Lee George Gauci - HND in Interactive media Yr2

# Contents

Task 1 – (P1.1)	4
Functionalities:	4
Repository	5
Commit	5
Issue	6
Sync	6
Add	6
Pull Request	7
TASK 3 – (P1.3)	8
Story Board	9
TASK 4 – (P2.1)	10
Graphics Artist	10
Apprentice 3D game Artist	11
CGI Artist - Maya Based Generalist - Creative agency London	12
What is your favored area of expertise?	14
TASK 5 – (P2.2)	14
TASK 6 – (P3.1)	15
TASK 7 – (P3.2)	16
TASK 8 – (P3.3)	16
TASK 9 – (P4.1)	18
TASK 10 – (P4.2)	19
TASK 11 – (M1.1)	20
What are they?	20
Sourceforge vs GitHub vs GitLab	20
TASK 12 – (M2.1)	21
TASK 13 – (M3.1)	22
Managing Branches	22
Forking the Initial Project.	22
TASK 14 – (D1.1)	23
TASK 15 – (D3.1)	24

Forked Project:	. 24
Modified project	. 25
ASK 16 – (D2.1)	
bliography	
 ~ . ~ A. ~ L	

# Task 1 - (P1.1)

# **Functionalities:**

## **Jump**

When the mouse is clicked the player jumps.

# Megajump

When the mouse is clicked and the player is located on a platform the player megajumps

# **Shrink**

When the player collects a power pellet the player shrinks.

## **Grow**

When the player collects a power pellet the player grows.

# Speed Up

When the player collects a power pellet the player has increased speed.

# **Collect Powerups**

Through collisions the user can collect power ups.

## **Collect Score Items**

Through collisions the user can collect score items.

## Move the player using WASD

When the user presses any of these keys the player moves in the desired direction.

# **Moving Platforms**

There are automatic platforms which are found throughout some levels.

# **Stop Moving in Air**

When the user clicks and the player is in the air the player will not be able to move.

# **TASK 2 - (P1.2)**

Define and analyse target group to identify user needs by familiarizing yourself with the communication tools available in Github.

# Repository

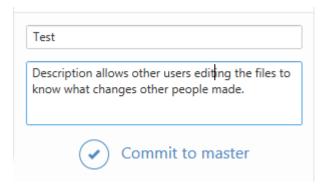


A repository is a directory where storage space is allocated, this allocates space contains all the project files; it also stores each files revision history and can be published both private and public.

## **Commit**

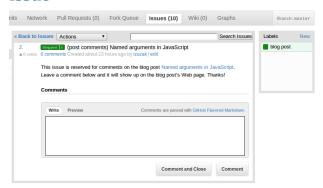


The commit function takes a "screenshot of" each step of your work in the given repository. This allows the user to restore or reevaluate previous versions of your work.



If one or more users are editing the same file, Commit description allows users to add detailed information regarding the changes done by that specific user. This helps organize work and reduce confusion.

## **Issue**



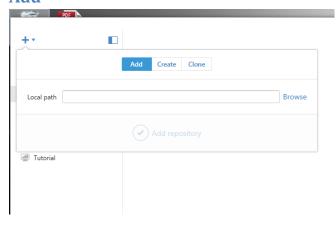
Users can comment problems regarding your code/files found in your GitHub repositories. Issues is a system built in GitHub which organizes those comments in sections and labels better viewing, the master can then tick the comments and resolved.

# **Sync**



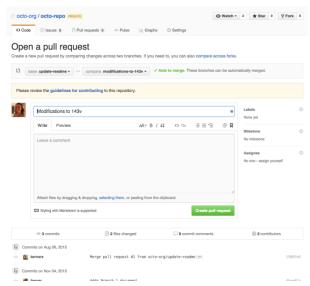
When editing files, everything is still done locally I.E your pc. Git Sync allows the user to upload the contents of the repository to the internet which can later be viewed and downloaded.

### Add



On GitHub desktop, you can add files to your current repository or if the location is not found there is an option to create it as new repository.

