

Evidence for Project Unit

Ewa Lipinska
Cohort E20

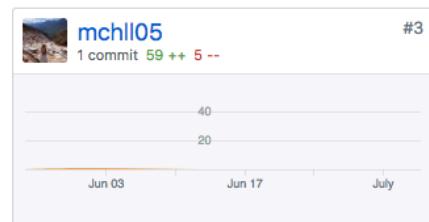
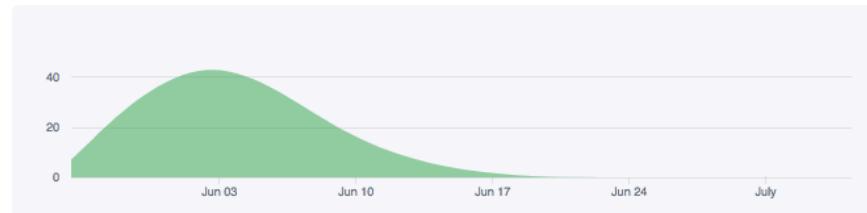
P. 1 Github Contributors page

Collaborators	Push access to the repository
 Sebastian Hatt SebHatt	X
 Michelle Austin mchll05	X
 b09	X
Search by username, full name or email address You'll only be able to find a GitHub user by their email address if they've chosen to list it publicly. Otherwise, use their username instead.	
<input type="text"/>	Add collaborator

May 27, 2018 – Jul 6, 2018

Contributions: Commits ▾

Contributions to master, excluding merge commits



P. 2 Project Brief

NASA International Space Station Tracker

Now deployed to Heroku at <https://nasa-iss-tracker.herokuapp.com/>

Project definition

An interactive educational app built with JavaScript using Node.js, webpack and Express for the server side. It uses positional data from the International Space Station (ISS) as a fun way for users to retrieve educational material about locations around the Earth! Created within a 4-person Agile team over the course of a week.

MVP

- Take user selection of a location and display a prediction of the next time the ISS will be above that location.
- A dynamic map showing the ISS current location related to city selection.

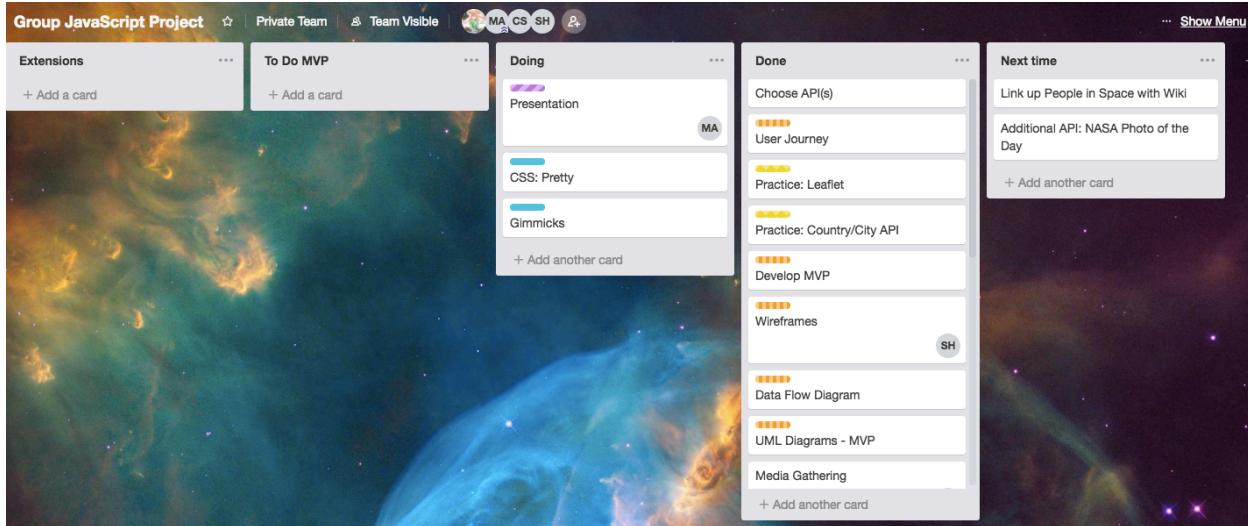
Extensions

- Take user selection of a location and display educational material relevant to that location.
- Use NASA media resources to provide relevant and appealing background content.

