

PDA - Project Unit

CodeClan Course Evidence

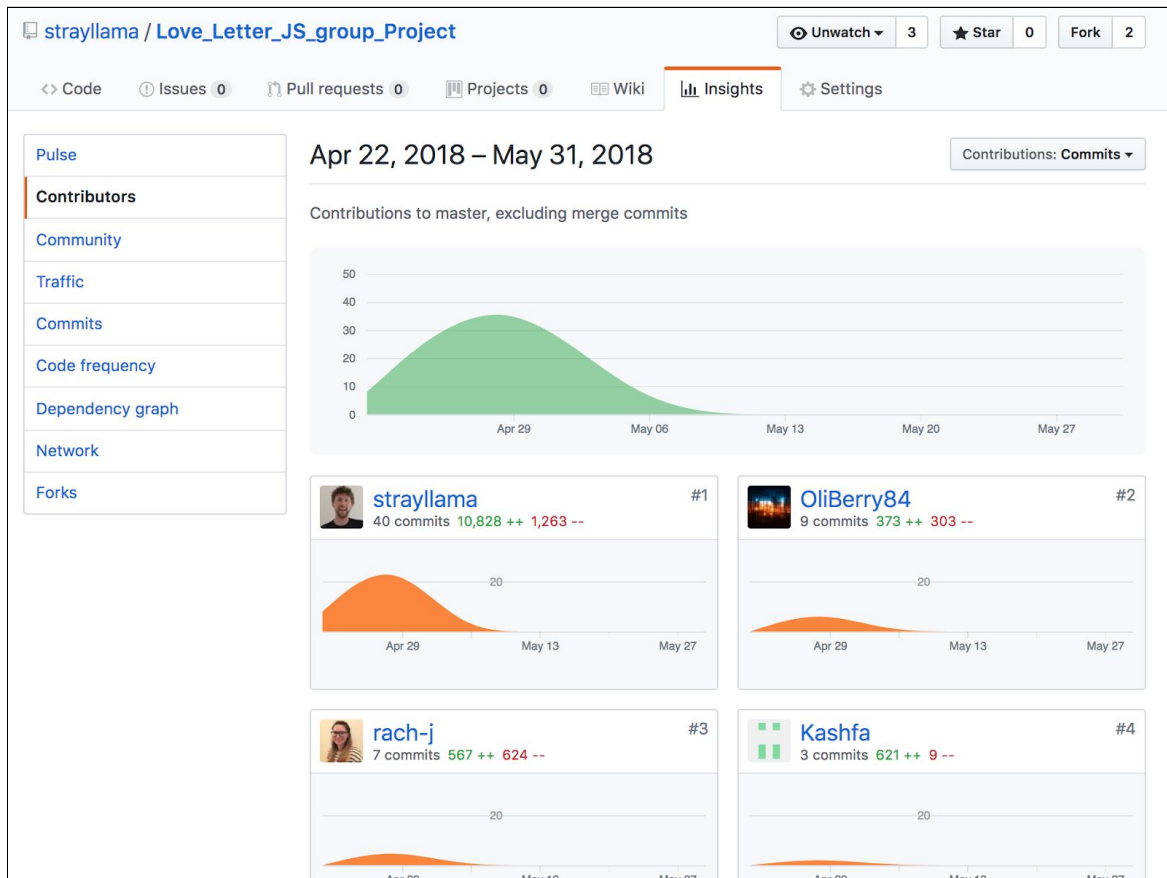
12.03.2018

Paul Clatworthy

2 Liberton Tower Lane, Edinburgh, EH16 6TQ

P 1

- Take a screenshot of the contributor's page on Github from your group project to show the team you worked with.



P 2

- Take a screenshot of the project brief from your group project.

JavaScript Project Specs

Situation

5 day Group project, after 3 weeks learning JavaScript. Paul, Oli, Rach, Kash. aka team PORK.

Browser Game

Create a browser game based on an existing card or dice game. Model the game logic and then display it in the browser for a user to interact with.

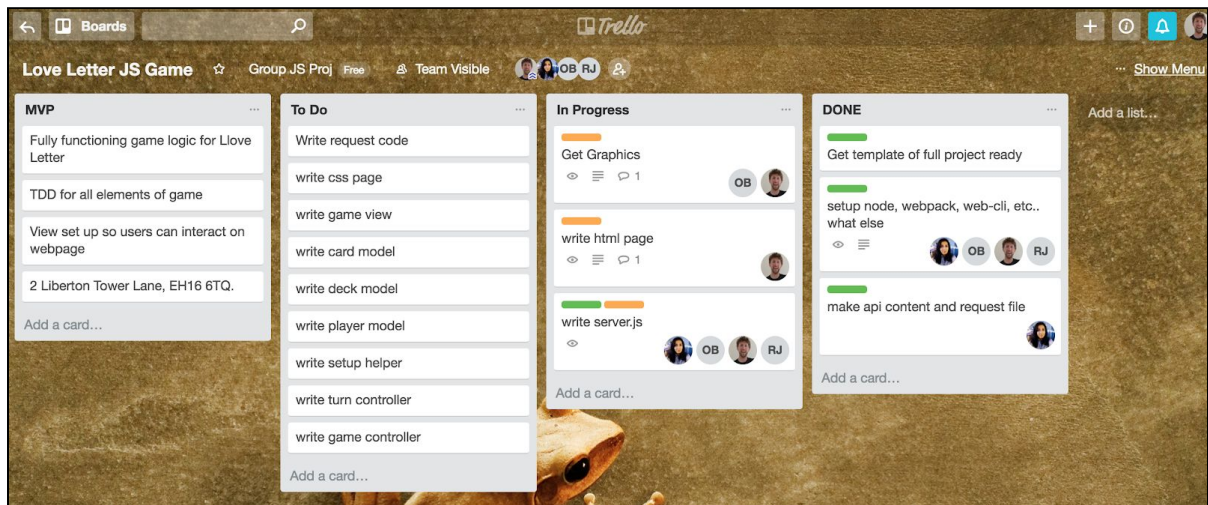
Make your own MVP with some specific goals to be achieved based on the game you choose to model.

