5.2 MULTI-SCREEN APPS

Unit 5
Building Apps

5.1 RECAP

- Learned how to add a button, image, and text to an app.
- Learned onEvent handlers for various UI objects
- Use the console.log to help debug your code
- Saw the difference between sequential and event-driven programming
- Today's lesson: expand and improve upon your Chaser Game v.1 with new capabilities.

IMPORTANCE OF PREDICTIONS

- Making predictions is an important part of the learning process.
- Prediction Process Explained:
  - It's a good habit to take time to examine the code and predict what will happen BEFORE immediately hitting "Run".
- The purpose of prediction is that you are more likely to remember the result of some action if you try to predict what will happen first. In fact, you might remember better if your prediction is wrong.
- Professional programmers do this implicitly out of habit - it's a great way to quickly gain insight into your code and what it is (or isn't) doing.

ACTIVITY

- Go to Stage 2 on Code Studio:

https://studio.code.org/s/csp5/stage/2/puzzle/2

- Complete Puzzles 2-15
  - For the prediction portions—follow the steps outlined in the last screen
- Homework: Submit Chaser Game v2

CONSOLE.LOG SUMMARY

- Every programming language has some way of displaying simple plain text output

- Printing to the console is a very common technique not only for debugging but also as an aid in program construction and development.

- In other words, you can use console.log to send messages to yourself to verify the program is doing what you think it's doing which helps prevent errors down the line.

onEvent("screen", "CLICK",...

- BE CAREFUL when choosing an onEvent with a screen because every action done in the app will be done "on the screen".
- If you want to avoid this issue, you should make a button that covers the screen.
- Buttons & Images can be layered, but the screen will always be clicked.

DISCUSSION PROMPT

- Today was one of the first time many of you saw error messages in your programs requiring you to debug your code.

- Is it “bad” to generate an error message?
- Under what circumstances would a programmer actually “like” to get an error message?

WRAP-UP

- Even expert programmers make errors, so debugging is a critical step of writing any program.

- We'd much prefer the computer to catch errors for us than on our own. Error messages usually include helpful information about how you can fix your code.

- This process of finding and fixing errors in your code is entirely normal and is just as important a skill as writing the code in the first place.

COMMON MISCONCEPTION

A program that does not generate any error messages has run as expected.

COMMON MISCONCEPTION

A program that does not generate any error messages has run as expected.

Not all errors will generate an error message because sometimes we write functional code that does something different than we want. In order to catch these logical errors, we'll need to understand how our code is supposed to run and then test it to make sure that it does.