Data sources used

- [Open Notify API](#) for the ISS data,
- [Leaflet](#) for map rendering
- [Wikipedia API](#) for additional information about locations
- [Countries](#) and [cities](#) npm packages

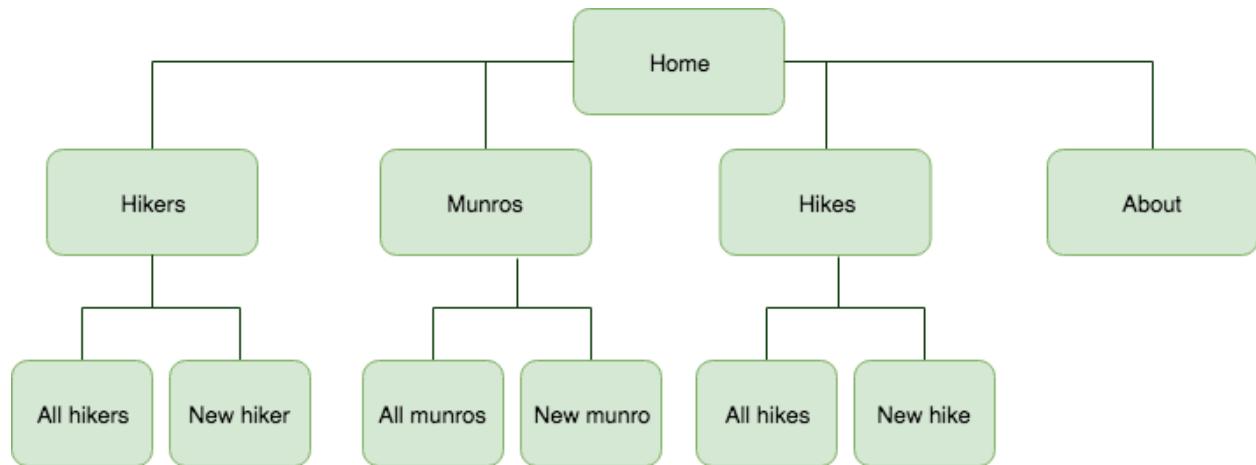
P. 3 Use of Trello



P. 4 Acceptance Criteria

Acceptance criteria	Expected result/output	Pass/Fail
The user can see the current position of the International Space Station (ISS)	The app displays a map with an icon placed at the coordinates of the ISS taken from the Open Notify API. The position of the icon is updated every 5 seconds.	Pass
The user can select a city to view ISS pass times.	The app displays a list of cities based on user input into a search box – when a city is selected, a div is displayed underneath which contains a list of 5 next times the ISS will pass over the selected city taken from the Open Notify API.	Pass
The user can filter cities by country	When a country is selected from a dropdown populated with data from the countries npm package, the cities displayed when searching are only limited to those in that country.	Pass
The user can view a random Wiki article for a chosen city	Upon city selection, a call is made to the Wikipedia API and a random article for the city's coordinates is displayed at the bottom of the page.	Pass
The user can see a random Wiki article for the current location of the ISS	When the 'What's below the ISS right now?' button is clicked, a call to the Open Notify API is made to get the ISS current location and that location is passed to the Wikipedia API to get a random article based on the coordinates, which is then displayed at the bottom of the page.	Pass
The user can see the current view from ISS	A NASA video feed embedded in the app provides the current view of Earth from ISS	Pass
The user can see astronauts currently in space	When the app has loaded, the 'People in space' table is populated with the list of names taken from the Open Notify API.	Pass

P. 5 User sitemap



P. 6 Wireframes designs

Home Hikers Munros Hikes About

More about John Smith:



