Virtual Turing Tumble

Luke Gall - 2298070g

December 17, 2020

1 Status report

1.1 Proposal

1.1.1 Motivation

Turing Tumble is a board game that uses marbles and small components to simulate the inner workings of a computer. This board game, with the aid of puzzles, can be used to help teach children and adults alike the simple yet powerful building blocks of computers while helping them develop their fundamental logic skills, which are vital in learning how to code. While the Turing Tumble board game is a fantastic tool for teaching kids how to code, it isn't always easily accessible for a large classroom, with costs for the school, therefore an online version of the game would allow classrooms with access to computer labs the ability to learn using the Turing Tumble through a fun and interactive medium.

1.1.2 Aims

This individual project will aim to produce an online web application that replicates the functionality of a Turing Tumble board game. A user should be able to access a plain board and place the 6 different Turing Tumble pieces onto the board and trigger a marble to fall down the board, following the game logic of the physical board game. After a marble is collected at the bottom of the screen it should trigger another marble at the top. All pieces placeable by the user should have identical logic to the physical pieces. A user should be able to a create puzzle with their own title, description, difficulty, starting set up, and correct results. A user should also be able to view and play a set of 'Original' puzzles that are from the physical game puzzle book plus have the ability to view puzzles created by other users of the site. The board should show the position of the current marble in play so that players can follow the flow of the marble going down the game board.

1.2 Progress

- Web development language and JS framework chosen for development. **Angular** will be used which uses **Typescript**, **HTML** and **SCSS**.

- UI framework chosen. **Angular Material** will be used to style and improve UI elements.

- Research into serious / educational games and the positive impacts on learning for school pupils conducted including a literature review.

- Research into the Turing Tumble game and its puzzles to understand the logic of the game and the types of puzzles available.

- Turing Tumble board and pieces implemented using Angular component design that allows users to place components and simulate the board correctly.

- Users can pause / play,'step forward' and change the speed of the marble falling in the simulation, giving greater user control and better understanding of how the board works.

- SVG icons were made for all components with the rotation of the pieces changing when interacted with to give a better replication of the physical board game.

- Unit and end-to-end tests where created to ensure that the logic of the board was correct and that the major refactors performed were safe.

- **Create puzzle** page implemented where users can make a puzzle and upload it to a Firebase backend database.

- 29 default puzzles from the official Turing Tumble puzzle book were uploaded that users can play.

- Tutorial page was added to teach users how to use the site and the Turing Tumble simulation.

- A home page was created to give context to the site and give new users some direction.

- Cleanup of code and design of the application was completed with quality of life improvements added to various areas of the application.

1.3 Problems and risks

1.3.1 Problems

- Only had limited knowledge of Angular framework at start of development so some bad design choices were made in early stages until later fixed in major refactors with new knowledge and experience with the framework.

- Uploading object data to the firebase backend involved sending up a JSON string representation as Firebase would ignore 'null' values which were needed to represent some elements of a puzzle board. The object's type information was lost so data had to be converted to it's correct type when downloaded.

- The JS library that converts Objects into JSON doesn't allow Map objects to be parsed so the data had to be converted into an array for upload then re-converted into a Map object when downloaded.

- When getting connecting GearBit components to spin in a group, the class that changed their direction bypassed Angular view and lifecycle hierarchy so went against fundamental design decisions and give warnings, so new GearBit pieces need to be created whenever they interact with connected Gears and GearBits.

1.3.2 Risks

- Performing user evaluations for two groups in a non physical environment due to COVID. **Mitigation:** ensure that the application can be easily accessed remotely by being hosted online.

- Ensure clear and full documentation is created for the application. **Mitigation:** Update the wiki pages for each major file of the application over the semester, ensuring all application logic is explained in detail.

- Gathering relevant data that can be used to analyse how useful the application is for teaching school pupils (aged 16+) coding skills. I can only ask a user to use the app in a single session which would be difficult to capture the long term educational effects it could have. **Mitigation:** Give users as long as they would like with the application and provide opportunity to give compressive feedback on if they think it could help with their understanding of coding concepts in the long run.

1.4 Plan

- Week 1-2: Create and send out user surveys to two groups with link to online version of the application. **Deliverable: User surveys and raw data provided by participants**

- Week 3-4: Analyse data gathered from surveys and draw conclusions about application usability and usefulness in simulating a Turing Tumble and helping to develop coding skills. **Deliverable: Evaluation conclusions and graphs of any relevant data gathered.**

- Week 5-7: Initial draft of dissertation. **Deliverable: First draft submitted to supervisor**

- Week 8-10: Final writeup of dissertation. **Deliverable: Second draft submitted to supervisor, then submitting final version via Moodle**

1.5 Ethics and data

I have verified that the ethics checklist will apply to any evaluation I need to do. I will sign and complete the checklist.

To evaluate my application I will measure the ease of use and gather user feedback from a control group of my peers at the university and a group of school pupils (aged 16+) undertaking Higher Computing. They will be given time to play about with the application and undertake some of the puzzles then will be asked to fill out a user survey on the ease of use of the app, how easy it was to understand the Turing Tumble game through the tutorial and if they think puzzles could help them understand the core computing concepts that the application aims to develop.

No sensitive data will be held and all participants asked to carry out this survey will be 16 and over, the ethics checklist will be appropriate.