

Contents

[Task 1 – P 1.1 5](#_Toc450983707)

[Player Jump 5](#_Toc450983708)

[Player Move left 5](#_Toc450983709)

[Player Move right 5](#_Toc450983710)

[Keyboard Control 5](#_Toc450983711)

[Console Control 5](#_Toc450983712)

[Mega size 5](#_Toc450983713)

[Mini size 5](#_Toc450983714)

[Super Jump 5](#_Toc450983715)

[Obstacles 5](#_Toc450983716)

[Animation on trigger 6](#_Toc450983717)

[Task 2 – 1.2 6](#_Toc450983718)

[Repository 6](#_Toc450983719)

[Commit 6](#_Toc450983720)

[Issue 6](#_Toc450983721)

[Sync 6](#_Toc450983722)

[Add 7](#_Toc450983723)

[Pull request 7](#_Toc450983724)

[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. 7](#_Toc450983725)

[Task 3 – (P1.3) 8](#_Toc450983726)

[Task 4 – (P2.1) 10](#_Toc450983727)

[Game Developer/Programmer 11](#_Toc450983728)

[Senior Designer 12](#_Toc450983729)

[3D Artist 13](#_Toc450983730)

[UI/UX Designer: 14](#_Toc450983731)

[My preference: 15](#_Toc450983732)

[Task 5 – (P2.2) 16](#_Toc450983733)

[Task 6 – (P3.1) 17](#_Toc450983734)

[Task 7 – (P3.2) 19](#_Toc450983735)

[Task 8 – (P3.3) 21](#_Toc450983736)

[Task 9 – (P4.1) 23](#_Toc450983737)

[Task 10 – (P4.2) 27](#_Toc450983738)

[Task 11 (M1.1) 28](#_Toc450983739)

[Task 12 – (M2.1) 29](#_Toc450983740)

[Task 13 – (M3.1) 30](#_Toc450983741)

[Task 14 – (D1.1) 31](#_Toc450983742)

[Task 15 – (D3.1) 32](#_Toc450983743)

[Task 16 – (D2.1) 34](#_Toc450983744)

[Bibliography 35](#_Toc450983745)

[Websites: 35](#_Toc450983746)

# Task 1 – P 1.1

Identify client requirements by listing the features you would like to implement in your software project. You need to identify at least 10 functionalities to be implemented. To achieve this task, include a list of 10 functionalities that were implemented as part of your CIDP assignment, or mention 10 functionalities from a software project of your choice.

### Player Jump

The player can jump to avoid obstacles either by scroll clicking, space button or clicking on the up button on the on screen console.

### Player Move left

The player can move leftwards until he hits the barriers, which are different for each level. Obstacles or collectibles spawn up until this point as well. This can be done by clicking the left button or pressing the left arrow on your keyboard.

### Player Move right

The player can move rightwards until he hits the barriers, which are different for each level. Obstacles or collectibles spawn up until this point as well. This can be done by clicking the right button or pressing the right arrow on your keyboard.

### Keyboard Control

The game can be played using the left and right arrows and space bar on your keyboard.

### Console Control

The game can also be played by clicking on the console which is displayed on screen.

### Mega size

One feature of the game is the ability of increase the size of the player by picking up the mega size power up (the red rotating cube). This feature lasts for a limited amount of time, after which the player will restore the size it was before the power up was collected.

### Mini size

Another power up is the mini size (blue rotating cube). This power up works in the same way as the mega size, however on the contrary of the mega size, it decreases the player’s size.

### Super Jump

The super jump gives the ability to the player to jump higher than the normal jump. This power up also has a limited amount of time, after which the original jump height is restored.

### Obstacles

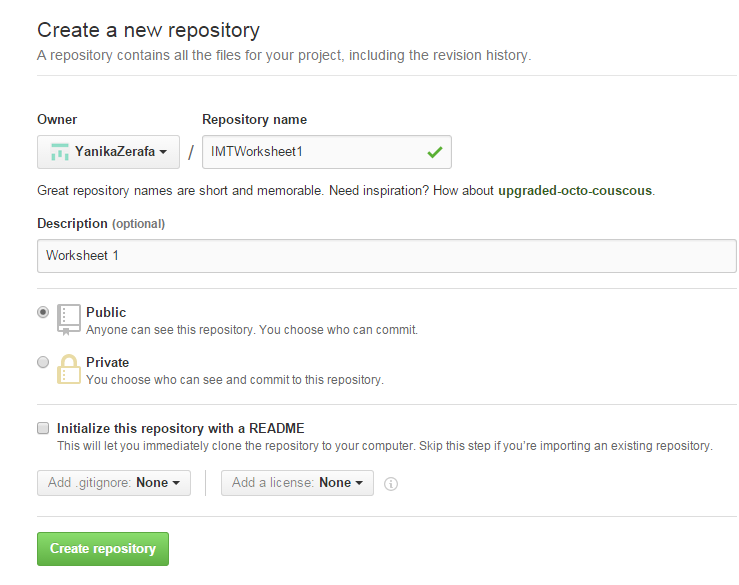
The objective of the game is to avoid the obstacles to stay alive and collect cans to clean the environment.

