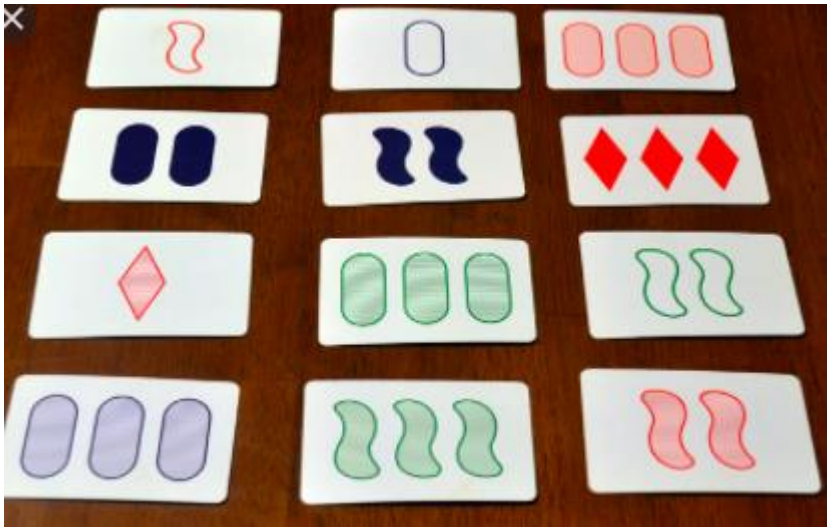


CSCU9N5 Assignment: Design report document | student #: 2718224

SET is a card game about finding patterns in cards with 4 variable graphic features.
The aim is to find sets of 3 cards, where they all have the same or different qualities in each variable.
The variables are:

- shape (oval, diamond, squiggle)
- colour (red, green, blue)
- shading (solid, striped, open)
- number (one, two, three)

The deck of 81 cards contains 1080 unique sets.

The game is played by laying 12 cards down face down and then turning them over.
The first player to call 'set' points out their set of 3 cards, and if it's agreed to be correct, keeps the cards. If they fail to point out a valid set, a card is removed from them (if they have any). The cards are kept to a minimum of 12 on the table, and if no sets are found, 3 more cards are added until a set is found. The player with the most cards in the end wins.
https://en.wikipedia.org/wiki/Set_(card_game)

Design goals

The aim of the project is to switch users on to algorithmic thinking by means of an interactive multimedia card game, demonstrating the power of programming over unassisted human brain power in a relatively simple, yet confounding pattern finding / logical reasoning task as the one in the game of Set. Time comparisons can be used to see how the player measures up to the code running in the browser.
A tutorial shall explain the rules of the game, showing example sets.
The different possible kinds of algorithms for finding sets can be demonstrated interactively and explained in a way that doesn't require programming experience.

The project was inspired by being *terrible* at the game.

Who is it for?

The program is mainly aimed at school-aged children, but also stands to benefit other user demographics, who might like (card) games, or might be interested in learning about computational thinking and algorithms. It will also help people understand the rules of the game (they are sometimes difficult to explain to new players) and perhaps even get better at the game itself.
Due to design constraints, the focus is exclusively desktop browser platforms.

User personas

Micke is a 15-year old school student from Eskilstuna, Sweden. He has a budding interest towards computing through his hobby of PC gaming, which includes him having built his own computer, and towards programming by way of making modifications for games that he enjoys playing. He's heard about the SET game app online due to it becoming viral due to its demotivating comparisons of human performance in the game vs. the mere milliseconds it takes the code to find all available sets. People have been sharing their results as screen snippets online.
He is able to learn how the game works quickly, and finds the explanation of the workings of the set-finding algorithm interesting.
At home, his main method of internet browsing is on his desktop pc, on windows 10 and chrome browser.