New hike

Edit profile

Delete profile

Highest munro bagged:

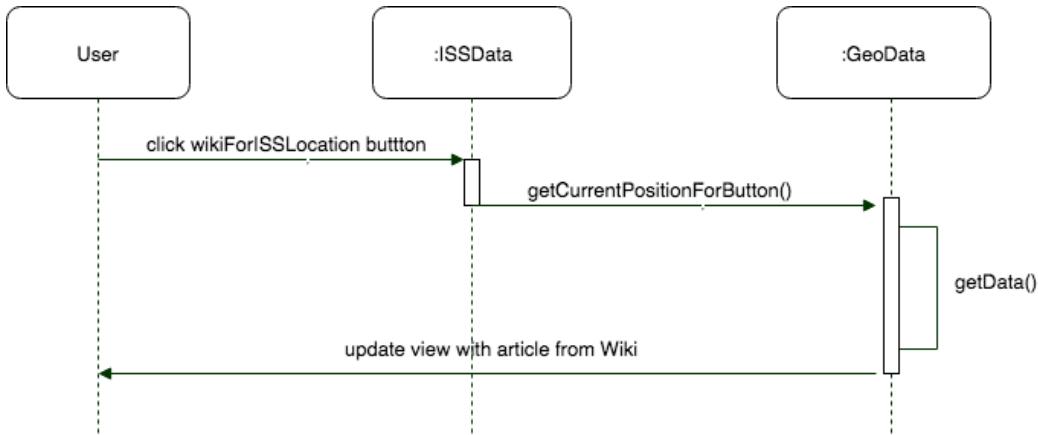
Ben Nevis
1345m



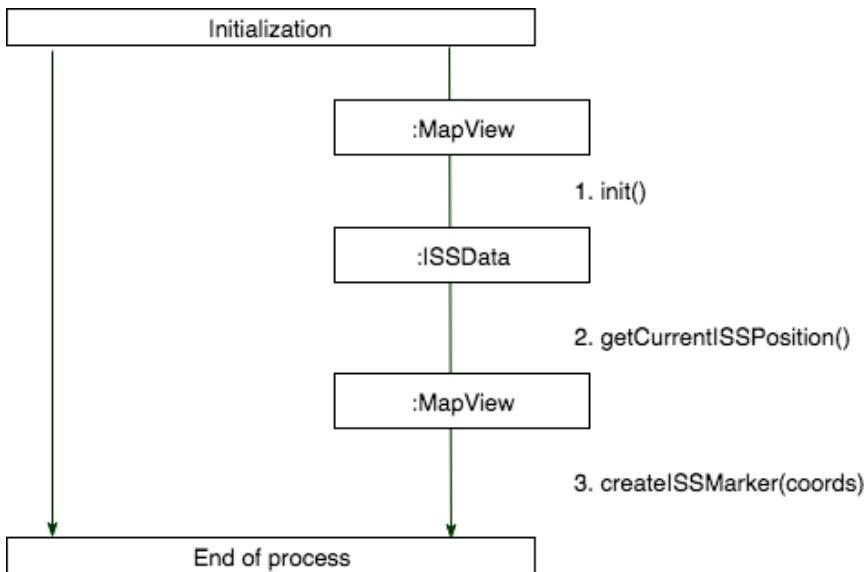
Table Header	Table Header
Content	This is longer content

P. 7 System interactions diagrams

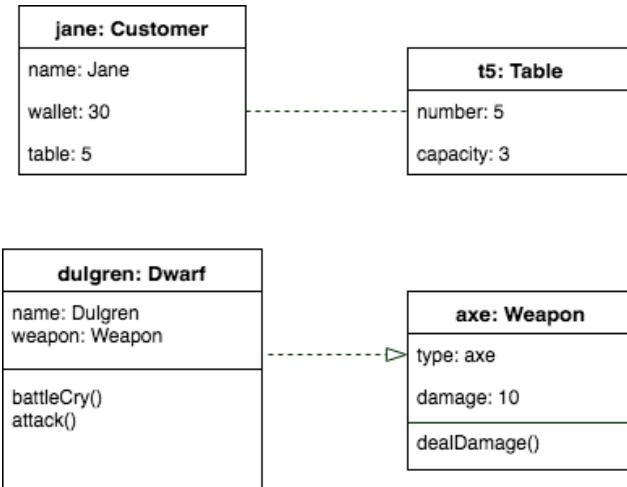
Sequence diagram – display a Wiki article for ISS location



