Week 4: Mapping the Flow

Getting with the Flow

At this point in our term together, we are just beginning to learn about coding and the JavaScript language. We are starting to see syntax – the right way to ask for things in code – as well as all sorts of instructions like conditionals and loops. On top of all of that, we started looking at grouping blocks of code statements together into functions and ways data/content can be organized in JavaScript (with object values being the most complex and often used.)

We have been given the tools to code, and now we need the mindset and practices necessary to create code in for a game system. We need to answer these kinds of questions:

- Why would I define a function?
- How do I plan the code I need to accomplish my goals?

The practice of programming is one part compartmentalization, one part screenwriting, and one part inspiration. Let me expand on these analogies:

- **Compartmentalization**: Programmers "factor" their code by planning a structure of useful blocks of code organized into re-usable chunks. A function is such a chunk. A complete application can (and must) be factored into functional blocks of code that interact and rely on each other, as "software architecture."
- **Screenwriting**: When we write code to control a Web browser, you can imagine you are the master of a virtual stage play. The web page is your stage and the browser your theater. You control objects – actors – in the page via the Document Object Model discussed in this week's reading.
- **Inspiration**: Like poets and authors who use a human language to craft their creative works, programmers also must rely on inspiration to come up with the actual code they write. There are many ways to accomplish the same goal when using JavaScript to control a browser. Your inspiration, based on knowledge and exposure to common patterns of coding, gives you the ability to program.

Why Plan?

Creating an interactive experience is often a complex undertaking. Programmers have developed approaches to planning just like architects have processes in place to design and develop a building. If blueprints are one of architecture's visualization tools, flowcharts are an analogous tool for programmers.

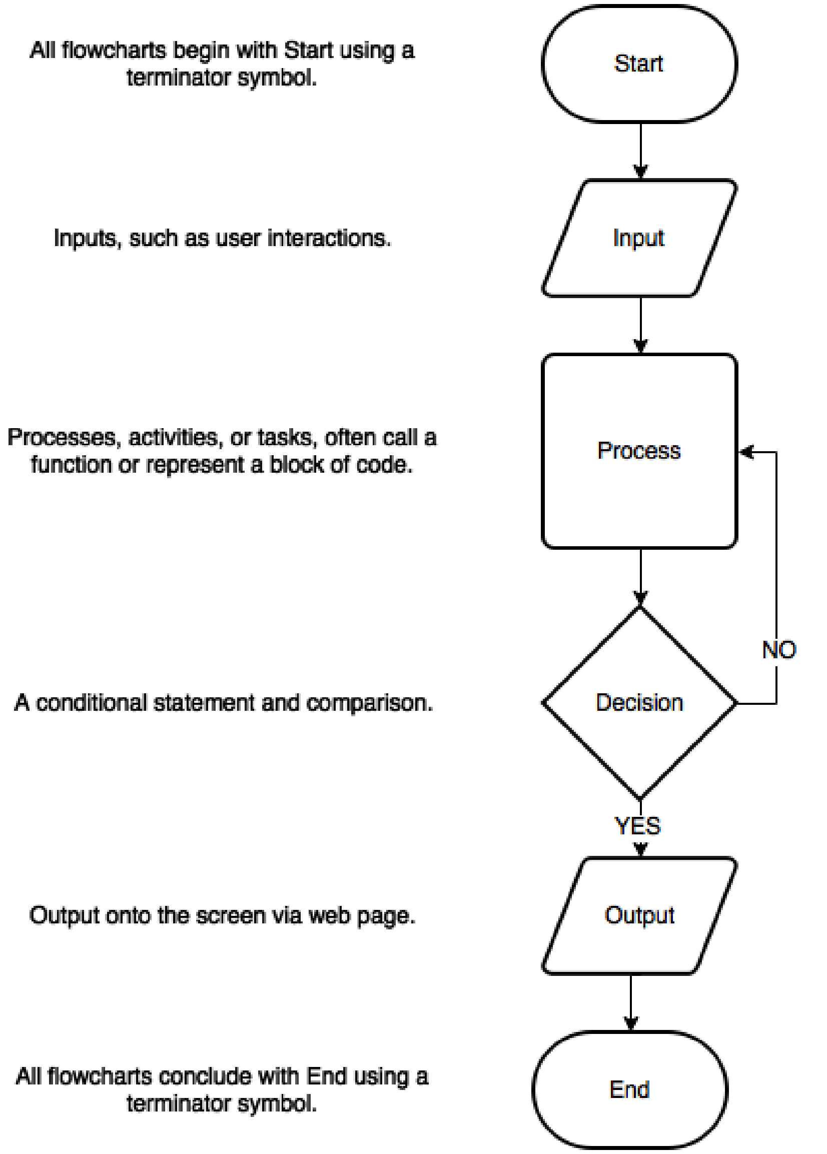
Flowcharts were a mainstay of procedural programming. With the rise of object oriented programming, the use of flowcharts for programming is no longer as commonplace as it once was.