Yana is a 20-year old university of applied sciences student living in Finland, by way of Russia. She's studying applied computing and found the SET game app mentioned on Reddit. She's curious about web design and computational thinking and has also played the game SET previously at a friend's place. She remembers feeling embarrassed at how slow she was to find the sets compared to friends who had played the game more frequently. She's been thinking about if there would be a 'recipe' for finding sets that she could use to become more proficient at the game.

Design details

general design direction

The app was developed with skeuomorphism in mind. A skeuomorph is a derivative object that retains ornamental design cues from structures that were inherent to the original. The game table and other viewports appear to physically protrude from the background. This makes the application stand out.

Use flow

A tutorial modal will explain the rules of the game, describing the aim and rules of the game, along with showing example sets. after this, the player can have a go at it. After cards are dealt out in a grid of 12, the player can start finding sets. After clicking on three cards, if they form a set, the time comparison is shown in the score board and points are added. The ”previous set” shown under the card grid is updated with the cards and an explanation of why it was a set. If they do not form a valid set, the cards will flash red denoting a failure to find sets, and a point is removed from the scores.

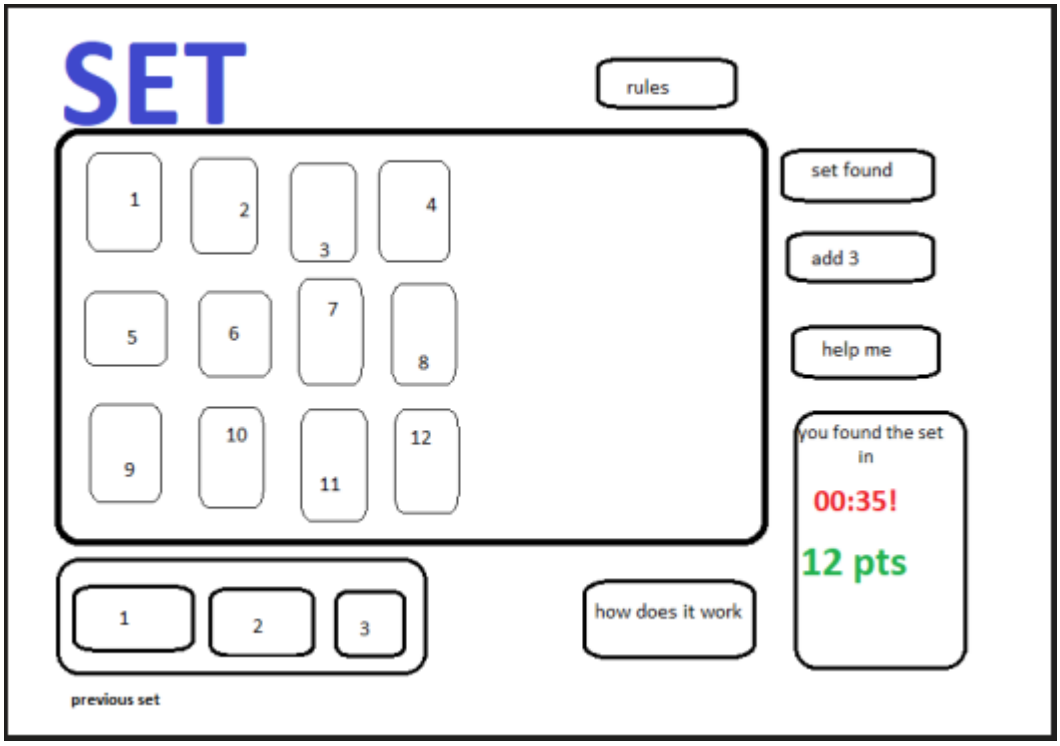
There is also a button that adds three more cards to the table, up to 15 cards on the table at a time. There is a help button, showing the player a possible set by highlighting the cards if they struggle to find one. If help is used, the player will not get points.

A button titled “how does it work?” will show an explanation of the set-finding algorithm, and some options for other ways of implementing it. This, along with the time comparisons for set-finding forms the educational aspect of the application.