# **Pull Request**



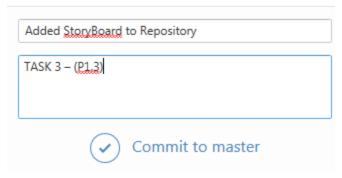
This allows you to notify others about the personal changes you have pushed in the master repository on GitHub. Once the request is sent, interested people can then review and discuss the changes and further discuss potential modifications. They can also push follow-up commits if necessary.

Once you have defined the above, explain which one of the above features would be useful to create a list of requirements as requested by users.

The issue system would be the best way to allow users to submit their requirements as comments and allows them to submit them in a more organized fashion using sections and labels.

# TASK 3 - (P1.3)

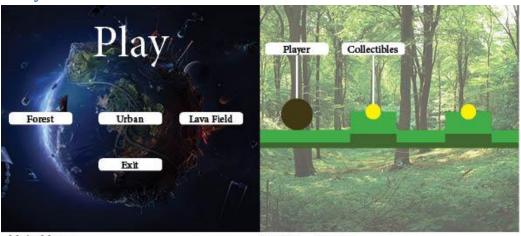


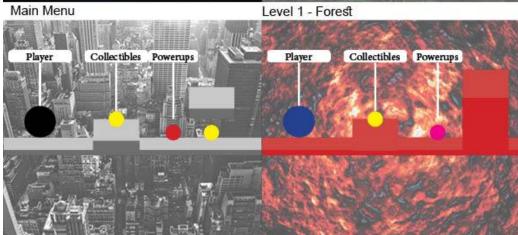


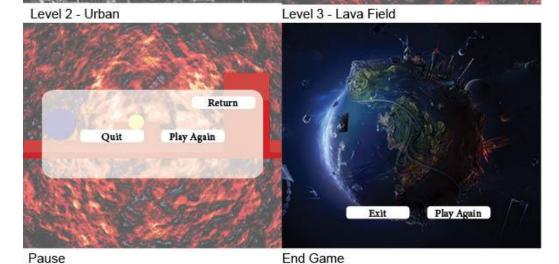
Link to GitHub Commit:

 $\frac{https://github.com/lggauci/IMTAssignment2016/commit/d6ad3c9a03a6d7b03ba6c6469c}{7254c78b5d864a}$ 

# **Story Board**







# TASK 4 - (P2.1)

Identify and apply own area of expertise by listing the areas of expertise required to implement game Functionalities.

# **Graphics Artist**



Recruiter Electus Recruitment

Location Bournemouth
Salary Negotiable
Posted 24 Mar 2016
Closes 21 Apr 2016
Ref HQ00011852
Contact Alex Ronald

Clearance Level None / Undisclosed

Engineering, Information Technology

Job Type Permanent

Sector



Join a fast-growing team of talented engineers within Graphic Development. The successful candidate will be working on supporting our worldwide customer base as a Graphics Artist.

Job Description Graphics Artist

#### Job description

Creating 3D art assets for real-time simulations within the Graphics Department. Our impressive simulation hardware that comprises of projection domes, large multi-screen displays, HMI touch screens and cyber-chairs will be used to test assets you produce.

#### Key Responsibilities

The candidate must be motivated, personable and willing to work hard. Strong 3D modelling skills are a must, with an additional mix of animation and 2D abilities.

You will need to be well versed in the modelling, UV mapping, and hierarchy tools within 3ds Max software. Other skills such as Particles, Animation and Unity experience would be an advantage. Photoshop skills of a high level are also necessary for texturing and in the production of HMI\ UI assets.

Graphics Artist Location: Bournemouth Salary: DOE

Link: <a href="https://www.securityclearedjobs.com/job/801797688/graphics-artist/?TrackID=10&utm\_source=Indeed&utm\_medium=cpc&utm\_campaign=Indeed#sc=jobfeed&me=feed&cm=Indeed">https://www.securityclearedjobs.com/job/801797688/graphics-artist/?TrackID=10&utm\_source=Indeed&utm\_medium=cpc&utm\_campaign=Indeed#sc=jobfeed&me=feed&cm=Indeed</a>

# **Apprentice 3D game Artist**



what:	where:	
3d Modelling Artist	Newcastle upon Tyne	Find Jobs
ioh title, keywords or company	city or postcode	

# Apprentice 3D Games Artist (ITHH)

3AAA - Aspire Achieve Advance - Newcastle upon Tyne £495 a month - Apprenticeship

Bring the world to life by crafting fluid and responsive in-game animations, as well as compelling cinematic sequences.