However, even within the realm of object-oriented programming, complex logic can be visualized clearly using a flowchart. In interactives, diagramming the user's experience as they navigate through an app or online experience is an important prerequisite to designing the user interface.

Flowcharts still have their place in the world of programming, and are a great way to visualize how the various control structures work.

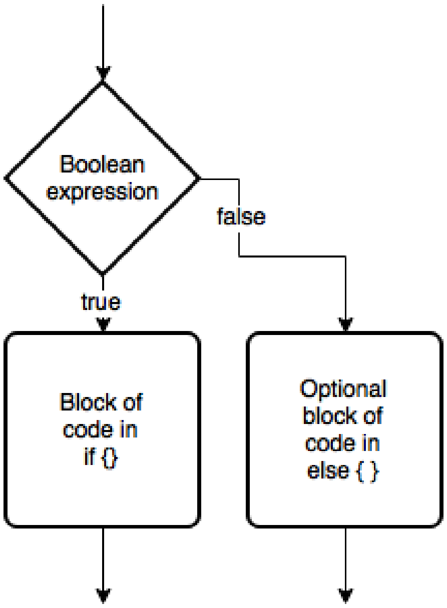
Basics of Flowchart Symbols

Each flowchart should start and end using a termination symbol (rounded rectangle), and all symbols should be connected with at least one line to another symbol. The chart below offers a simple and complete flow, with definitions for the symbols used.


All flowcharts begin with Start using a terminator symbol.
Start
Inputs, such as user interactions.
Input
Processes, activities, or tasks, often call a function or represent a block of code.
Process
NO
A conditional statement and comparison.
Decision
YES
Output onto the screen via web page.
Output
All flowcharts conclude with End using a terminator symbol.
End

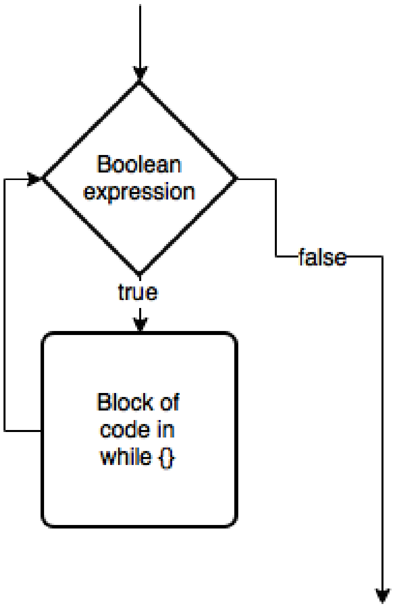
Flowchart for If...Else Conditional Statement

Every conditional statement can be modeled in a flowchart as a decision that determines which of two possible processes are followed.



