The halting problem – we don’t know when the program ever stops. Search later

Terminology

* Program Requirements (in the form of user stories)
  + Statements that define and quantify what the program needs to do
  + The work requirements isn’t used in the same way as elsewhere
  + Software requirements tend to be negotiable
* Functional Requirements (Do you have the functionality or not)
  + What a programs needs to do
  + Tend to be yes/np
* Non-functional requirements (may just be as important as the functional requirements, but don’t add to the user visible functionality)
  + The manner in which the functional requirements need to be achieved
  + Performance, usability, maintainability
  + Tend to be on a scale
* Design Constraints
  + Statements that constrain the ways in which software can be designed and implemented
  + Platform, language, DB, webapp, GUI, etc.

We will need a maze by the end of the next sprint

* Find an algorithm that helps us draw mazes