Working closely with the other artists and designers you will be responsible for creating high quality assets from vehicles and weapons to characters for Unreal Engine 4. You will be skilled to model and texture quickly and efficiently, with a natural flair for PBR workflows and UE4 implementation. Good knowledge of optimisation is preferred but not essential. A strong portfolio demonstrating the wireframes and texture maps is important when applying for this role.

#### Responsibilities

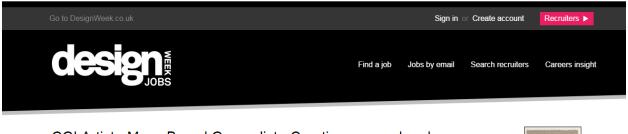
- · Model everything from characters and environments to vehicles and weapons.
- · Create and maintain a texture library.
- · Collaborate closely with designers, animators and programmers.
- · Organise, plan and manage your workloads for a streamlined pipeline.
- · Brainstorm new ideas and continue giving constructive criticism.

#### Requirements

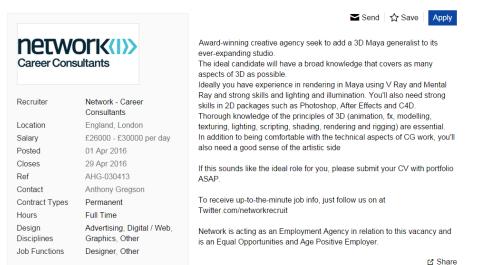
- Up to date with the latest techniques and tools for generating photoreal but optimised art in an efficient manner.
- · A strong knowledge of the following:
- Zbrush and/or Mudbox
- 3DsMax and/or Maya
- Unity and/or Unreal Engine

Link: <a href="http://www.indeed.co.uk/cmp/3aaa---Aspire-Achieve-Advance/jobs/Apprentice-Game-Artist-4051eeed0c8d16d3?q=3d+Modelling+Artist">http://www.indeed.co.uk/cmp/3aaa---Aspire-Achieve-Advance/jobs/Apprentice-Game-Artist-4051eeed0c8d16d3?q=3d+Modelling+Artist</a>

# **CGI Artist - Maya Based Generalist - Creative agency London**



# CGI Artist - Maya Based Generalist - Creative agency London



Senior Interior Designer 6 Month Fixed Term Contract (Maternity Cover)



Link: <a href="http://jobs.designweek.co.uk/job/446079/cgi-artist-maya-based-generalist-creative-agency">http://jobs.designweek.co.uk/job/446079/cgi-artist-maya-based-generalist-creative-agency</a>

london/?utm\_source=Indeed&utm\_medium=organic&utm\_campaign=Indeed

#### **Unity Developer**

Frontend Software Developer | Santa Monica, CA, United States





#### The Stuff You Need:

- . 5+ years of experience with C#
- · Shipped titles using Unity3D engine
- · Experience using version control such as Git
- · Strong interest in test driven development
- Mobile development experience
  - Experience creating systems using design pattern fundamentals
- · Self-motivation with a strong desire to learn and stay abreast of game developments/standards
- · Advanced computer science fundamentals in problem solving, object-oriented programming, and data structure

#### The Stuff You'll Do

- · Write super-clean, testable, and reusable code in C#
- · Work within Unity to build robust game systems
- · Implement systems for the games based on thorough design specifications
- · Work with other frontend developers to devise customized solutions to difficult problems
- · Be an integral part of a team focused on excellence

« Back to Current Openings

# Link:http://app.jobvite.com/CompanyJobs/Careers.aspx?k=Job&c=qWC9Vfwm&i=osDt0 fwi&s=Indeed

#### Job #16495: Developers & Programmers

#### Back to Search Apply to This Job

Title: Unity 3D Developer Date Posted: 04/18/2016 Job Location: San Jose California

Bringing Solutions Th

Take your career to the next level as a Unity 3D Developer with Resolvit. We've grown 68% over the last five years and have been included on the Inc. 5000 list of