Collaboration diagram – display map marker for selected city



P. 8 Two Object Diagrams



P. 9 Choice of two algorithms (find the algorithms on a program you might have written, show the code you have used.)

```
public void uncoverFieldAndNeighbours(Field field) {  
    if (!field.getIsLongPressed()) {  
  
        if (field.getFieldType() == FieldType.GHOST) {  
            if (!field.isUncovered()) {  
                ((GhostField) field).activate();  
            }  
            uncoverAll();  
        } else {  
  
            field.markAsUncovered();  
            addToUncoveredFieldsCount();  
  
            if (field.getFieldType() == FieldType.EMPTY) {  
                ArrayList<Field> neighbours = board.getAllNeighboursForField(field);  
                for (Field neighbour : neighbours) {  
                    if (neighbour.getFieldType() != FieldType.GHOST && !neighbour.isUncovered()) {  
                        uncoverFieldAndNeighbours(neighbour); //recursion  
                    }  
                }  
            }  
        }  
    }  
}
```

This algorithm is taken from a Minesweeper-like game; when a field on the board is pressed, the game uncovers it (unless the field is blocked with a trap). If the uncovered field contains a ghost, the player loses. If the uncovered field is empty, the method will be called again in a recursive manner on all its neighbours. I used this algorithms to practice recursion and to make my code as efficient and DRY as possible.

```
// should be able to find all records which match on multiple attributes
RecordStore.prototype.findRecordsByAttributes = function(query) {

  let attributeTypes = Object.keys(query);

  return this.recordCollection.filter((record) => {
    return attributeTypes.every((attribute) => {
      return record[attribute] === query[attribute];
    });
  });
};
```

The above search algorithm is part of a RecordStore class. It takes a query (a hash with key-value pairs such as

```
{
  title: 'Master of Puppets',
  artist: 'Metallica',
  genre: 'metal',
  price: 15
}
```

and iterates through the store's recordCollection in order to find all Record objects which match the query on all parameters. I chose this algorithms as it uses JavaScript's efficient enumeration methods and because it's easy to read.