### Animation on trigger

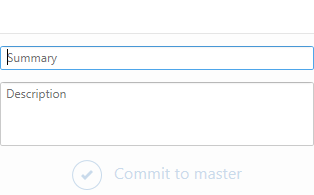
3D models are animated when the player collects power ups. The models are 3D text and are different for each powerup.

# Task 2 – 1.2

Define and analyse target group to identify user needs by familiarizing yourself with the communication tools available in Github. To do this, please define the following terms in the context of git: Repository, Commit, Issue, Sync, Add & Pull request. 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.

Repository - A repository is a ‘folder’ that contains your projects. It also stores the changes made on the files in it. Repositories can be viewed by public or can be kept private.

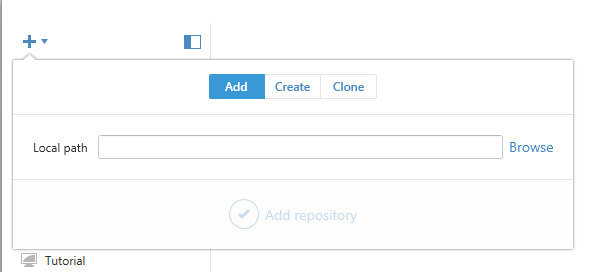
Commit - Commit means when you change something in a file and the GitHub records the change. The file is then updated upon commitment and a description of what is changed is usually kept.



Issue – Issues in GitHub is a way keeping track of any problems that are encountered in the project. This section is accessible by the whole team working on the project. Using issues you can keep track of

Sync - Git Sync is when you synchronize the files from the desktop app so that every update is recorded and amended online as well.

Add – Add can refer to two different things. In the Desktop Application, add is when you have a folder which is already a repository on your pc, and you find its path through the desktop app so you can continue working on it as shown below.



Secondly, there is the ‘Add Command’. This command is used to update the index, a structure tree of the workflow.

Pull request – This feature allows the participants of a project to ask for changes. Instead of amending the changes right away, these have to be accepted and

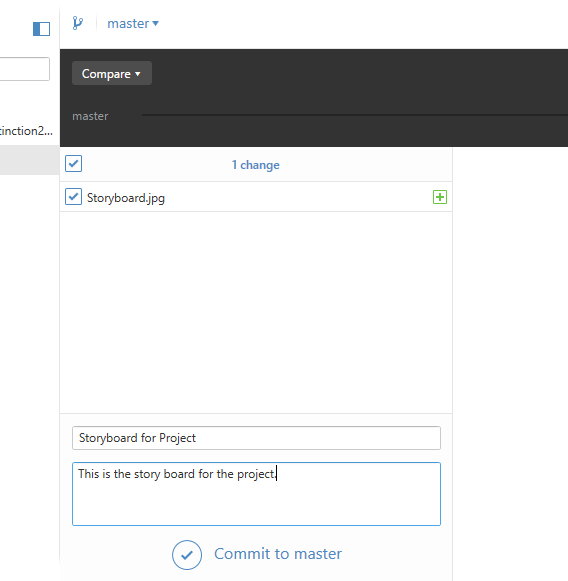
### Once you have defined the above, explain which one of the above features would be useful to create a list of requirements as requested by users.