#### Are You Up to the Challenge?

Put your Unity 3D expertise and developing skills set to work by innovating original games in a fast-paced environment. As the Unity 3D Developer you will have the

#### What You'll Need to be Successful:

- Strong knowledge of object-oriented design, design patterns, 3D math, linear algebra vector math, and associated data structures
   Expert knowledge of Unity3D and United including GUI, asset bundles, and materials/shaders
   Strong knowledge in 3D graphics development (particle systems and shaders) and performance optimization within game development

. Knowledge of working with Unity on iOS is a plus

At Resolvit, you'll be given the support you need to grow in your consulting career. In addition to opportunities for advancement and ongoing training, we offer:

- Medical, dental, and vision insurance
   Life insurance overage
   Long-term and short-term disability coverage
   401(k) retirement plan with matching
   Professional support from a dedicated Consultant Care Manager
   Networking opportunities with other consultants in your area

If you're ready to join our team, click "Apply" to register with our online talent community. Completion of our online registration process will take 5 minutes

We currently have more than 100 open career opportunities across the country, so be sure to mention the appropriate Job Code with any correspondence!

#### Apply to This Job

Powered by Bullhorn

Link:http://cls6.bullhornstaffing.com/JobBoard/Standard/BHContent\_JobDetail.cfm?jobP ostingID=16495&privateLabelID=8814&originalsource=Indeed.com&CFID=178433549& CFTOKEN=a7cb759d347de4ff-49837AA7-E600-25D7-2C713908770A71AB

# What is your favored area of expertise?

I would mostly want to work in the 3d Industry as has been one of my favorite types of art for as long as I can remember, one of the reasons I started my studies. The rewards are great, an expert in 3d design can have a very high pay, can work on many different projects and bring some fun to someone's life using the assets/ characters you created.

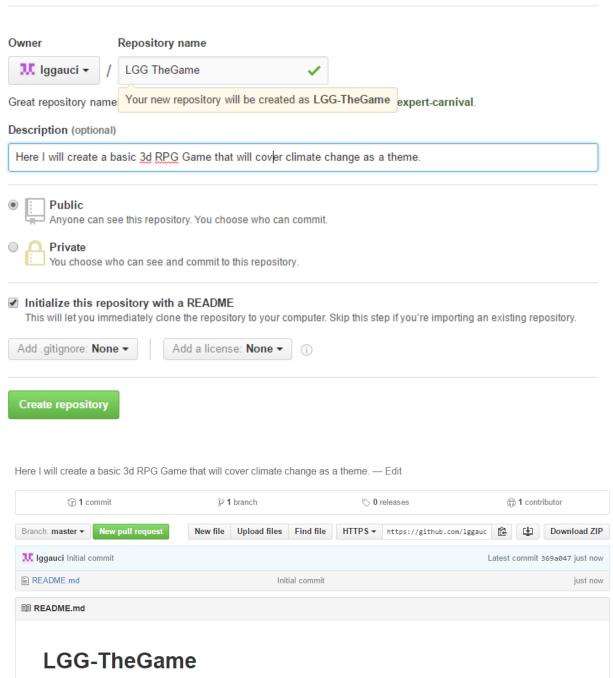
# TASK 5 - (P2.2)

I would fit in the Game Design as it would allow me to create the actual items used based of the areas discussed in the Idea generation phase. Generally, a team of concept artists would create countless of drawings which I could work off when creating/texturing/animating my 3D models. When done I could send the necessary files to the game developers to implement them in the game.

# TASK 6 - (P3.1)

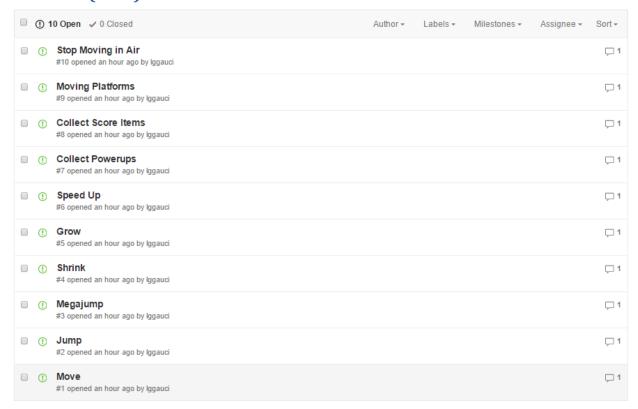
# Create a new repository

A repository contains all the files for your project, including the revision history.