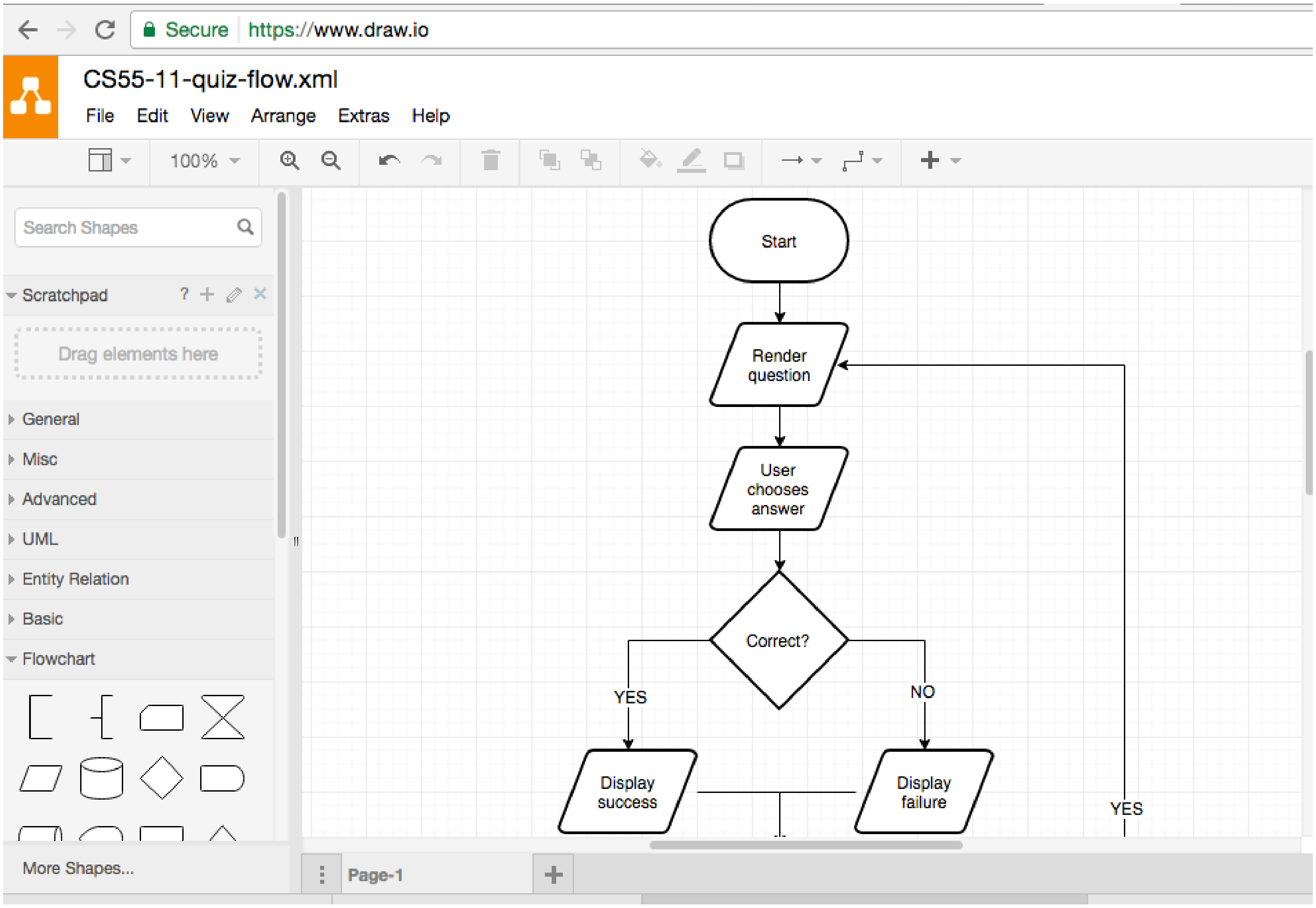
Flowchart for While Loop

Every loop can be modeled as a decision (whether to proceed or continue) and a process that repeatedly happens.