Animations and sound

The game should have a satisfying look and feel by responsive animations and sounds to the players actions. Cards will shuffle on to the table from the deck. Cards will highlight when moused over, pop when selected, and as a correct set is chosen, the cards will animate to the scoreboard with an audible clink and an animation of the score tally showing that it increases by one from each card. In the prototype, this hasn't been implemented yet due to the game logic being unfinished. The animations will be implemented using *framer/motion*.

The entire app runs in a single screen without any reloading of the page. The *tutorial* and *'how does it work'* modals focus user attention into the content they provide by shading out the background.



[initial wireframe for the main screen]

Description of prototype contents

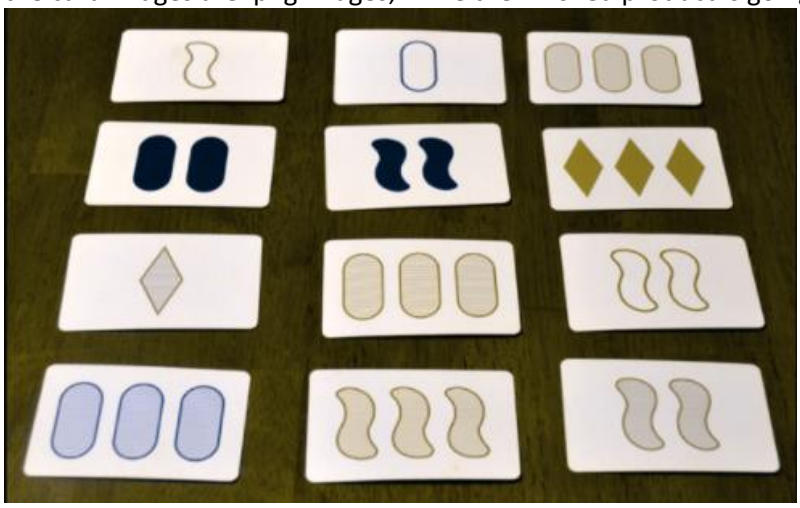
The prototype is a sketch of the finished product, showing the major design direction sufficiently where the game logic can be finished behind a near finished façade. Due to the nature of the game, animations and sound effects have not yet been implemented.

The prototype was built using ReactJS, and majority of the styling was implemented with simple CSS, save for a couple of components from react-bootstrap.
The card images were sourced online and split into individual .png files using GIMP 2.10.6. The image assets are stored in the `src` folder of the project.

Testing

Known accessibility issue: colour blindness

Due to the colour choices of the original game (red, green, blue), the game isn't well suited for people with various types of colour blindness (protanopia, deutanopia, tritanopia) to the result that the physical card game is unplayable for people with colour blindness.
To mitigate this in a digital version of the game alternate colour schemes can be provided with relative ease. This will be easy to implement once out of the prototype phase, as in the prototype the card images are .png images, while the finished product is going to have them modelled in SVG.



[the card colours with protanopia filter (https://www.toptal.com/designers/colorfilter/)]

User testing

User testing can be done for the game by giving a test subject tasks to perform and seeing if problems arise that need to be considered.
Some example tasks include:

- read the rules and find a set. can you determine how long it took you?
- try choosing an incorrect set. can you determine that it was wrong and why?

automated testing can be done using cypress, an end-to-end testing suite which allows tests for web content to be written to test through the UI as opposed to directly through the code. This way, more exhaustive testing for robustness of functionality can be done than with user testing.