Here I will create a basic 3d RPG Game that will cover climate change as a theme.

# TASK7 - (P3.2)

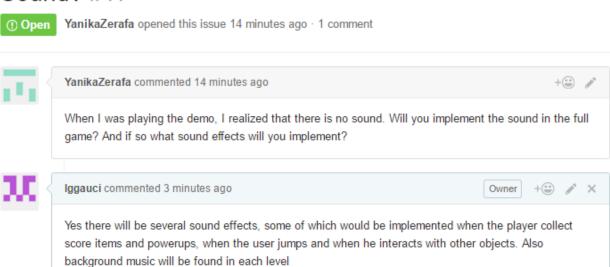


# **TASK 8 - (P3.3)**

# Megajump platform doesn't seem to work properly. #12



# Sound? #11



# I can't move in the air? #13



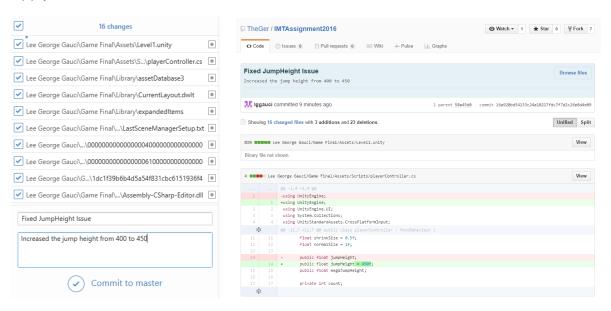
# **TASK 9 - (P4.1)**

# Create an issue, describe in full and apply bug lable



# Implement Fix in Code

# **Apply Commit**



## Reply to Issue with commit



# TASK 10 - (P4.2)

Almost all of the requirements are that the client required are met in the interactive application, some adjustments could be made to further incorporate better mechanics into the game, the powers ups (Shrink, Grow, Speedup), could have certain gateways where they can be activated so that the user would not lose the power up before passing a certain part of the level. Also the wasd work for pc based export, this will not work on tablets so I have also implemented a joystick and a button to move and jump, these can prove to be a little tedious on small screens, so it would be a good idea to export at different aspect ratios depending on the device.

# TASK 11 - (M1.1)

# What are they?

These are database servers that hold open source code. It is easily accessible by clients across all around the world. These are generally used when creating software development projects between software engineers everywhere. These can be used to create, manage, and tweak any open source software. These database servers provide an invaluable set of tools for today's world where open source software requires regular upgrades and collaboration in development.

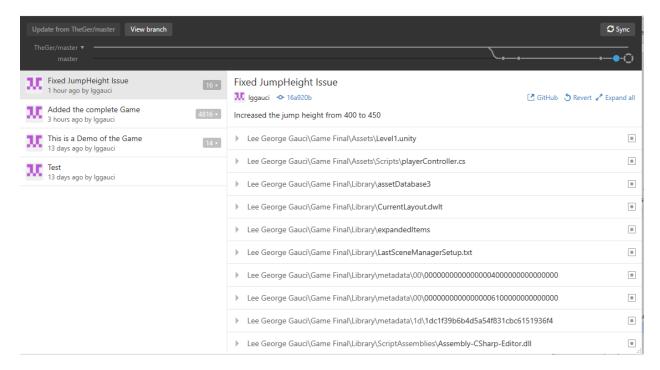
# Sourceforge vs GitHub vs GitLab

Sourceforge only allows code hosting whereas gitHub provides code hosting and code review. Source forge has a much weaker way of collaborating on projects, in gitHub you can simply fork a project and you can do what you want to it. This can prove to be a little more different in source forge as you have to contact the developer. In the end Source forge has a very bad business model which prevents it from improving, it sets users up with countless ads before they can get what they want in the first place, none of these are found in gitHub which makes for a cleaner and smoother workflow.