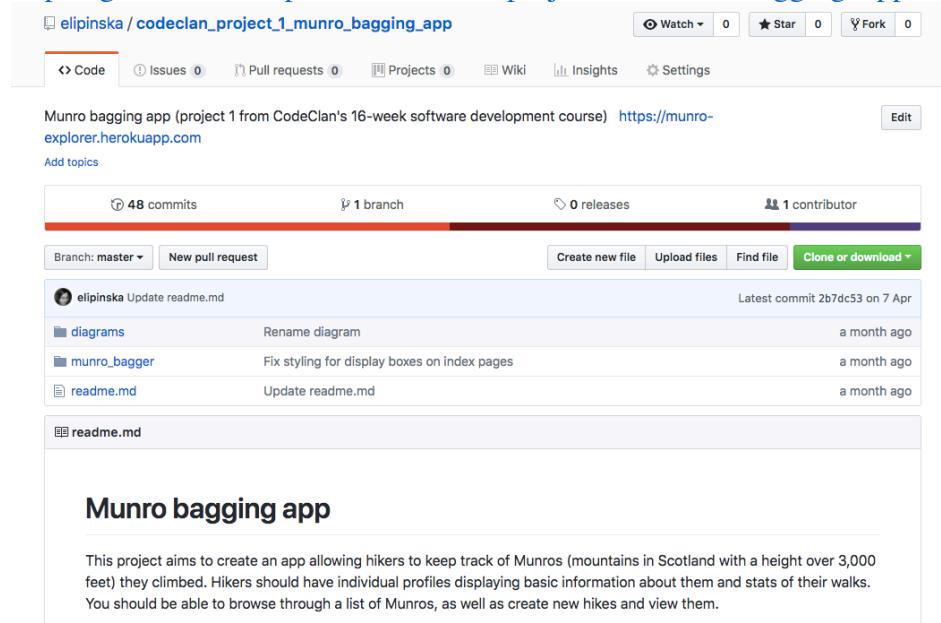
P. 10 Example of Pseudocode

```
//  
//  
// ghostsToAdd is equal to the ghostCount attribute of the Board  
//  
// for the number of times equal to ghostsToAdd {  
//     pick a random y coordinate smaller than the Board's rowNo  
//     pick a random x coordinate smaller than the Board's column number (10 by default)  
//     get the Field at the coordinates y, x and put it in a newGhostField variable  
//  
//     if newGhostField doesn't already contain a ghost {  
//         replace HintField with GhostField  
//     } else {  
//         add one to ghostsToAdd  
//     }  
}  
  
private void addGhostsToGhostSweeperBoard() {  
  
    int ghostsToAdd = ghostCount;  
  
    for (int i = 0; i < ghostsToAdd; i++) {  
        Random rand = new Random();  
        int y = rand.nextInt(rowNo);  
        int x = rand.nextInt(bound: 10);  
        Field newGhostField = fields.get(y).get(x);  
        if (!(newGhostField.getFieldType() == FieldType.GHOST)) {  
            fields.get(y).set(x, new GhostField(new Position(x, y)));  
        } else {  
            ghostsToAdd +=1;  
        }  
    }  
}
```

(The function aims to populate the board for a GhostSweeper game (similar to MineSweeper) with Ghosts when a new Board is created)