Sources

ReactJS: https://reactjs.org/
react-bootstrap: https://react-bootstrap.github.io/
framer/motion: https://github.com/framer/motion
cypress: https://www.cypress.io/

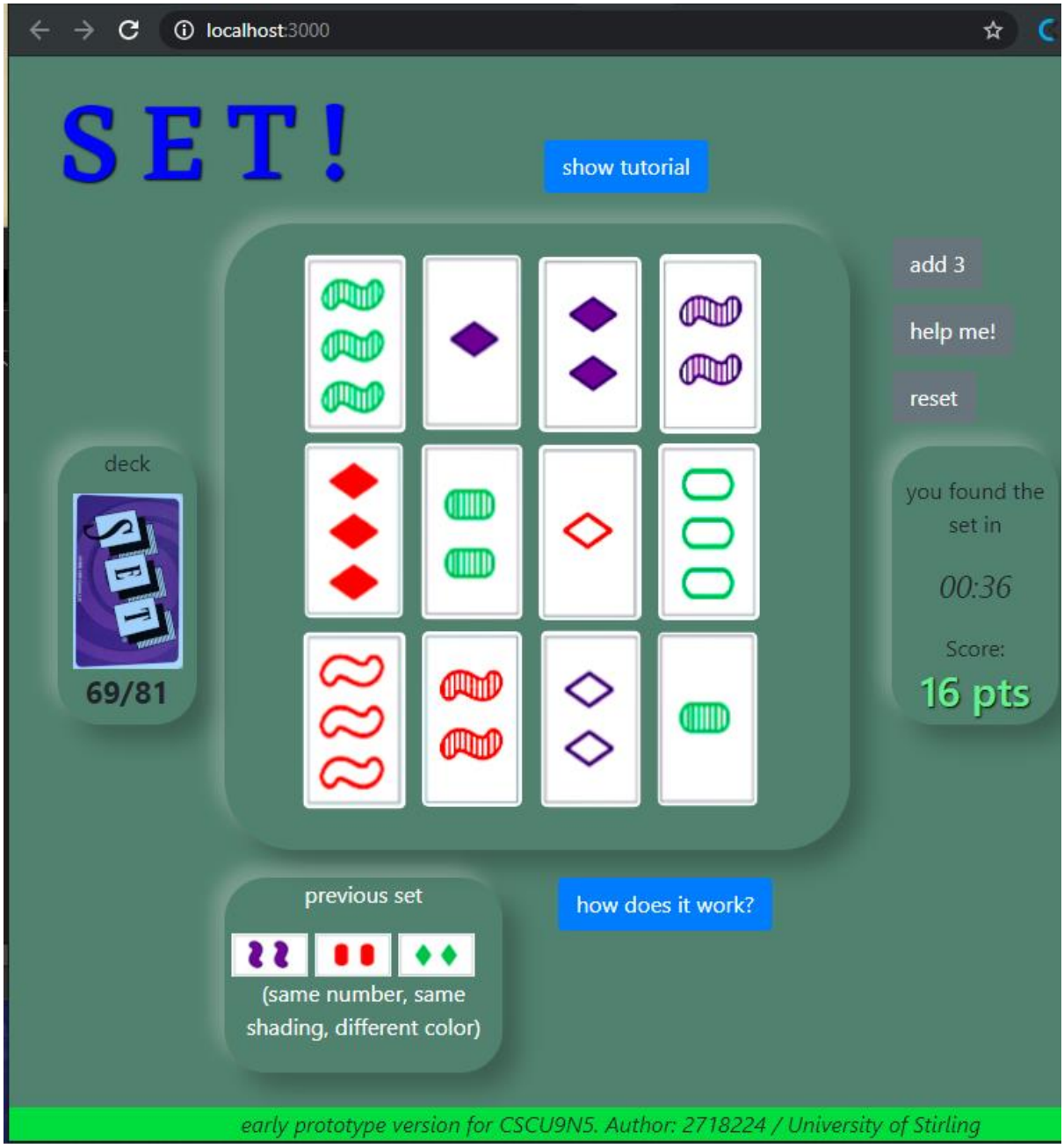
Installation of the prototype

The finished product will be published online either on a github webpage or on Heroku, requiring no installation, but the prototype needs certain actions to be viewed.

