with the class

## Investigate and Reflect Phase

**Step 1. Brainstorm Topic Ideas:** Your app can teach your classmates about any topic you and your partner agree on. Your topics could be a hobby, something you’ve always been interested in, a piece of your personal history, or just something you think your classmates should learn more about.

Write down three ideas for a topic that you brainstorm individually.

Idea 1: Talk about math in a trivia way.

Idea 2: humorous and funny

Idea 3: user input, audio links to the local file, loop, if statements(condition)

**Step 2. Choose One Topic:** Now talk through your ideas with your partner. Together pick a topic both of you are interested in teaching your classmates about. Explain in a few sentences what would be covered. For example, if your topic is basketball, you would write a few sentences explaining that you would cover the rules and the origin of the sport.

Our Topic:

Trivia mathematic problems, how to add up the geometric series, how to take integrals, how to do basic algebra.

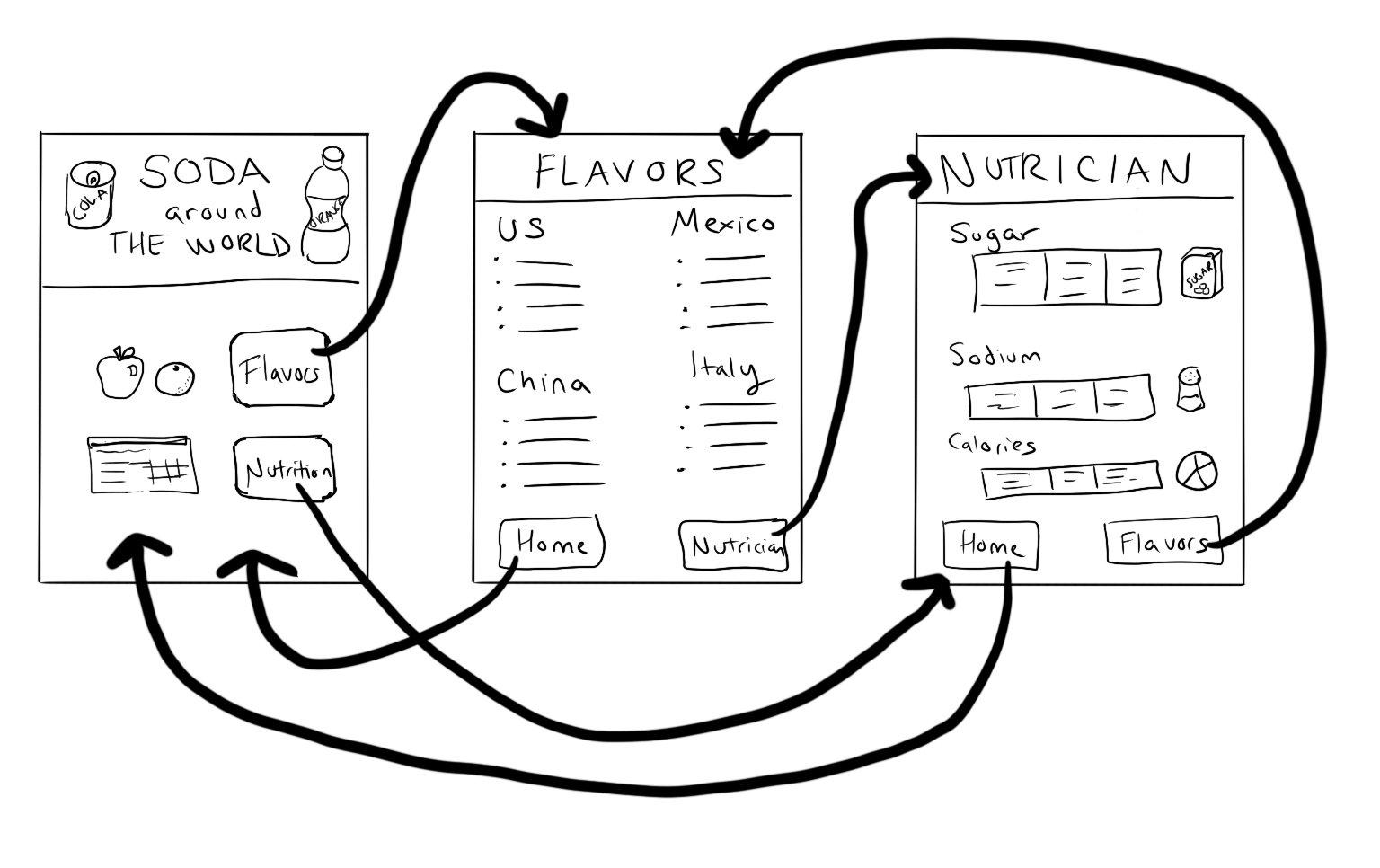
**Step 3. Interview Your Classmates:** To design your app you’ll need to understand your users. For this project your user is your classmates, and you’ll need to understand what they already know about your topic.