Make 4 player text based game logic and function work, of Love Letter!

You might use persistence to keep track of the state of the game or track scores/wins. Other extended features will depend on the game you choose.

P 3

- Provide a screenshot of the planning you completed during your group project, e.g. Trello MOSCOW board



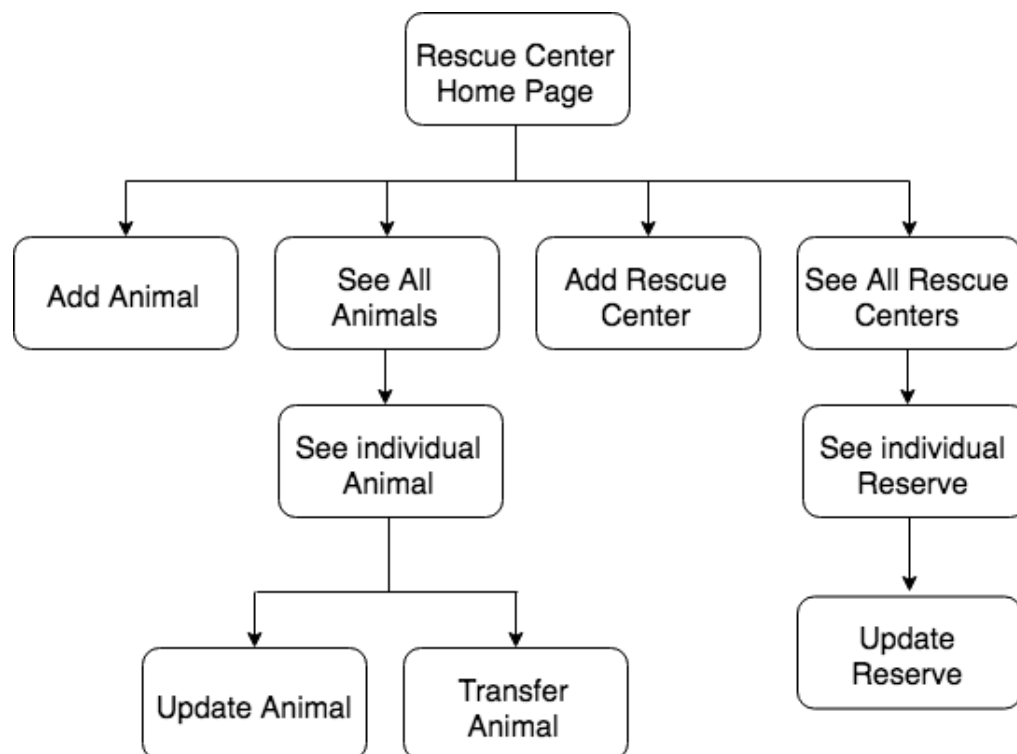
P 4 - An Acceptance Criteria and Test Plan.

Acceptance Criteria	Expected Result/Output	Pass/Fail
Clicking Start to initiate game	Text input boxes for names, change to labels with same contents. First players cards show	Pass
Running out of deck cards ends the game	When more than one player is alive and the next go can't be played to to empty deck, the winner should be announced and be the	Pass

	player with highest value card	
User can choose one of the two cards to play	The chosen card's action should be carried out. The other card is discarded	Pass
Played cards should be discarded	A discarded card should appear on the discard pile - even if its not been used but the result of the action of another player's card	Pass
Certain cards mean the player has no choice of which card to play	When Countess is in hand with king or prince, the king or prince must be the only card playable.	Pass
Player can win by killing all other players	If there is only one active player, they automatically win, regardless of card in hand and if the deck has cards left or not	Pass

