



Virtual Turing Tumble

4th year project - Luke Gall



Introduction

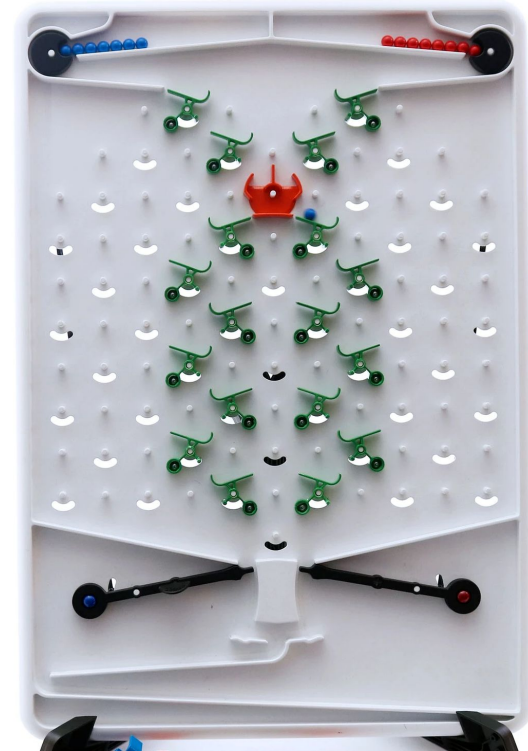
- Turing Tumble is a marble based game where users can create marble-powered mechanical computers to solve logical puzzles
- A fun educational game designed to teach users the small logical building blocks of computers

Project goals

- Create an accurate simulator that could be a replacement for the rather large and expensive physical version
- Allow users to build their own configurations using a graphical interface
- Allow users to play puzzles using the simulator
- Program should be easily to use and understand
- Users should be able to create their own Turing Tumble puzzles that others can play

Turing Tumble

- Marble dispensers top two corners
- Marbles travel down the board interacting with different pieces which change the path it travels
- At the bottom it is collected by the board and releases a new marble



Turing Tumble Pieces

- Ramp - directs marble left or right
- Crossover - Keeps marble going in its current direction
- Bit - send marble in opposite direction then change direction. Stores state
- Interceptor - capture marble and stop execution
- Gear and Gear Bit - acts as a bit but turns any connecting Gear Bits. Makes the game Turing Complete



(a) A Ramp piece



(b) A Crossover piece



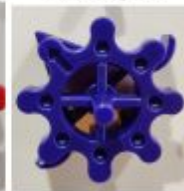
(c) A Bit piece



(d) An Interceptor piece



(e) A Gear piece


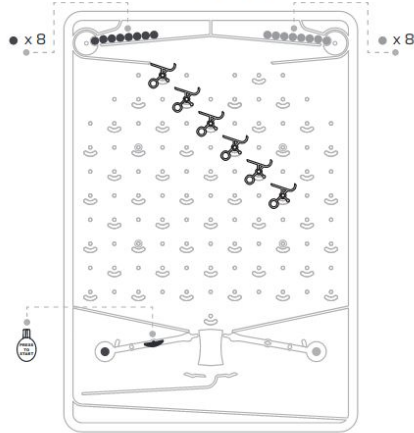



(f) A Gear Bit piece

Turing Tumble Puzzles

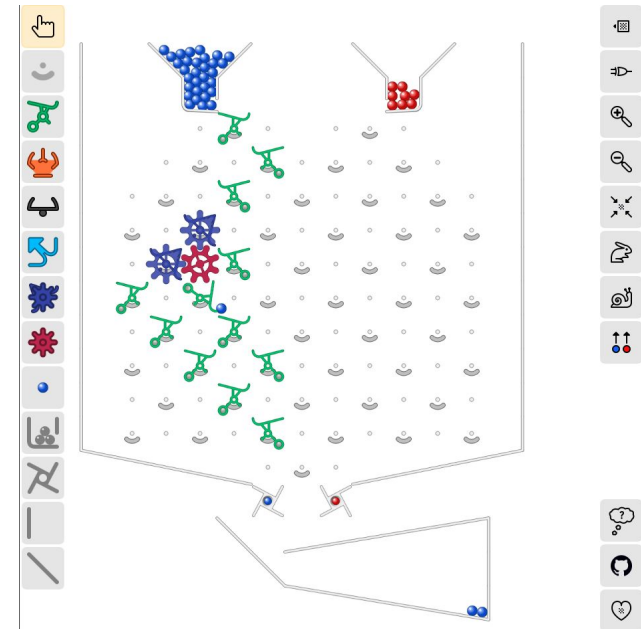
Puzzles in Turing Tumble are the main way to play the game and give a clear objective with constraints to obtain a certain marble output