P. 11 Github link to one of your projects

https://github.com/elipinska/codeclan_project_1_munro_bagging_app

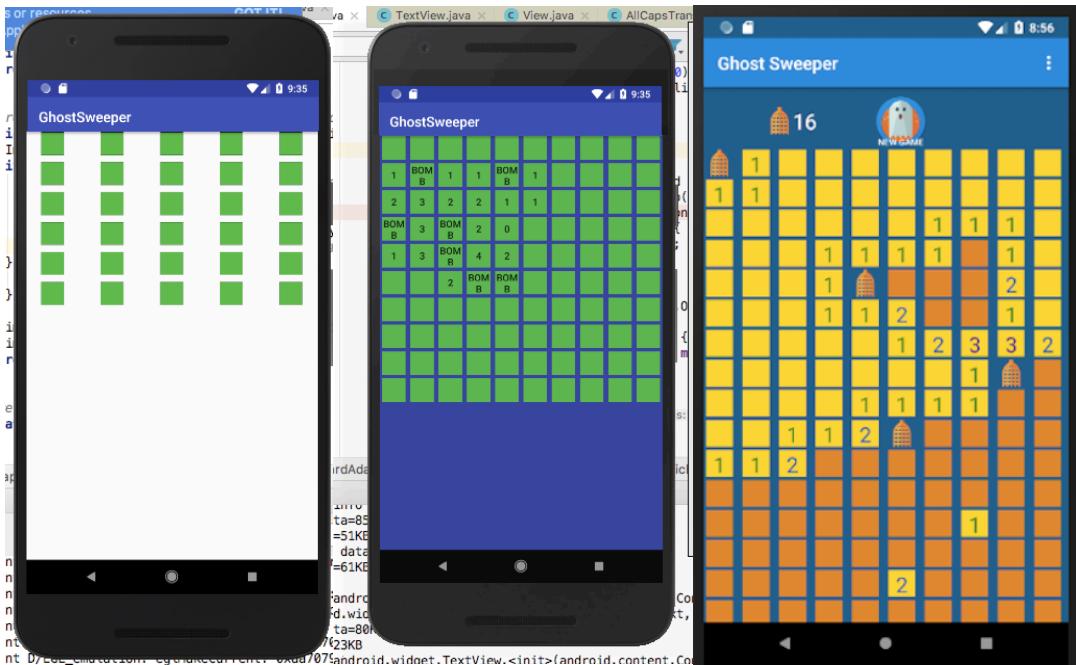
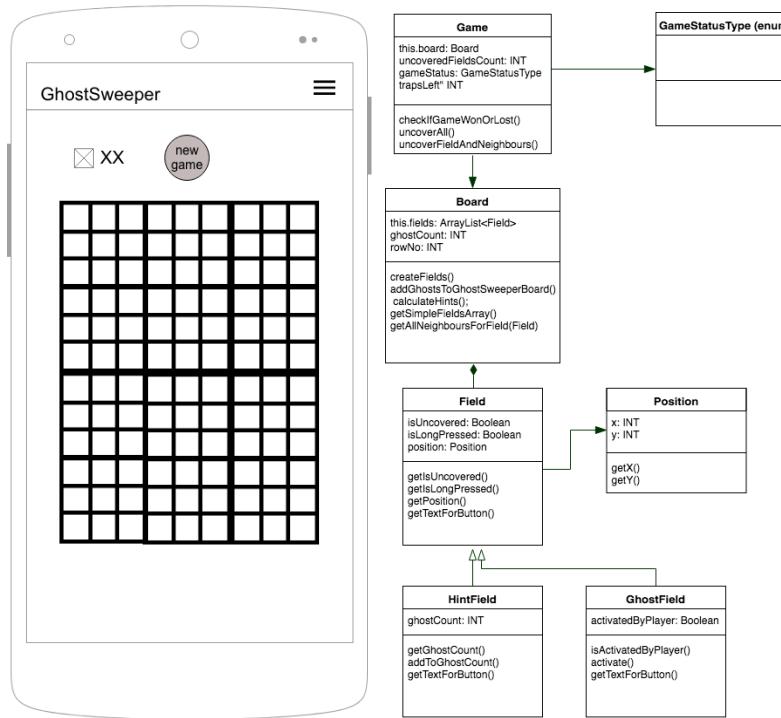


The screenshot shows a GitHub repository page. At the top, there is a header with the repository name 'elipinska / codeclan_project_1_munro_bagging_app' and buttons for 'Watch 0', 'Star 0', 'Fork 0'. Below the header, there are tabs for 'Code' (selected), 'Issues 0', 'Pull requests 0', 'Projects 0', 'Wiki', 'Insights', and 'Settings'. A note below the tabs says 'Munro bagging app (project 1 from CodeClan's 16-week software development course) <https://munro-explorer.herokuapp.com>' with an 'Edit' button. The main area shows statistics: 48 commits, 1 branch, 0 releases, and 1 contributor. Below this, a list of commits is shown:

Author	Commit Message	Date
elipinska	Update readme.md	Latest commit 2b7dc53 on 7 Apr
diagrams	Rename diagram	a month ago
munro_bagger	Fix styling for display boxes on index pages	a month ago
readme.md	Update readme.md	a month ago

At the bottom, there is a section titled 'Munro bagging app' with a description: 'This project aims to create an app allowing hikers to keep track of Munros (mountains in Scotland with a height over 3,000 feet) they climbed. Hikers should have individual profiles displaying basic information about them and stats of their walks. You should be able to browse through a list of Munros, as well as create new hikes and view them.'

P. 12 Screenshot of your planning and the different stages of development to show changes.



P. 13 User input

The image shows two consecutive screenshots of a Java application window titled "experiment — java -jar CastleCapture.jar — java — java -jar CastleCapture.jar...".

Screenshot 1: The window displays the message "Welcome, brave adventurer! What's your name? Jen".