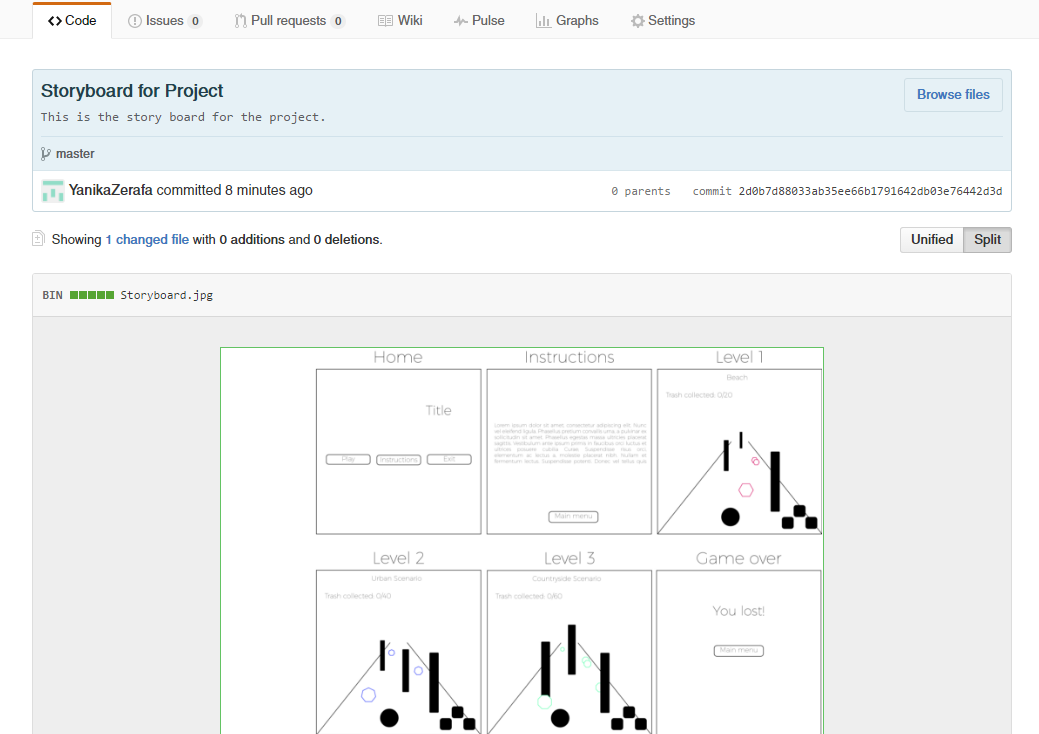
Issues in Github are the best way for users to communicate and submit comments about any questions or requirements they would have about the project. This system easy for both viewer and owner to work around with and it is also very organized.

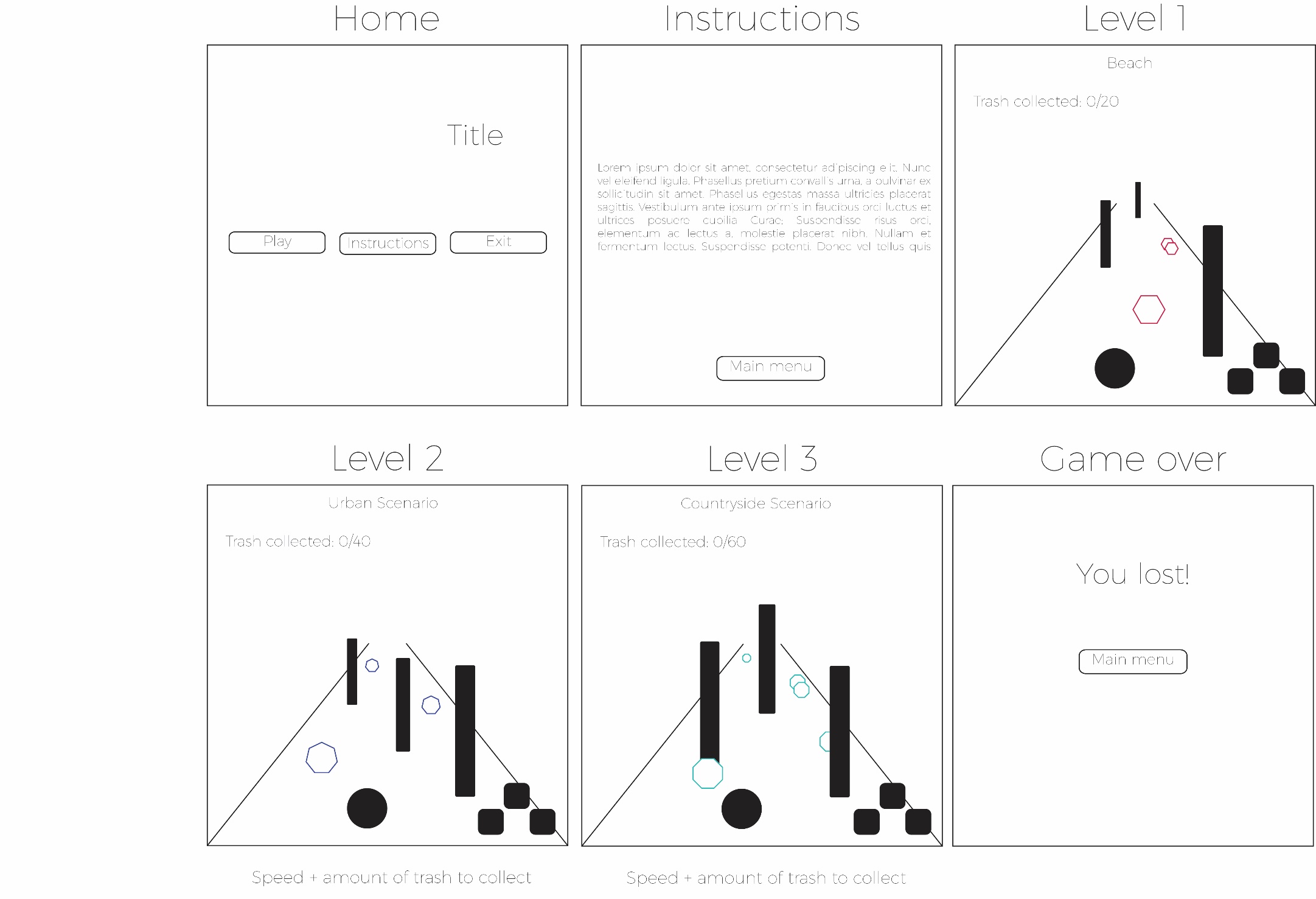
# Task 3 – (P1.3)