- Users have a output they must aim to obtain
- The board has a starting setup which users can't change
- They only have a certain number of pieces to complete the output, they start off simple but can get complicated quickly

Challenge 2: Re-entry	
<p>Objective: Make all of the blue balls (and only the blue balls) reach the end.</p> <p>Required output:</p> 	
<p>Starting setup</p> 	<p>Available parts</p>  <p>x 5</p>

Existing Turing Tumble simulators

- JSTumble - JavaScript emulator running online
- Turing Tumble Simulator (Pictured) - Impressive online simulator with full gravity simulation
- Emtumble - A desktop based simulator that allows users to make custom board layouts
- All impressive but can't play puzzles within the program





Requirements

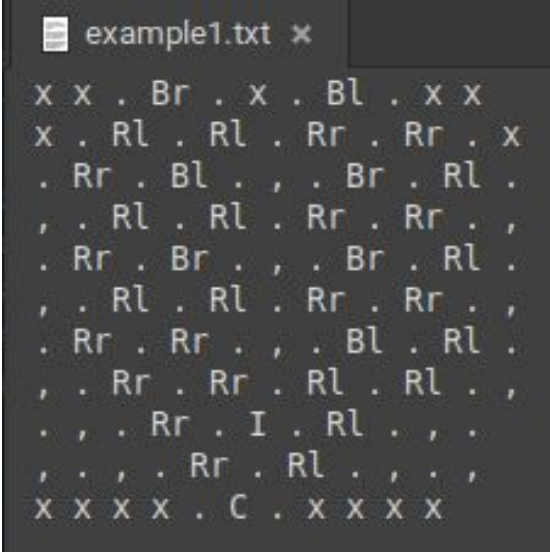
The program requirements were split into if they would match functionality found in existing simulators or add new features to distinguish this program. Some example requirements include

- Users should be able to make their own Turing Tumble configurations
- Add playback features to the simulator
- Allow users to play puzzles
- Allow users to create their own puzzles
- Allow users to filter puzzles based on difficulty

Design

Design decisions were explored before starting the implementation phase of the project including

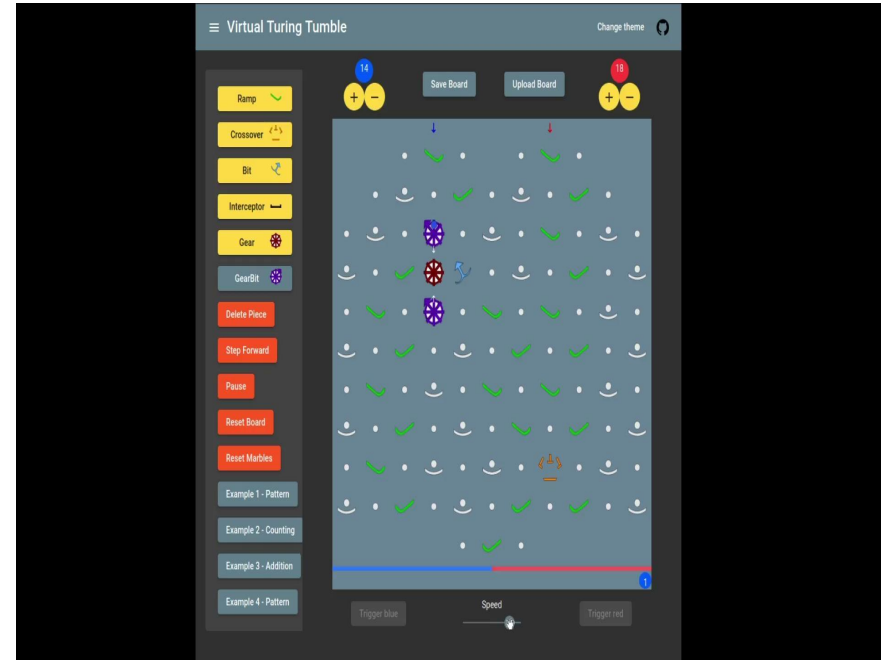
- Web application using the Angular framework
- Created wireframes for various pages
- Allow users to save board configurations as a txt file



```
example1.txt x
x x . Br . x . Bl . x x
x . Rl . Rl . Rr . Rr . x
. Rr . Bl . , . Br . Rl .
, . Rl . Rl . Rr . Rr . ,
. Rr . Br . , . Br . Rl .
, . Rl . Rl . Rr . Rr . ,
. Rr . Rr . , . Bl . Rl .
, . Rr . Rr . Rl . Rl . ,
. , . Rr . I . Rl . , .
, . , . Rr . Rl . , . ,
x x x x . C . x x x x
```