Find two classmates and talk to them about your topic for a couple minutes. Then fill in this table

|  |  |  |
| --- | --- | --- |
| **Name** | **What do they already know about your topic?** | **What do they need or want to learn about your topic?** |
| Jaymin | How to do basic algebra | More information that is calculus related. |
| Phili | How to sum up the geometric series | Some higher-level math. |

## Design Phase

**Step 4. Create a Program Specification:** Based on your research you identified **requirements** for what your app must teach your classmates. On the next page you should draw a **specification** that shows how your app will actually run to meet those requirements. This means you should include all the buttons, text, and images that the user will be able to use. Write notes or draw arrows showing how different user interface elements should work. For example, if clicking a button takes me to another screen, I should draw an arrow from that button to the drawing of the screen.



|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

## Building Phase

**Step 5. Start Building Your App:** Work with your partner to build your app. Along the way make sure you:

* Use the program specification you drew as a starting point, but it’s OK to update as you go.
* Use pair programming
* Use your debugging skills to check that your app is working

Before you begin to code, fill out the chart below for any Event Handlers in your program:

|  |  |  |
| --- | --- | --- |
| Element ID | Action | What happens? |
| *“dogButton”* | *“click”* | *A picture of a dog appears*  *The background of the screen changes to green* |
| Continue.screen1 | click | Go to next screen (Question 1), stops sounds |
| Continue.screen2 | click | Go to next screen (Question 2), stops sounds |
| 2x, 10, 4, 7 (. screen1) | click | Hide the yes, reveal the no, play the two sounds |
| Libido.screen2 | click | Hide no, reveal yes, play cena, reveal continue 2 |
| mrG.question2 | click | Brings you to geistfeld page, starts siren |
| GOBACK.geistfeld | Click | Goes back to question 2, stops sounds |
| Jaymin.question2 | Click | Brings you to jaymin page, starts bop music |
| Phili.question2 | Click | Brings you to phili page, starts AMOGUS song |
| Robin.question2 | click | Brings you to robin page, starts island boy |

Use the chart to guide you in adding programming statements to your program.

## Testing Phase

**Step 6. Testing & Feedback:** You will need to test your app to make sure it works as expected. To do that find at least two classmates to use your app.

* Ask students to read through your program specification and requirements
* Ask them to use the app and test out the different behaviors included in your specification
* Write down anything you noticed them finding confusing or broken
* Ask them to share anything they recommend improving

|  |  |  |
| --- | --- | --- |
| **Name** | **Things that could be improved based on watching them use the app** | **Improvements this person recommends** |
| Mr. Compton | Sometimes he doesn’t know the potential buttons that could be clicked | Instruct users to click a particular area on the screen. |
| Jaymin | Clicked the button too many times, and the audio repeats playing for many times | Make the audio play only once if the button is clicked. |

**Step 7. Pick Improvements:** Pick at least one improvement you plan to make to your app based on feedback you collected from your classmate.

Improvement 1: make the screen less crowded, and the increase the contrast between the text and background.