Screenshot 2: The window displays the message "Hello, Jen!". Below it, a prompt says "Pick your race by typing one of the options: dwarf, barbarian, knight". The text "Hello, Jen!" is highlighted with a red box.

P. 14 Interaction with data persistence

The image shows a web application interface for managing Munros (mountains).

Add a new Munro: A modal dialog box is open, prompting for "Name" (Ben Lawers), "Region" (Perthshire), and "Altitude (in meters)" (1214). An "Add Munro" button is at the bottom right, and a small mountain icon is to the right of the input fields.

All munros to explore: A grid of eight cards, each representing a mountain with a green mountain icon:

- A' Bhuidheanach Bheag**
Region: Cairngorms
Altitude: 936m
- A' Chailleach**
Region: Ullapool
Altitude: 997m
- A' Chailleach (Monadhliath)**
Region: Cairngorms
Altitude: 930m
- Ben Hope**
Region: Sutherland
Altitude: 927m
- Ben Lawers** (highlighted with a red border)
Region: Perthshire
Altitude: 1214m
- Ben Lomond**
Region: Loch Lomond
Altitude: 974m
- Ben Macdui**
Region: Cairngorms
Altitude: 987m
- Ben Nevis**
Region: Fort William
Altitude: 1345m

P. 15 User output result

All intrepid hikers:



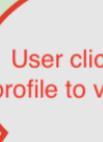
Ewa Lipinska

4 munros bagged to date.
Last spotted on the slopes of Ben Nevis on 29/03/2018.



James Yates

3 munros bagged to date.
Last spotted on the slopes of A' Bhuidheanach Bheag on 28/09/2017.



1 munro bagged to date.
Last spotted on the slopes of A' Bhuidheanach Bheag on 07/02/2018.

User clicks on a hiker's profile to view more details

More about James Yates:



New hike

Edit profile

Delete profile

Distinction



Most individual hikes

Name: James Yates
Age: 29
Last spotted on the slopes of A' Bhuidheanach Bheag on 28/09/2017.

Munros climbed (3 in total):

Name	Last climbed
A' Bhuidheanach Bheag	2017-11-09
Ben Macdui	2018-03-28
Inaccessible Pinnacle	2018-04-05

All of James's hikes:

Name	Date
Inaccessible Pinnacle	2018-04-05
Ben Macdui	2018-03-28
Ben Macdui	2018-02-08

P. 16 API being used in a program

Code that implements an API

```
index.html movie_filt... movie_dat... app.js movie_sc X movie_sor... movie_vie... config.json
1 const PubSub = require('../helpers/pub_sub.js');
2 const Request = require('../helpers/request_helper.js');
3 const config = require('....config.json');
4
5 const MovieData = function() {
6   this.movieData = null;
7   this.filter = null;
8   this.director = 'All';
9 };
10
11 MovieData.prototype.getData = function () {
12   const request = new Request(config.movieAPI);
13   this.selectSortBy();
14   request.get( (movieData) => {
15     this.movieData = movieData;
16     PubSub.publish('MovieData:movies-ready',
17       this.movieData);
18   });
19 }
20 MovieData.prototype.receiveFilter = function() {
21   PubSub.subscribe('MovieFilterView:filtered-director',
22     (event) => {
23       this.director = event.detail;
24       this.sortAndFilter();
25     });
26 }
```

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

API being used in a program whilst running

Ghibli's PubSub Service

Filter by Director Sort by

Movie	Year
 Castle in the Sky	1986
 My Neighbor Totoro	1988
 Kiki's Delivery Service	1989

P. 17 Bug tracking report showing the errors diagnosed and corrected.

Action	Result	Fix	Result
Leaflet API map should display when the page is loaded	Fail	Change map layer to a more lightweight one	Pass
Move ISSN icon across the map every 5 seconds based on the coordinates from API	Fail	Implement a function which ensures that every 5 seconds the icon is removed and instantly recreated at an updated ISS location	Pass
Display people currently in space in a table	Pass		
Display a list of cities matching user search	Fail	Limit the list of cities to display by only showing the dropdown after the user enters 3 first letters of the city's name	Pass