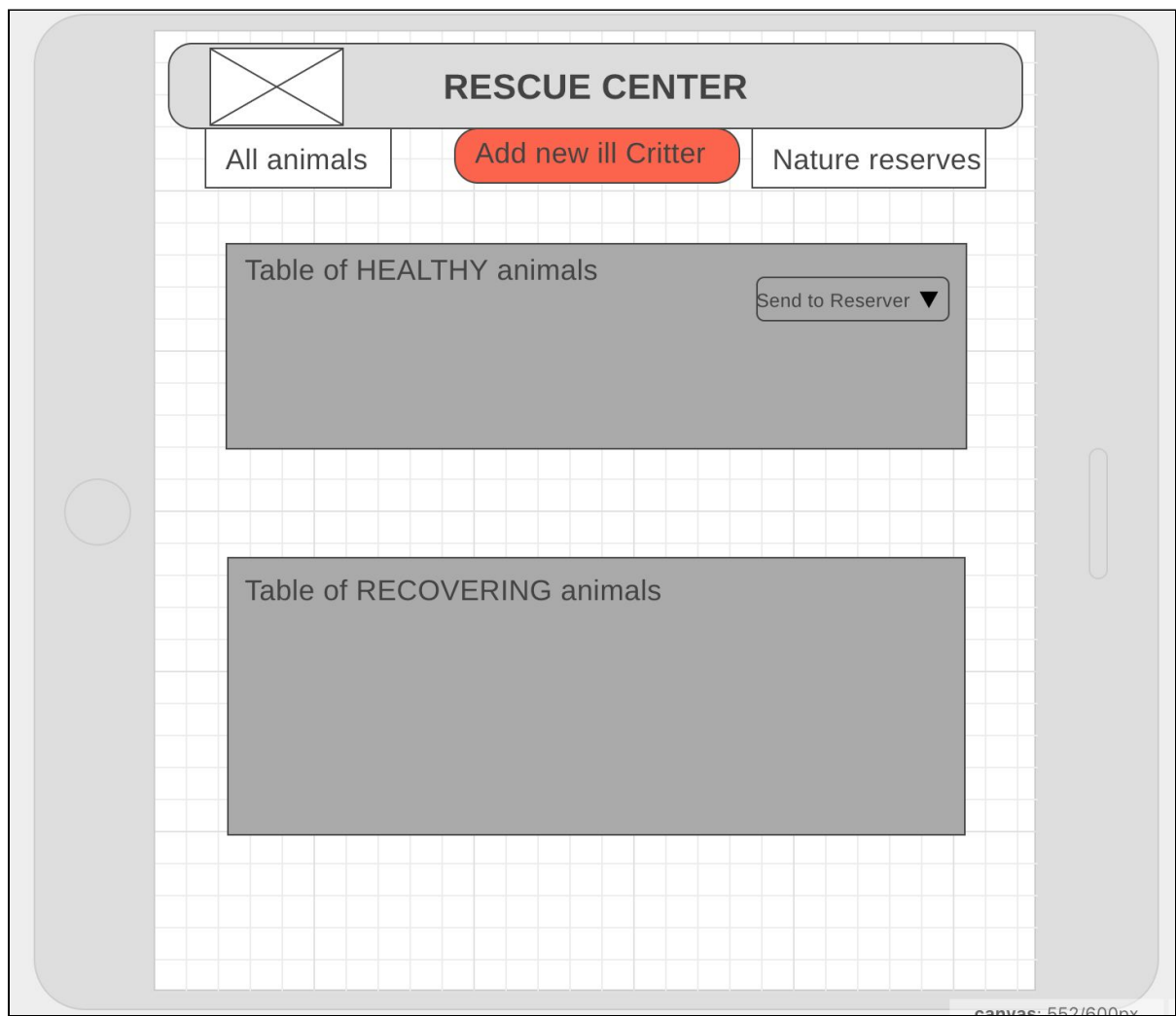
P 5 - Create a user sitemap

- A Site map:

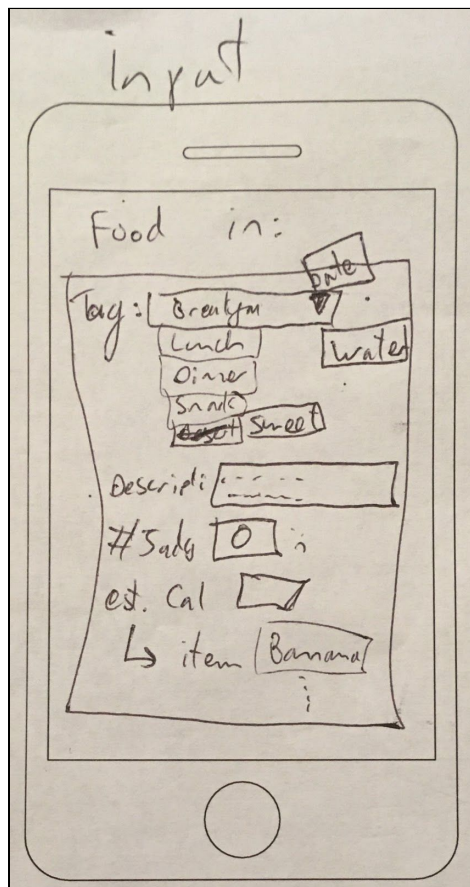


P 6 - Produce two wireframe designs

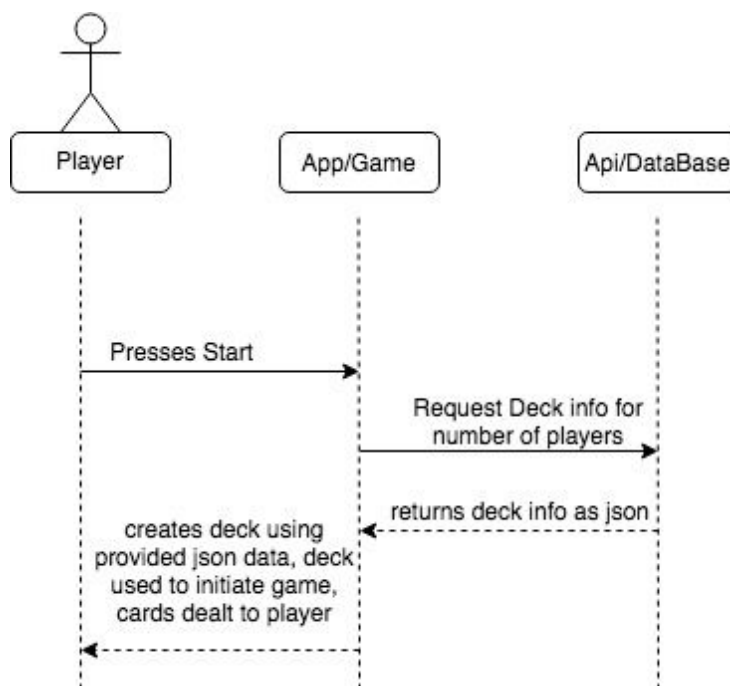
- Wire Frame 1:



