Presentation Plan

* Title slide
  + Project title: **Implementing Visualisations of Conway’s Game of Life**
  + Name: **Robbie Jakob-Whitworth**
* Outline slide
  + Definition
    - What is Conway’s Game of Life?
  + Problem
    - No good visualisations
  + Research Question
    - ...
  + Deliverables
    - Basic
    - Intermediate
    - Advanced
  + Progress so far
  + Analysis and evaluation
  + Conclusion
* Definition
  + What is Conway’s Game of Life?
  + It’s a zero-player game
    - Initial state determines evolution with no further input required
  + 2-dimensional infinite (or not?) grid of cells, each cell can be alive or dead
  + Set out the **rules**
    - Survival (two or three neighboroughing live cells survives for next iteration)
    - Death (four or more alive neighbours => death) (one or zero live neighbours => death)
    - Birth (exactly three live neighbours => becomes alive)
  + Example (with gif?)
* Problem
  + There are no good existing visualisations
  + List a few from Zotero
  + There are some existing implementations but they’re old / use outdated tech / don’t have a user-friendly interface
  + ...build up to ‘performant’ / define ‘good’
* Research Question
  + **What is the most performant technique for visualising Conway’s Game of Life in a web browser?**
* Deliverables
  + Let’s compare a few different implementation techniques:
    - Basic
      * <div>
        + Just lots of plain boxes on the page (100x100 = 10,000)
      * React
        + Advanced diffing algorithms, only re-renders relevant areas
      * Canvas
        + Designed for graphics work
    - Intermediate
      * Client/server architecture
        + Offload the state computation to the server
      * Send whole state or just the changes?
      * AJAX vs WebSockets
        + WebSockets are designed for two-way communication
    - Advanced
      * Intelligent computation, use hashing to avoid recomputing periodic regions
      * **Outside the scope of this talk**
* So far
  + Spent time researching existing implementations
  + Built out basic <div> implementation
    - Show gif
  + Built out most of reusable interface
    - Drawing / erasing mode
  + Diagram / flowchart of existing modular JS architecture
    - Static webpage
      * index.html
    - JS modules compiled together using Webpack
      * Render
      * Computation
      * Interface
    - Styles (SASS)
      * Grid
      * Interface
    - Packaged up into
      * app.js
      * style.css
      * index.html
* Analysis
  + Increase size of grid until performance drops
  + Use profiling tools in browser to measure CPU load, memory usage, frame rate etc
* Conclusion
  + There aren’t many good implementations of Conway’s Game of Life to date
  + We define ‘good’ as XXX
  + We’ll implement Game of Life in a number of different ways, using different client-side techniques and server-side techniques
  + Next steps: implement in other ways, e.g. React and canvas, try doing server-side computation
* Bibliography