On the other hand, gitLab proves to be for user friendly and safe than gitHub. It has a more convenient user Interface which enables users it has access to everything from one screen (stats, groups. Latest users, users, latest projects, and projects) it also has the Snippet support function which lets users share small pieces of code from their project without actually giving the whole project. It has higher protection systems for branches to keep code safe, this way only selected people can push, forcepush and delete code in the selected branch. There is are Improved milestones which enable you to set milestones at a group level not only at a developer level, this helps as developers will see the whole teams project as a whole and not just the work they are limited to. Also there is a work in progress status that developers can use to label their projects. This helps to let collaborators know that the code is still unfinished which in turn prevents accidental merging of nonworking code.

# TASK 12 - (M2.1)

Rolling back a commit will allow you to revert the commit you have previously made. This will not remove the commit completely but rather create another commit stating that you have reverted back to the previous state of the repository



In GitHub Desktop you can find the commit you want but navigating the timeline located to the top of the application.



When found you can then click revert which is located to the top right just under the bar.



This created another commit stating that you reverted the previous commit. Now just Sync the commit and you are good to go.

# TASK 13 - (M3.1)

# **Managing Branches**

You must know beforehand how you are going to structure you branches to finally be merged to the master. If all developers have commit access to the repository they can simple pull changes from the repository, make their own changes, commit the changes on their local pc and then simply push back into the public repository.

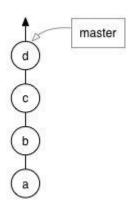
Another method is best suited for one main developer and other sub developers who are applying patches and fixes. The sub developers would clone the repo to their desktop commit the changes locally and then send a pull request to the main developer. A prime example of how this works is how I am managing my project locally and then asking for a pull request from the teacher.

There is also another method which allows you to make specific clones for working on specific parts of the repo. It will use the same method as before where the user would work locally and then post a pull request to the main repo.

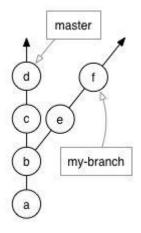
# Forking the Initial Project.

When I forked the repository from TheGer I made a complete copy of the repository which is now mine. Any changes I apply whether be it adding files or changing existing ones will only effect my version of the repository this created a branch. Now the whole class has to fork this project creating a branch for each student all of which have their own changes and work locally. After each student makes the pull request TheGer can choose which branches to merge to the master repo with the help of complete details and descriptions of how and what changes were made to the original repo.

# TASK 14 - (D1.1)

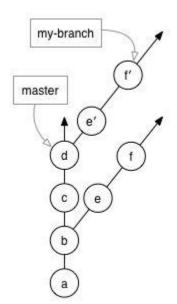


The best way to understand rebasing in Git is to understand how the whole system works. Repository's has a tree structure where the circles are commits applied. In this case there are 4 a,b,c and d respectively. The latest commit (d) is considered to be the head of the master branch.



To the left we see that there is the previous master branch and also my-branch. Both of these branches contain the commits from a and b but when they are separated my-branch gained e and f whilst the master branch gained c and d. As b is connected to both branches it is considered to be the "base" of the workflow.

Now you would think that my-branch is missing important information which is found in c and d so you would like to merge them, this could lead to weird merge commits and make reviewing pull requests much more difficult.



This is where rebasing comes in. Rebasing finds all the commits between the base and the head, in this case e and f and re-commits them on the head of the branch you are rebasing into which in this case is the master. Git creates these as new commits which can be seen as e' and f'. Your previous commits will still be left untouched and if something doesn't go according to plan you can always revert back to the original commits.