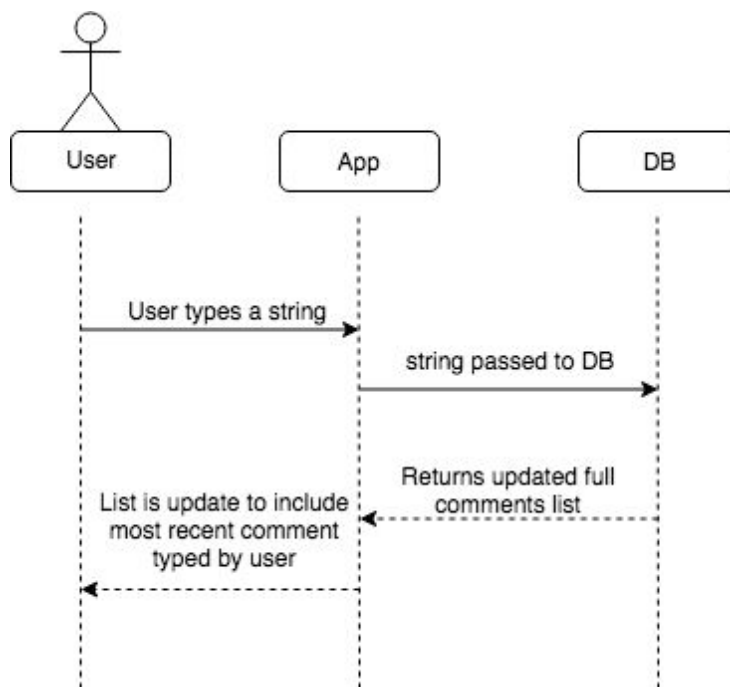
- Wire Frame 2:



P 7 - Produce two system interaction diagrams



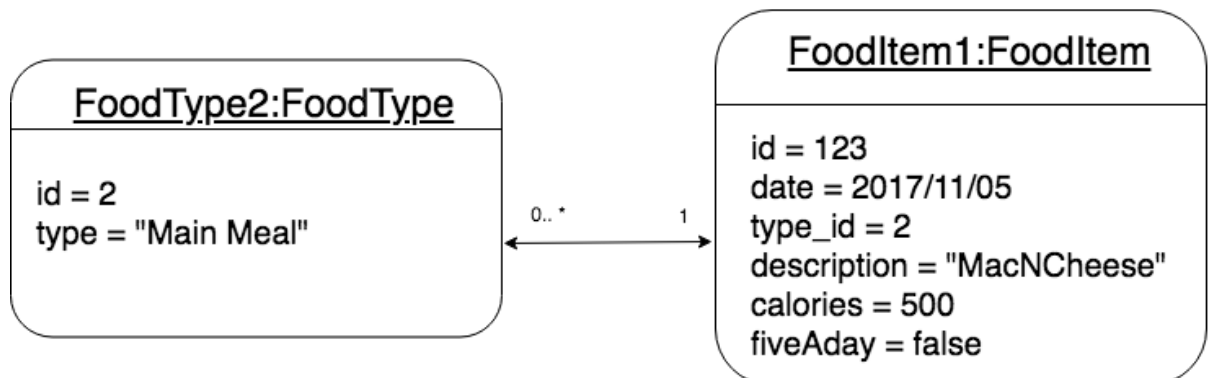
Game being started



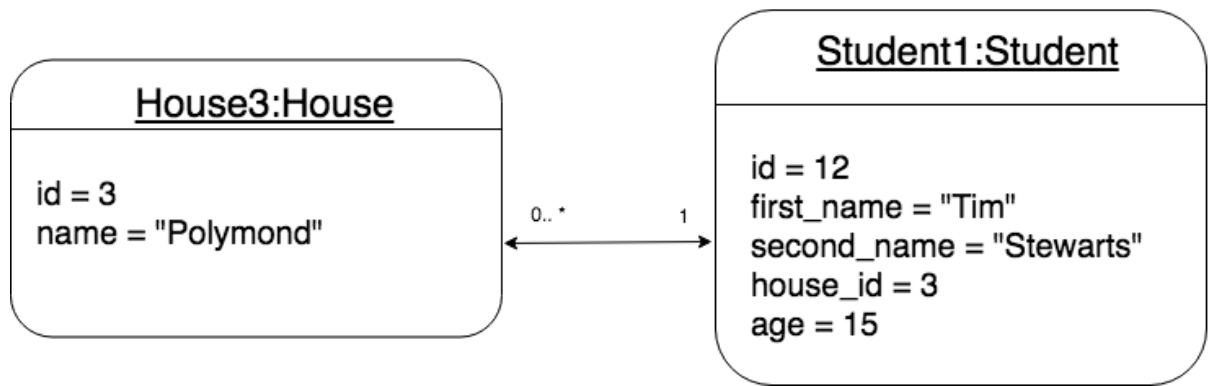
Web App chat list page, on comment input.