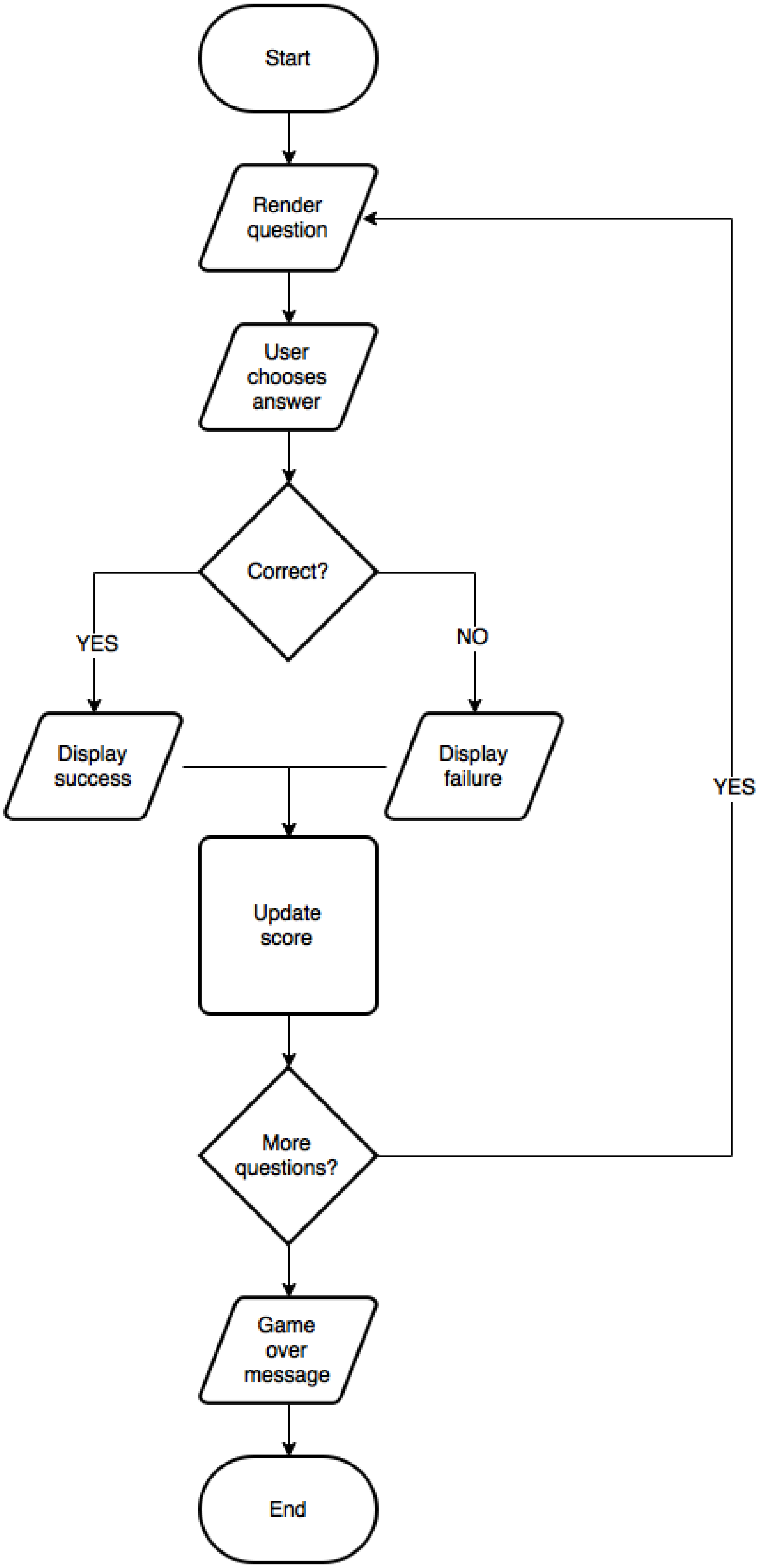
Flowcharting Tools

There are many drawing applications that can be used to create flowcharts. Dedicated applications designed specifically for flowcharting offer some efficiencies and purpose-built tools. If you aren't familiar with a drawing tool already, I recommend trying the free browser-based diagramming tool demonstrated in this week's Screencast, **draw.io**, found online at https://www.draw.io/ (https://www.draw.io/) .



Example Flow of Quiz App

To conclude this portion of the module, I have provided below an example flowchart of one possible solution to this week's assignment. You will be asked to create a flowchart and provide an upload of your chart with this week's assignment. Remember that each flowchart should start and end using a termination symbol (rounded rectangle), and all symbols should be connected with at least one line to another symbol.


Start
Render question
User chooses answer
Correct?
YES
NO
Display success
Display failure
Update score
More questions?
YES
Game over message
End