## 

## Reflection

**Question 1:** Provide a written response that:

* describes the overall purpose of the program
* describes the functionality of your app
* describes the input and outputs of your app

(Approx 150 words)

CAP

**The overall purpose is to entertainment of users, and to allow the authors to learn more about the way Code.Org has us program in things. Some of the things that needed to be done was sound editing, file management, and the implementation of importing various pictures and sounds outside of the actual programming for the screens. The function is a simple joke quiz applet, like the old Flash game “The Impossible Test.” By being able to press buttons and even pictures in one case, you can allow the code to spit back results such as sounds and text, or even different endings. This interaction was the input output coding that took the majority of the work. All of these were put together with the coding by yours truly, and after some helpful advice from others in the classroom, including Mr. Compton, we were able to make an applet that brings a smile to the face. Note, a confused smile is still a smile.**

ROBIN

**The overall purpose of the app is to be as funny as possible, using pictures, audios and other methods to make the boring math funny. We took some of the hard problems and put them in the app. We used as a multiple choice format to make the users click among the options(input). However, they would soon find out that none of them are correct(based on the output). And it turns out that the picture was correct. Same with the second question. The different options construct different endings. After the feedback from Mr. Compton and other people in class, we made the background cleaner so the texts are easier to read.**

**Question 2:** This project was created using a development process that required you to incorporate the ideas of your partner and feedback from your classmates. Provide a written response that describes one part of your app that was improved through input from EITHER your partner or feedback you received from classmates. Include:

* Who specifically provided the idea or recommendation
* What their idea or recommendation was
* The specific change you made to your app’s user interface or functionality in response to the recommendation
* How you believe this change improved your app

(Approx 150 words)

CAP

**Mr. Compton was able to see that he could edit the text boxes, so I was able to change it so you cannot now. Some things that could have been improved was adding a hyperlink to the applet on the Phili screen, but I don’t know how to do that, even after looking it up and searching. Another thing that Robin suggested was to format the sound files and create easier to understand IDs as to allow the user to adjust the code or see what makes it tick. Though is is unfortunate I could not create the hyperlink, the advice from the class that spurred these changes made it much better. For example, Maryam kept highlighting some text, and couldn’t hit the buttons. This was fixed with Mr. C’s advice, which improved the user friendly aspect of the applet.**

**ROBIN**

**After the feedback from Mr. Compton and other people in class, we made the background cleaner, so the texts are easier to read. For example, there were green texts that blurred into the background, we switch it to yellow, so it is easier to see. And some sounds effects, too. Some of the sounds won’t stop if we go to another slide, after the inspection, we found out that we forgot the “pause” command, so after we added that, the app worked normal. And other times we have to link to a local file of the audio instead of the online hyperlink, which is potentially confusing at first.**

## 

## Rubric

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Category** | **Extensive Evidence** | **Convincing Evidence** | **Limited Evidence** | **No Evidence** |
| User Interface Screens | User interface includes at least three screens | User interface includes two screens. | User interface is on a single screen. | The screen is blank. |
| User Interface Navigation | The user can easily navigate between all screens. | The user can easily navigate between most screens. | The user can easily navigate between some screens. | The user cannot navigate between screens. |
| User Interface Elements | The app includes at least one example each of:   * Text * Image * and audio. | The app includes at least one example of two of the following:   * Text * Image * And audio | The app includes at least one example of one of the following:   * Text * Image * And audio | The app includes no text, images, or audio. |
| Code | Code runs without errors | Code runs with a few errors. | Code does not run or has a lot of errors. | Code is blank. |
| Element IDs | Screen elements all use meaningful IDs. | Screen elements mostly use meaningful IDs | Some screen elements use meaningful IDs | Screen elements do not use meaningful IDs. |
| App Topic | Topic is clearly communicated and explained. | Topic is somewhat clearly communicated and explained. | Topic is not communicated well. | App appears to be a random collection of elements with no clear topic. |
| App Development Planning Guide: | Planning guide is fully completed. | Planning guide is mostly completed. | Planning guide has a few parts completed.. | Planning guide is empty. |
| Written Response 1: | Response accurately describes the purpose, functionality, and inputs/outputs of the app. | Response mostly describes the purpose, functionality, and inputs/outputs of the app. | Response is not complete, but does describe the purpose, functionality, or inputs/outputs of the app. | Response does not address the prompt in any way or is blank. |
| Written Response 2:. | Response fully describes an idea or recommendation provided by a partner / peer and how it improved the app. | Response mostly describes an idea or recommendation provided by a partner / peer and how it improved the app. | Response is not complete, but does describe some of the work with a partner. | Response does not address the prompt in any way or is blank. |