P 8 - Produce two object diagrams:

- FoodItem Object:



- Student Object:



P 9

- Select two algorithms you have written (NOT the group project). Take a screenshot of each and write a short statement on why you have chosen to use those algorithms.

```
| public String countWords(String stringOfWords) {
    String[] words = stringOfWords.split("\\s+");

    for (int i = 0; i < words.length; i++) {
        // This replaces all no a-z_A-Z_0-9 with nothing, i.e. removes punctuation.
        words[i] = words[i].replaceAll("[^\\w]", "").toLowerCase();
        if (wordListCounted.containsKey(words[i]) == true ) {
            int oldcount = wordListCounted.get(words[i]);
            int newcount = oldcount + 1;
            wordListCounted.put(words[i], newcount);
        } else {
            wordListCounted.put(words[i], 1);
        }
    }
}
```

This algorithm takes a string of words input, and outputs a hashmap of unique words as keys with their count as the value. Without it there is no good way to actually get the number of times each word is used in the sentence. Any string can be input, it's repeatable and efficient.


```
render() {
  var buttons = this.props.buttons.map((button, index) => {
    return (
      <GameButton
        turn={this.props.turn}
        value={button}
        key={index}
        index={index}
        handle={this.props.handle}>
      </GameButton>
    );
  });
}
```

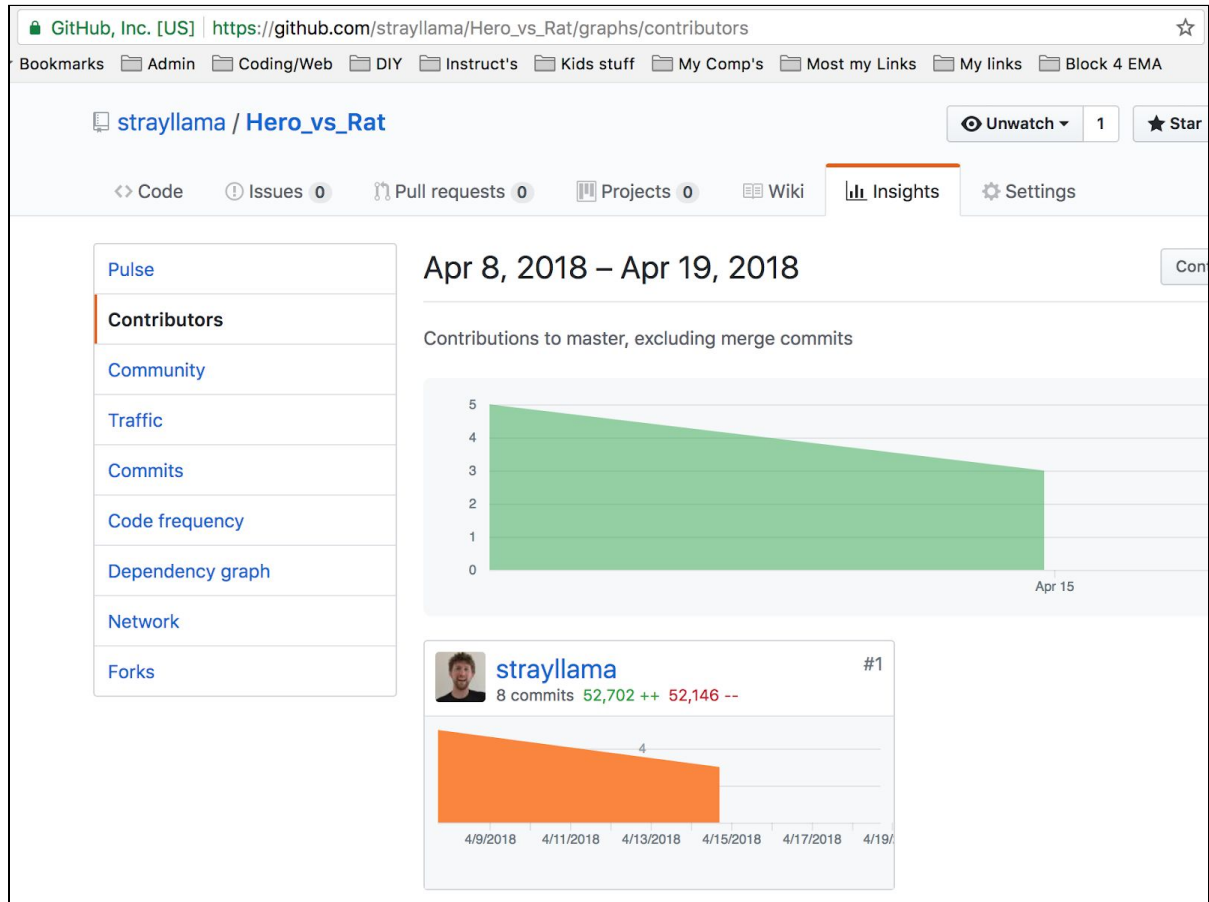
This one maps over an array `this.props.buttons` and returns an object `<GameButton>` for each item in the array. I used it as the action is repeatable, so saves writing it out many time, but also now doesn't rely on you knowing how many are being input, just that the format is the same.