# TASK 15 - (D3.1)

# **Forked Project:**

https://github.com/gabrielecirulli/2048



```
<!DOCTYPE html>
<head>
         <title>2048</title>
       <link href="style/main.css" rel="stylesheet" type="text/css">
<link rel="shortcut icon" href="favicon.ico">
<link rel="apple-touch-icon" href="meta/apple-touch-icon.png">
<link rel="apple-touch-startup-image" href="meta/apple-touch-startup-image-640x1096.png" media:
<link rel="apple-touch-startup-image" href="meta/apple-touch-startup-image-640x200.png" media:
<meta name="apple-mobile-web-app-capable" content="yes">
</meta name="apple-mobile-web-app-capable" content="yes">
<meta name="apple-mobile-web-app-capable" content="yes">
</meta name="apple-mobile-web-app-capable" content="yes"
</me
         <meta name="apple-mobile-web-app-status-bar-style" content="black">
         <meta name="HandheldFriendly" content="True">
<meta name="MobileOptimized" content="320">
<meta name="viewport" content="width=device-width, target-densitydpi=160dpi, initial-scale=1.0;</pre>
</head>
         <div class="container">
                          ch1 class="title">2048</h1>

ch1 class="title">2048</h1>

ch2 class="scores-container">0</div>

cdiv class="score-container">0</div>

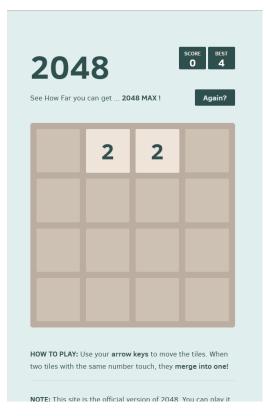
cdiv class="best-container">0</div>

                            </div>
                  <div class="above-game">
                       Join the numbers and get to the <strong>2048 tile!</strong>
<a class="restart-button">New Game</a>
                <div class="game-container">
     <div class="game-message">

<div class="lower">
                                              <a class="keep-playing-button">Keep going</a>
<a class="retry-button">Try again</a>
                            </div>
```

# **Unmodified project:**

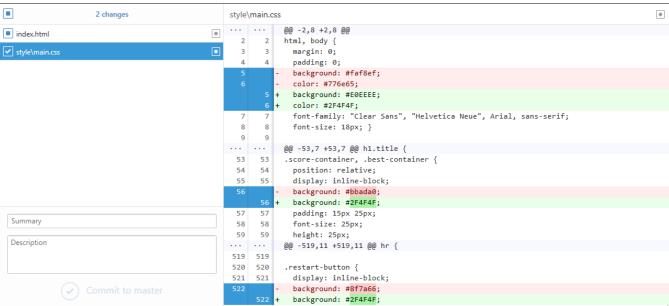
# **Modified project**

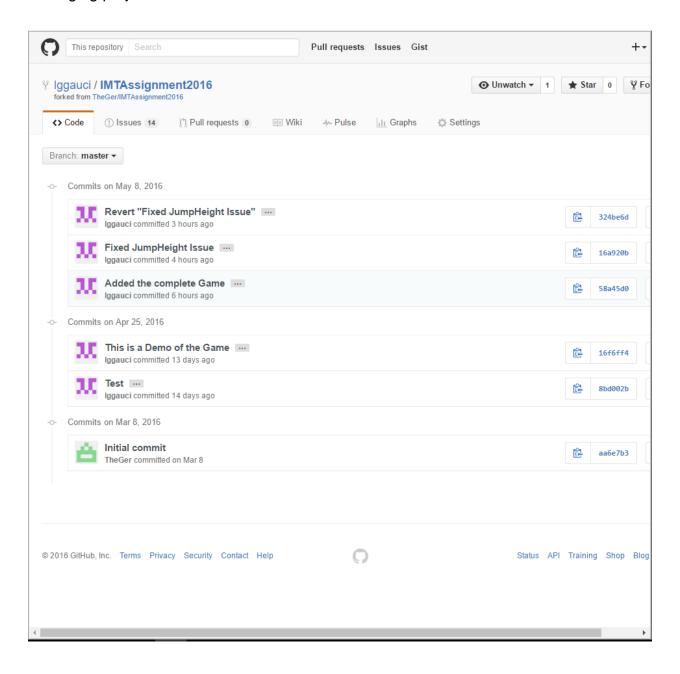


I changed the background colour of the buttons and the page itself, I also changed the colour of the text.

I also changed the words located at the top under 2048 from "Join the numbers and get to the <strong>2048 tile" to "See How Far you can get ... <strong> 2048 MAX!"

TASK 16 - (D2.1)





# **Bibliography**

http://stackoverflow.com/questions/2715085/rebasing-and-what-does-one-mean-by-rebasing-pushed-commits

 $\underline{http://programmers.stackexchange.com/questions/156519/managing-multiple-peopleworking-on-a-project-with-git}$ 

https://guides.github.com/activities/hello-world/