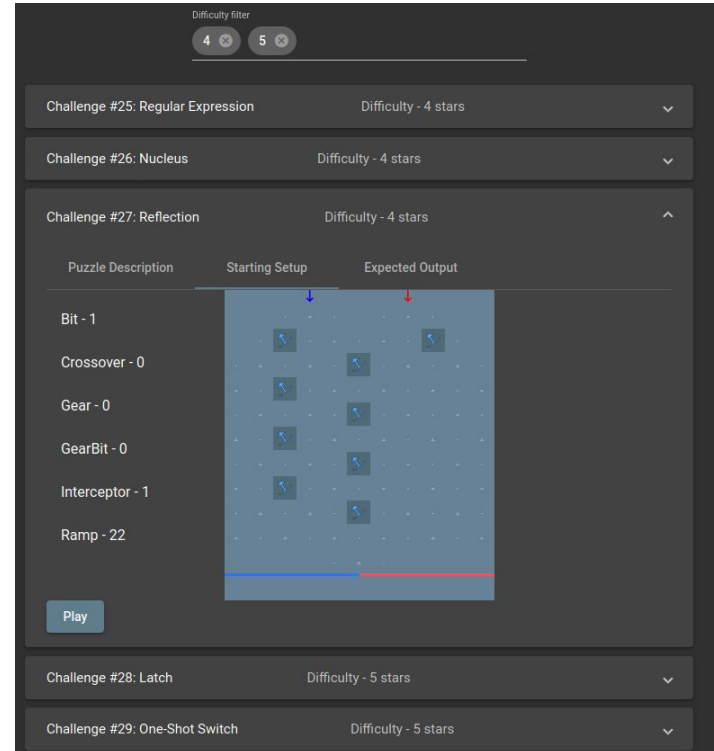

Implementation - Simulator Features

- Users can create configurations by placing and deleting pieces
- Users can simulate a Turing Tumble with marbles updating in real time
- Playback and board options were added
- Different example configurations can viewed to explore some more complicated Turing Tumble boards



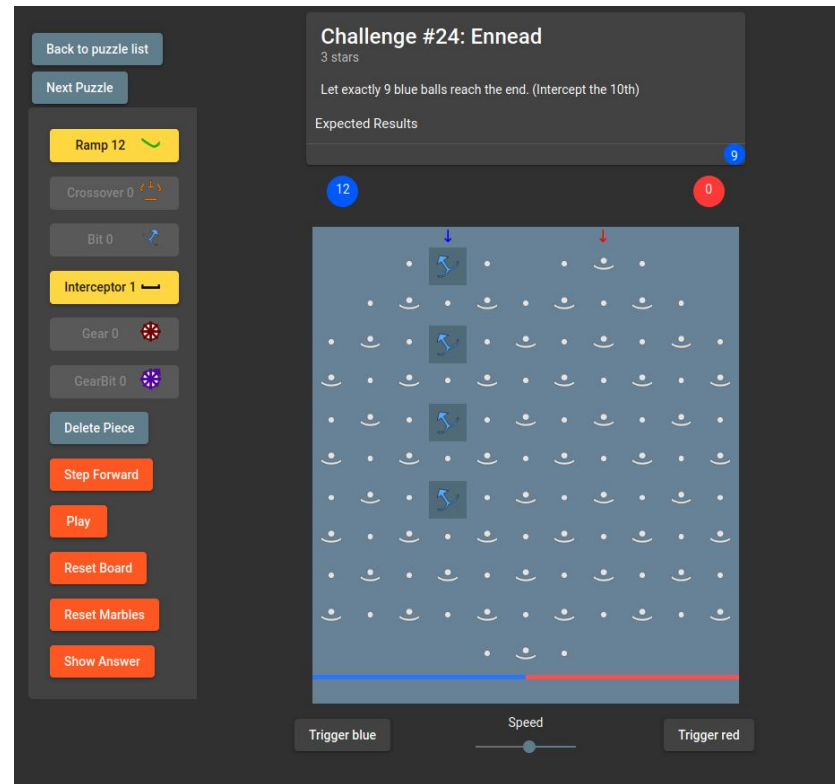
Implementation - Choosing Puzzles

- Users can select from the puzzles found within the Turing Tumble game guide or user created puzzles
- Users can filter the puzzles using a difficulty filter