P 10 - A screenshot of an example of pseudocode for a function:

```
sudo_code.txt
1 Method called pick random card, input an ArrayList of Card objects.
2 Create new local variable of type Random called rand
3 Initialise rand and use 0 to length of input ArrayList as limiter for range of numbers.
4 return object from ArrayList of index rand.
```

P 11

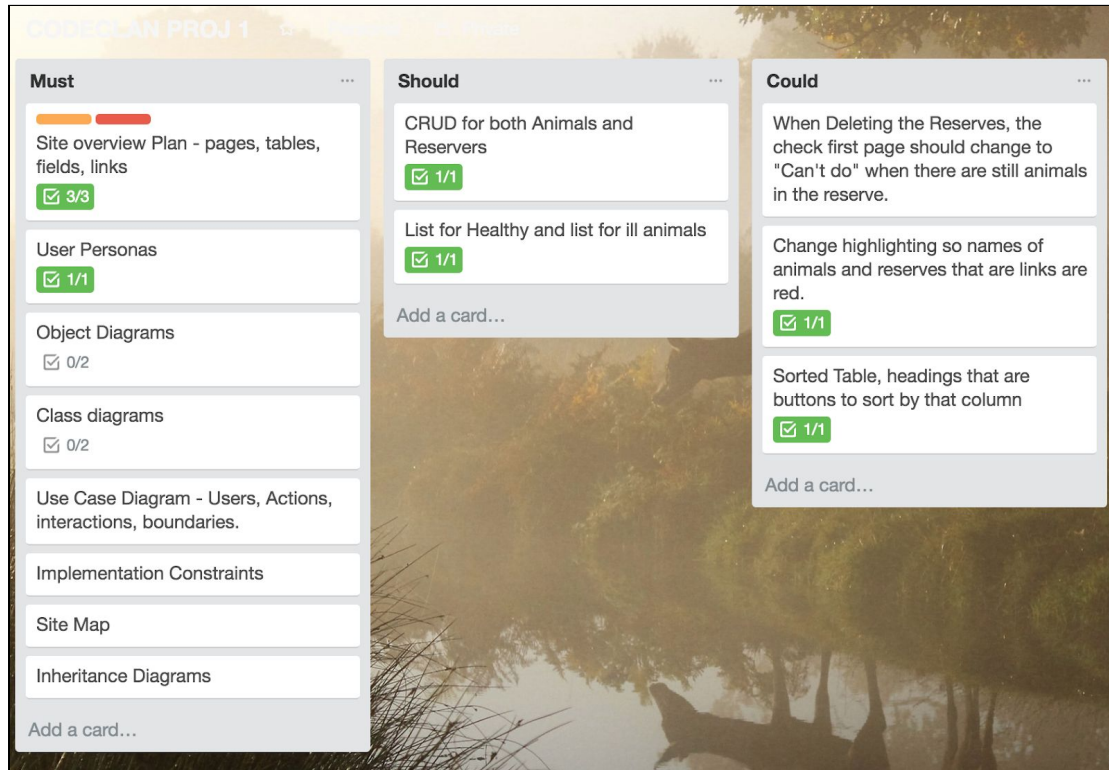
- Take a screenshot of one of your projects where you have worked alone and attach the Github link:



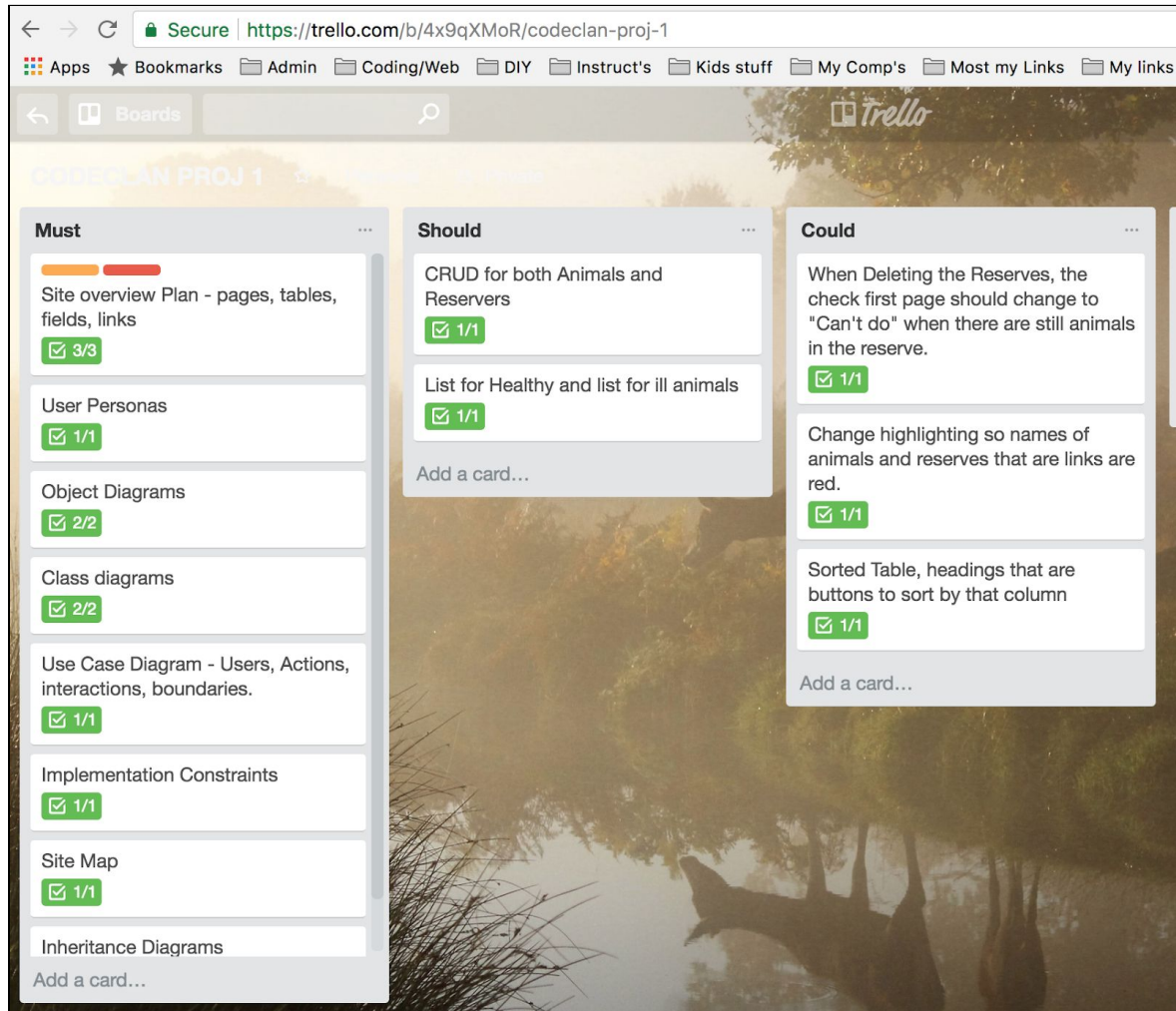
LINK: https://github.com/strayllama/Hero_vs_Rat

