Hangman & Process

Let’s talk about Process

* You will continuously evolve how you plan and build a web app
* Let’s start with some options
* We’ll pick one to start with, see if it’s good or not, modify as needed

**Making sure we understand the game elements**

Play the game on paper

Write down explicitly anything that’s needed for the game

* Easily we think
  + Hangman gallows
  + Underscrores for where the letters will be,
    - Number of underscores to match letters
* The maybe not so obvious, but super-important for coding, the usually not seen
  + What the word is that that game is centered around
* The things that make it better
  + The letters we have guessed

All that’s left to do is to play the game on paper & see if we missed anything

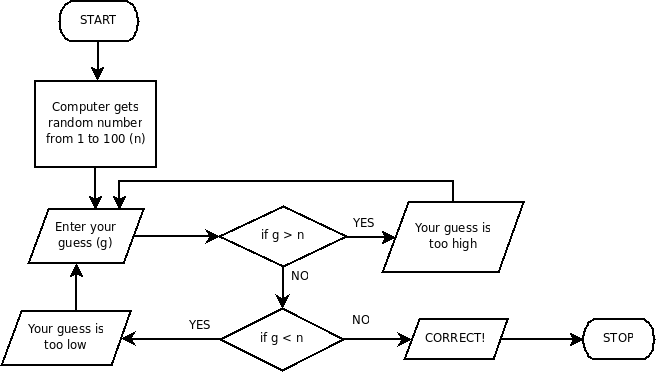
**Next Bit of moving towards code**

Flowchart – simple, we’ll refine

No, this flowchart is not for Hangman building that is something we should scheme

Oval for start and end, diamond for decision, rectangle for logic, parallelogram for UI

Start



Pseudocode-Flowchart

* Meaning let’s write some pseudo code based on the flowchart
* Once you do a first cut, then organize pseudocode into blocks of
  + Variables
  + Logic
  + UI elements
  + In the future we might incorporate objects

Pseudocode to real code – in stages

Start Simple

* Outline (outline will evolve as we do things like object oriented, or Patterns, or React’s version of patterns)
  + Variable Declaration with variables set to real values
  + Logic
    - Consider these cases
      * You guess “z” when The word is “top”
        + When the you still have guesses left
        + When that was your last guess
      * You guess “z” when The word is “zoo”
        + When you haven’t guessed “o”
        + When you have already filled in the “oo”
      * You guess “z” when The word is “zzz”
      * You guess “z” when the word is “zoo” and you already guessed “z”
* UI
  + Input (code after all the
  + Output (maybe code as console.log first)

**Coding Strategy to keep you sane (like bulleted lists apparently keep me sane)**

* Your goal is just to rough out the application
* You’ll refine it in stages
* Get the basic application roughed out
  + Variables with hard-coded values – really explicit variable names
  + Comments to make blocks of code understandable
  + Really visible logic so you can see what’s working, what’s not
  + Really simple UI – probably just output, maybe in console.log or easy screen output

Go Simple, everyone’s attracted to what they like to play with and you can noodle with the UI until time’s up, so save that for last

Next stage of refinement – lots of choices

* + Upgrade UI
  + Make sure when game is over we disable the submit button, change it to play again

A quick aside – and interview question

Why does the UI work?

Mmm, http