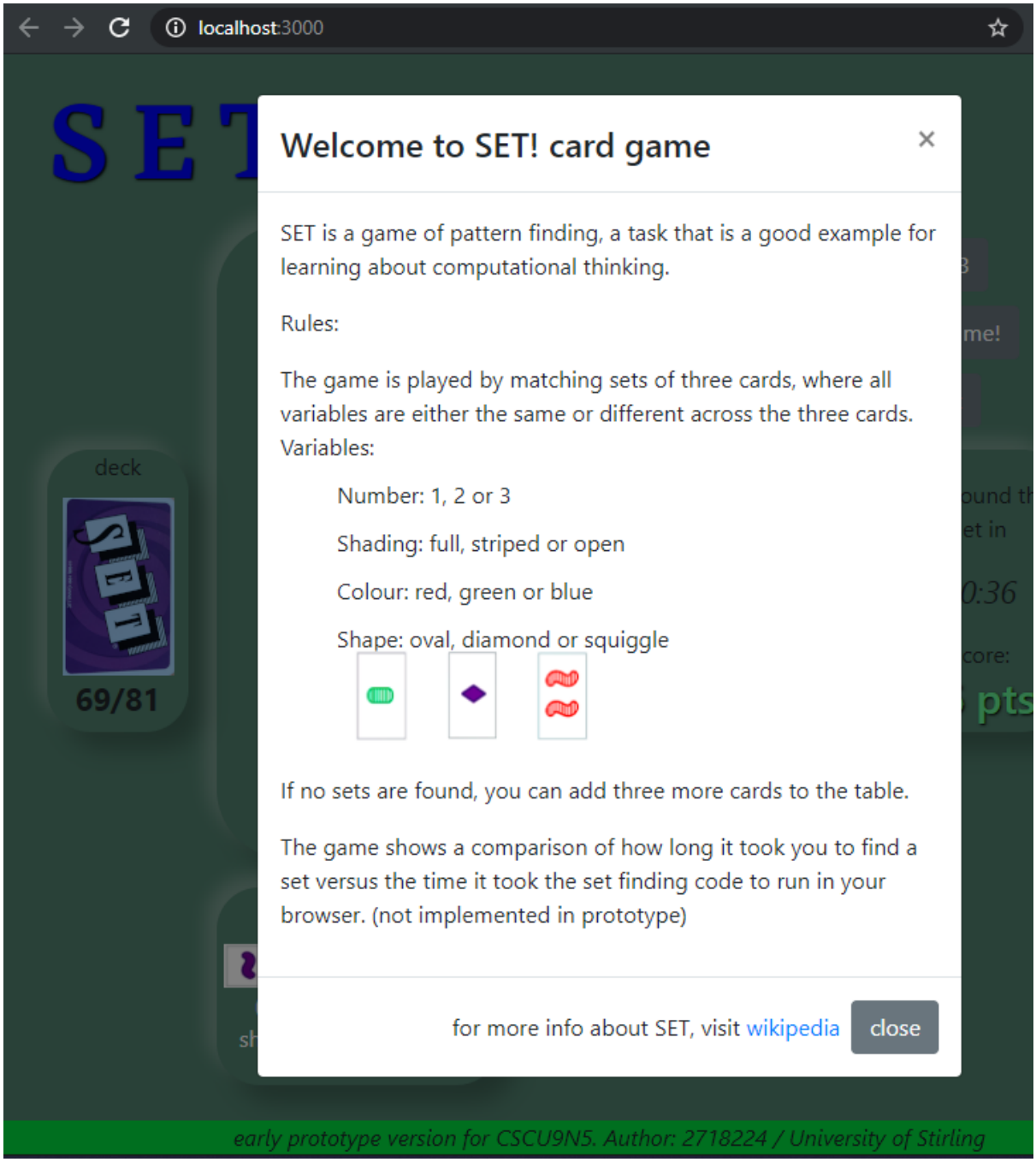
- Make sure you have Node package manager installed: https://www.npmjs.com/get-npm

1. in cmd, navigate to the folder `/setgame`
2. type `npm install` to install project dependencies
3. run the project with `npm start`
4. the application will open in a browser window

Screenshots



[main screen]



[tutorial screen]