P 12

- Take screenshots or photos of your planning and the different stages of development to show changes:



Midway through project



End of Project

P 13 + P 14 + P 15

- User input being processed according to design requirements AND show an interaction with data persistence AND show the correct output of results and feedback to user:
 - The user inputting Data into your program - FILLED OUT FORM:

Admit New Animal

Name:	<input type="text" value="Flappy"/>
Species:	<input type="text" value="Seagul"/>
Terrain Type:	<input type="text" value="Aquatic"/>
Admission Date:	<input type="text" value="02 / 04 / 2018"/>
Healthy Percentage:	1% <input type="range" value="50"/> 100%
Age:	<input type="text" value="Young"/>
Gender:	<input type="radio"/> TBC <input checked="" type="radio"/> Male <input type="radio"/> Female
Size:	<input type="text" value="Large"/>
Colour:	<input type="text" value="Gray"/>

- The user input / data being saved and processed in some way - LISTED:

Dashboard for Rescue Center

[Admit New ILL Critter](#)

Healthy Animals

Admission Date	Name	Species	Terrain Type	Age	Size	RE-HOME
2018-01-30	Eli	Wild Cat	Land	Young	Large	<input type="button" value="Send to reserve"/>
2018-02-05	Ted	Human Boy	Land	Young	Small	<input type="button" value="Send to reserve"/>

Recovering Animals

Admission Date	Name	Species	Terrain Type	Age	Size	Health Status
2018-03-01	Bunster	Bunny	Land	Young	Small	50%
2017-12-18	Mr Blobby	Blob	Land	Elderly	Large	30%
2018-03-01	Frodo Bags	Hobo	Land	tbc	Small	11%
2018-03-01	Molly	Sheep	Land	Young	Small	50%
2018-02-26	John Snow	Crow	Tree top	Young	Medium	1%
2018-04-02	Flappy	Seagul	Aquatic	Young	Large	50%

P 16

- Show an API being used within your program. Take a screenshot of:
 - The code that uses or implements the API

```

9
10 // Starting URL is selecting todays earthquakes
11 const url = `https://earthquake.usgs.gov/fdsnws/event/1/query?format=geojson&starttime=2018-04-19&endtime=2018-04-20`;
12 // then we pull the default data
13 const quakeData = new QuakeData(url);

```

QuakeData is Request.js

```

1  const Request = function(url) {
2    this.url = url;
3  };
4
5  Request.prototype.get = function(onComplete) {
6    const request = new XMLHttpRequest();
7    request.open('GET', this.url);
8    request.addEventListener('load', function() {
9      if(this.status !== 200) {
10        return;
11      }
12
13      const responseBody = JSON.parse(this.responseText);
14      onComplete(responseBody);
15    });
16    request.send();
17  };
18
19  module.exports = Request;
20

```

- The API being used by the program whilst running

localhost:3000
Apps
Bookmarks
Admin
Coding/Web
DIY
Instruct's
Kids stuff
My Comp's
Most my

EarthQuake Looker-upper

Magnitude Above:

Pick a maganatude range:
Click to re-fresh EarthQuake list

Choose an EarthQuake: Select from below to Zoom:

Earthquake occurred on: Thu Apr 19 2018 07:16:42 GMT+0100 (BST)
32km WNW of Rincon, Puerto Rico.
Magnitude: 2.57.

Google
Map data ©2018 Imagery ©2018 NASA Terms of Use

- 150km SSE of `Ohonua, Tonga, Date: Fri Apr 20 2018 00:59:55 GMT+0100 (BST), Magnitude: 5.3
 - Fri Apr 20 2018 00:59:55 GMT+0100 (BST)
- 23km NNE of Badger, Alaska, Date: Fri Apr 20 2018 00:55:23 GMT+0100 (BST), Magnitude: 1.2
 - Fri Apr 20 2018 00:55:23 GMT+0100 (BST)
- 8km SSW of Huron, CA, Date: Fri Apr 20 2018 00:51:46 GMT+0100 (BST), Magnitude: 2.92

P 17 - A Bug tracking Reports:

BUG	Fail / Pass
Player can save can change their name	Pass
Discarded cards appear on the discard pile	Pass
Number of active players can be chosen as 2,3 or 4	Fail
Deck takes out one card before dealing player cards	Fail
When player wins, as the deck runs out of cards, the message should be	Fail