P. 18 Testing your program

```

card_game_spec.rb -- ~/codeclan/e20/ewa_lipinska_pda/PDA_Static_and_Dynamic_Task_A
testing_task_2.rb      Static_&_Dynamic_Testing.md    testing_task_1.md      card_game_spec.rb
1 require('minitest/autorun')
2 require_relative('../card')
3 require_relative('../testing_task_2')
4
5 class CardGameTest < Minitest::Test
6
7   def setup
8     @ace_of_spades = Card.new("Spades", 1)
9     @five_of_hearts = Card.new("Hearts", 5)
10    @five_of_clubs = Card.new("Clubs", 5)
11    @ten_of_diamonds = Card.new("Diamonds", 10)
12    @seven_of_clubs = Card.new("Clubs", 7)
13    @card_game = CardGame.new()
14  end
15
16  def test_checkforAce_returns_true
17    assert_equal(true, @card_game.checkforAce(@ace_of_spades))
18  end
19
20  def test_checkforAce_returns_false
21    assert_equal(false, @card_game.checkforAce(@five_of_hearts))
22  end
23
24  def test_highest_card_return_first_card
25    assert_equal(@ten_of_diamonds, @card_game.highest_card(@ten_of_diamonds, @five_of_hearts))
26  end
27
28  def test_highest_card_return_second_card
29    assert_equal(@ten_of_diamonds, @card_game.highest_card(@five_of_hearts, @ten_of_diamonds))
30  end
31
32  def test_highest_card_its_a_tie
33    assert_equal("It's a tie.", @card_game.highest_card(@five_of_hearts, @five_of_clubs))
34  end
35
36  def test_cards_total
37    assert_equal("You have a total of 23.", CardGame.cards_total([@ace_of_spades, @five_of_hearts, @ten_of_diamonds, @seven_of_clubs]))
38  end
39end

```

PDA_Static_and_Dynamic_Task_A git:(master) ✘ ruby specs/card_game_spec.rb
 specs/card_game_spec.rb:3:in `require': /Users/macbook/codeclan/e20/ewa_lipinska_pda/PDA_Static_and_Dynamic_Task_A/testing_task_2.rb:25: syntax error, unexpected keyword_end, expecting end-of-input (SyntaxError)
 from specs/card_game_spec.rb:3:in `<main>'
 → PDA_Static_and_Dynamic_Task_A git:(master) ✘ ruby specs/card_game_spec.rb
 specs/card_game_spec.rb:3:in `require_relative': /Users/macbook/codeclan/e20/ewa_lipinska_pda/PDA_Static_and_Dynamic_Task_A/testing_task_2.rb:34: syntax error, unexpected keyword_end, expecting end-of-input (SyntaxError)
 from specs/card_game_spec.rb:3:in `<main>'
 → PDA_Static_and_Dynamic_Task_A git:(master) ✘ ruby specs/card_game_spec.rb
 specs/card_game_spec.rb:3:in `require_relative': /Users/macbook/codeclan/e20/ewa_lipinska_pda/PDA_Static_and_Dynamic_Task_A/testing_task_2.rb:18: syntax error, unexpected TIDENTIFIER, expecting ')' (SyntaxError)
 def highest_card(card2)
 ^
 /Users/macbook/codeclan/e20/ewa_lipinska_pda/PDA_Static_and_Dynamic_Task_A/testing_task_2.rb:34: syntax error, unexpected keyword_end, expecting end-of-input
 from specs/card_game_spec.rb:3:in `<main>'
 → PDA_Static_and_Dynamic_Task_A git:(master) ✘ ruby specs/card_game_spec.rb
 Run options: --seed 38646
Running:
.....
Finished in 0.001925s, 3116.8831 runs/s, 3116.8831 assertions/s.
6 runs, 6 assertions, 0 failures, 0 errors, 0 skips
→ PDA_Static_and_Dynamic_Task_A git:(master) ✘