Implementation - playing puzzles

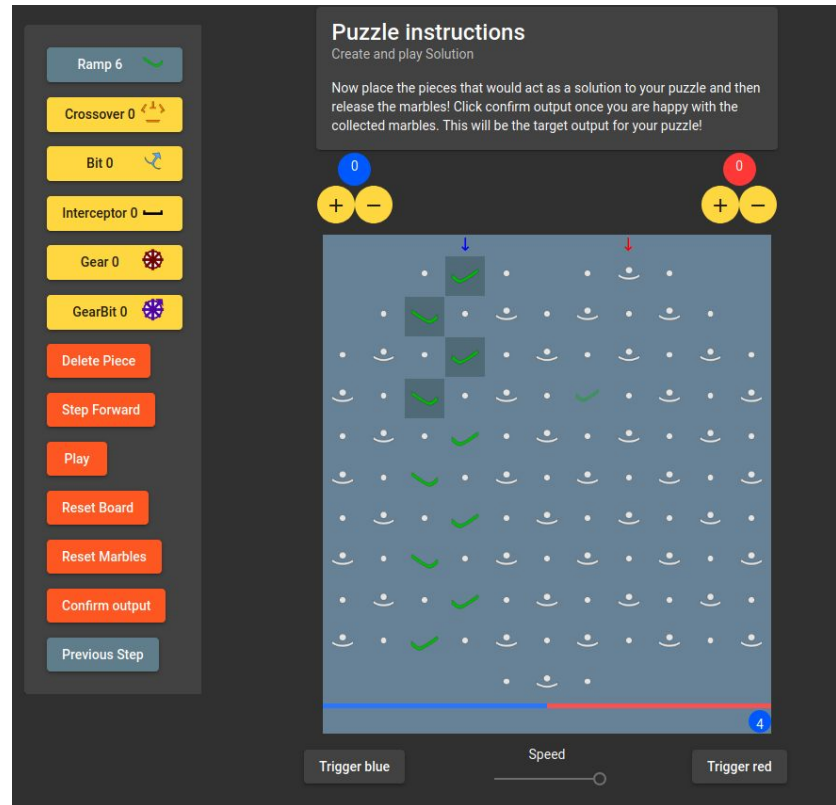
- Users can play through puzzles as they would in the physical game
- The rules present in Turing Tumble are enforced within the simulator
- Users can also move to the next puzzle, while staying within their filtered list



Implementation - Creating a puzzle

Users can create puzzles through a phase system that makes use of existing functionality

- The user creates the starting setup for a puzzle
- They then create the solution for the puzzle and play through it to get the required output
- Finally they input the title, description, and difficulty through a simple form



Implementation - Usability

- A home page was created to give users a starting point for the site.
- A detailed tutorial page was created to give details about using the program and playing Turing Tumble

Tutorial

General Instructions

Component information

Puzzles

This virtual turing tumble runs in the browser and there is nothing for you to install. Find the **sidenav** to the left of the screen to go to the different pages of the site. These include

- **Home** which gives a brief introduction to the site and why it was made.
- The **Plain Board** page will give a plain board which will allow you to play with a Turing Tumble simulation.
- The **Original Puzzles** page will give a list of puzzles to play which will be explained in a different tab.
- **Online Puzzles** will require you to log in to access and play other users created puzzles.
- Finally the **Create Puzzle** page will let you make your own puzzles for other users to play.

The site has two themes, light and dark, which can be changed on the fly using the button on the top header bar.

A Turing Tumble board involves blue and red marbles, shown at the top left and right corner of the board respectively. These marbles travel down the board interacting with various **pieces** which mimic parts of a computer circuit. There are 2 triggers at the bottom of the board which start the marble down the board. The marble can only travel down the board along routes of pieces, if not it will 'fall'. When a marble gets to the bottom of the board it will release another blue or red marble depending on the side the marble gets to. It will also be 'collected' in which the board will store the marbles that have reached the bottom. An example is shown below in which 2 blue marbles were collected, then 3 red and finally 5 blues.

5 3 2

The number of coloured marbles can be found in the top left and right of the board, the amount of marbles can be increased / decreased using the 'plus' or 'negative' buttons. The blue marbles drop in from the blue arrow at the top of the board while the red marbles drop in from the red arrow. In the board below you'll see different 'slots' on the board. Component Slots and Pins. Component slots allows all 6



Evaluation

Two sets of use evaluation was carried out, monitored and unmonitored. Users were asked to follow a task sheet that focused on various features of the program then to rate the ease of use and give any suggestions for the program.

- User feedback was positive with but one average response giving high ease of use
- Different strong suggestions were given including an interactive tutorial, different filter types, and a clearer representation when multiple Gear Bits are connected.
- Unit tests were created to test underlying logic of the program
- Automated User Interface tests were created that tested various UI logic



Conclusion

- Project created a Turing Tumble simulator
- User evaluations were carried out to evaluate ease of use of the program
- Different areas of future work include adding more features or polishing the existing implementation