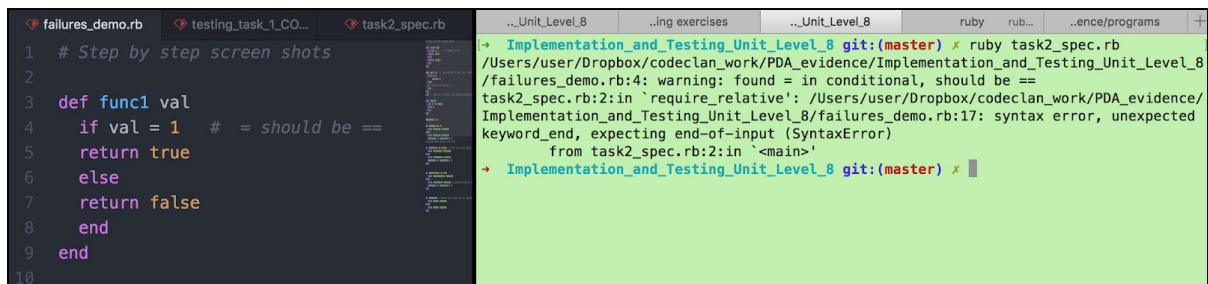
player won, not cards ran out.	
--------------------------------	--

P 18 - Testing in my program:

- Example of test code:

```
1 require("minitest/autorun")
2 require_relative("failures_demo.rb")
3
4
5 class Task_2Test < MiniTest::Test
6
7   def setup
8   end
9
10  def test_func1_gives_true_with_1
11    answer = func1(1)
12    assert_equal(true, answer) # test 1
13  end
14
15  def test_func1_gives_false_with_anything_else
16    answer = func1(2)
17    assert_equal(false, answer) # test 2
18  end
19
20  def test_max_returns_largest_number_when_first
21    answer = max(2,1)
22    assert_equal(2, answer) # test 3
23  end
24
25  def test_max_returns_largest_number_when_second
26    answer = max(1,2)
27    assert_equal(2, answer) # test 4
28  end
29
30 end
```

- The test code failing to pass:



```
1 # Step by step screen shots
2
3 def func1 val
4   if val = 1 # = should be ==
5     return true
6   else
7     return false
8   end
9 end
10
```

```
Implementation_and_Testing_Unit_Level_8 git:(master) x ruby task2_spec.rb
/Users/user/Dropbox/codeclan_work/PDA_evidence/Implementation_and_Testing_Unit_Level_8
/failures_demo.rb:4: warning: found = in conditional, should be ==
task2_spec.rb:2:in `require_relative': /Users/user/Dropbox/codeclan_work/PDA_evidence/
Implementation_and_Testing_Unit_Level_8/failures_demo.rb:17: syntax error, unexpected
keyword_end, expecting end-of-input (SyntaxError)
    from task2_spec.rb:2:in `'
Implementation_and_Testing_Unit_Level_8 git:(master) x
```

Failing on line 4 due to single = where double == is needed.

- Example of the test code once errors have been corrected + test code passing:

```
9 end
10
11 def max (a, b) # dif should be def
12   if a > b
13     return a
14   else
15     return b # should be return b
16   end
17 end # This is an extra end, needs r
18
19 def loopier
20   for i in 1..10
21     puts i
22   end
23 end
24
25 failures = 0
26
27 if loopier == 10
28   puts "loopier passed"
29 else
30   puts "loopier failed"
31   failures = failures + 1
32 end #<----- Insert end here
33
34 if func1(3) == false # Code runs bu
35   puts "func1(3) passed"
36 else
37   puts "func1(3) failed"
38   failures = failures + 1
39 end
```

```
task2_spec.rb:2:in `require_relative': /Users/user/Dropbox/codeclan_work/PDA_evidence/
Implementation_and_Testing_Unit_Level_8/failures_demo.rb:11: syntax error, unexpected
tIDENTIFIER, expecting ';' or '\n' (SyntaxError)
def max a b # dif should be def AND '
^
/Users/user/Dropbox/codeclan_work/PDA_evidence/Implementation_and_Testing_Unit_Level_8
/failures_demo.rb:17: syntax error, unexpected keyword_end, expecting end-of-input
from task2_spec.rb:2:in `<main>'
-> Implementation_and_Testing_Unit_Level_8 git:(master) x ruby task2_spec.rb
task2_spec.rb:2:in `require_relative': /Users/user/Dropbox/codeclan_work/PDA_evidence/
Implementation_and_Testing_Unit_Level_8/failures_demo.rb:18: syntax error, unexpected
keyword_end, expecting end-of-input (SyntaxError)
end # This is an extra end, need
^
from task2_spec.rb:2:in `<main>'
-> Implementation_and_Testing_Unit_Level_8 git:(master) x ruby task2_spec.rb
task2_spec.rb:2:in `require_relative': /Users/user/Dropbox/codeclan_work/PDA_evidence/
Implementation_and_Testing_Unit_Level_8/failures_demo.rb:55: syntax error, unexpected
end-of-input, expecting keyword_end (SyntaxError)
from task2_spec.rb:2:in `<main>'
-> Implementation_and_Testing_Unit_Level_8 git:(master) x ruby task2_spec.rb
1
2
3
4
5
6
7
8
9
10
loopier failed
func1(3) passed
max(100,1) passed
Test Failed
Run options: --seed 16771
# Running:
....
Finished in 0.000912s, 4386.7586 runs/s, 4386.7586 assertions/s.
4 runs, 4 assertions, 0 failures, 0 errors, 0 skips
-> Implementation_and_Testing_Unit_Level_8 git:(master) x
```