Clarify your creative intentions by creating a clear storyboard for your entire application. This should be a sketch of each screen in the application. You may use any authoring tool to build these screens. Save this work in a ‘storyboard’ folder in your project. To achieve this task, you must include the following items:

* A printed storyboard, which clearly shows the sequence of events in the application you wish to develop in your documentation
* A link to the relevant commit where you uploaded this storyboard on your Github page.

Link: <https://github.com/YanikaZerafa/GithubAssignment/commit/2d0b7d88033ab35ee66b1791642db03e76442d3d>





# Task 4 – (P2.1)

Identify and apply own area of expertise by listing the areas of expertise required to implement game functionalities. Following are some examples:

* For a game developer job, programming knowledge is required
* For a game artist job, knowledge of photoshop and how to create a sprite sheet is required
* For a sound engineering job, knowledge of sound editors such as audacity is required.

Find job offers related to the different areas of expertise on the Internet/classified ads. Include a screenshot of one job offer per area of expertise, with a sentence justifying your choice. For this task you must include at least 3 different job offers. What is your favoured area of expertise? Mention this in your task with a short paragraph justifying your choice.

Links (screenshots to follow):

Game Programmer:

<http://jobs.gamasutra.com/job/gameplay-programmer-durham-north-carolina-30187>

3D Artist:

<http://jobs.gamasutra.com/job/senior-environment-artist-burbank-california-29937>

Senior Designer:

<http://jobs.gamasutra.com/job/senior-designer-burbank-california-29908>

UI/UX Designer:

<https://career4.successfactors.com/career?career_ns=job_listing&company=EA&navBarLevel=JOB_SEARCH&rcm_site_locale=en_US&career_job_req_id=68202&selected_lang=en_US&jobAlertController_jobAlertId=&jobAlertController_jobAlertName=&_s.crb=wqtd1ZMRDlnjcI8V8t3qdESsRl4%3d>

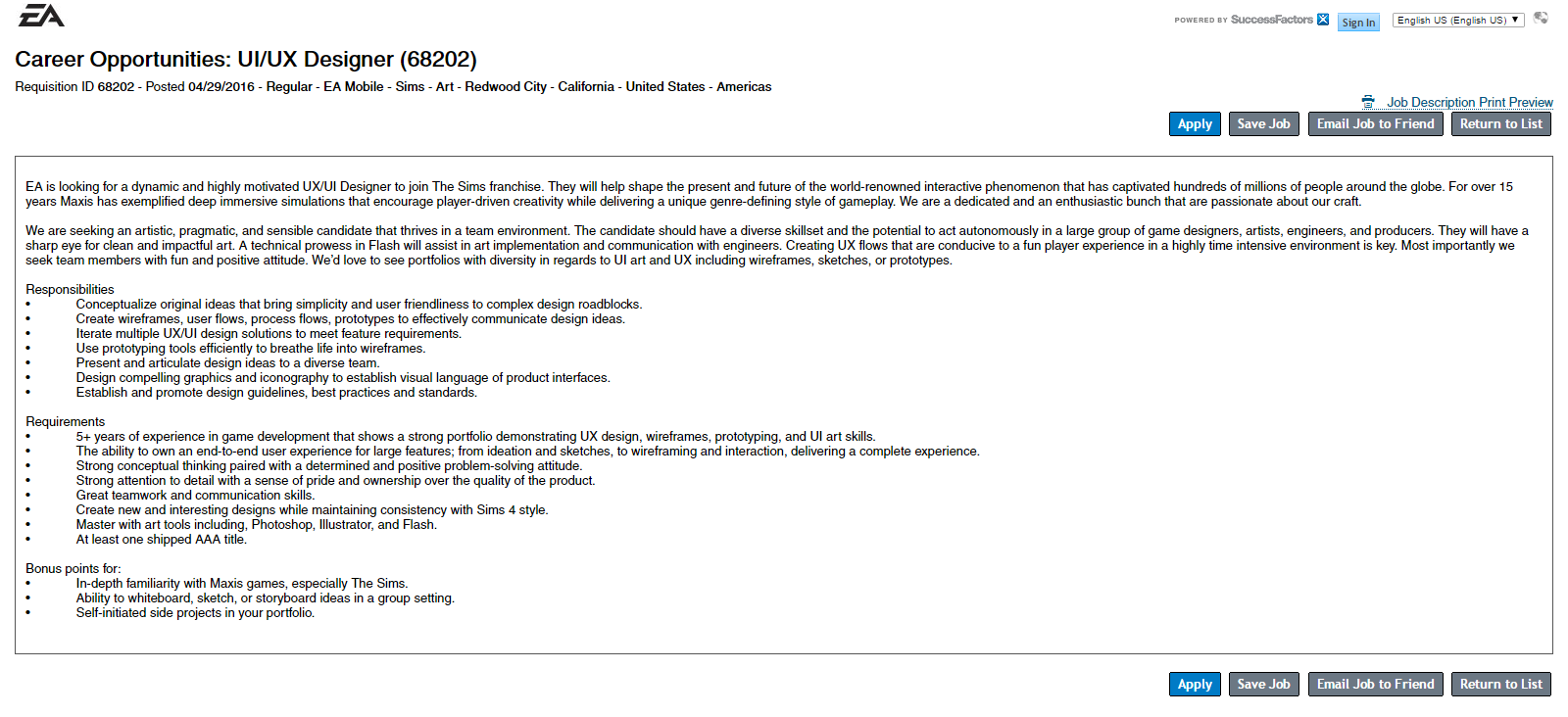
### C:\Users\yzera\Desktop\programmer.jpgGame Developer/Programmer

### Senior Designer



### C:\Users\yzera\Desktop\3d artist.jpg3D Artist

### UI/UX Designer:



### My preference:

My personal preference is UI/UX design. This would enable me to create assets and the navigational experience to be used for the game. I have recently developed the liking for this sector when I started to understand that this job is much more complex than it looks and I started admiring the works of other designers for various games.

# Task 5 – (P2.2)

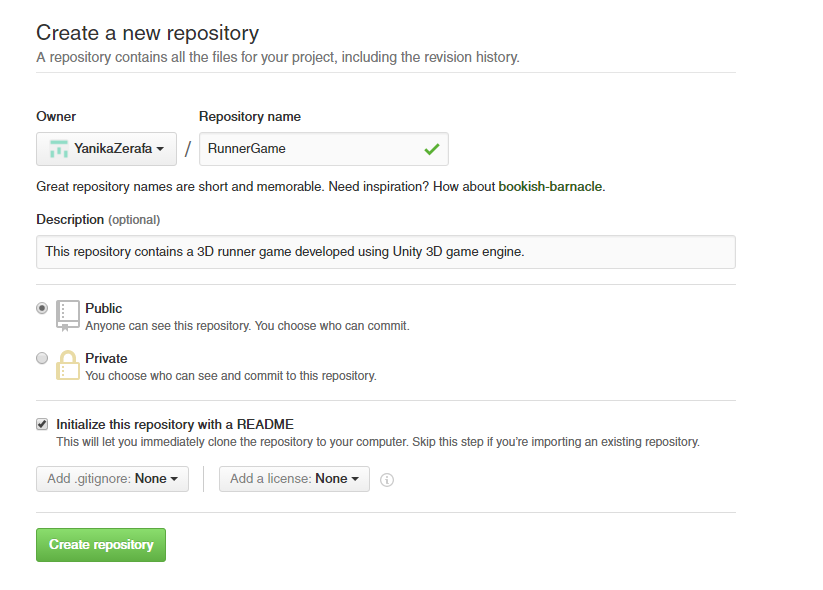
Clarify your own role in the team driven development schedule by explaining how your own area of expertise in game development may be applied to the following phases of game development. Write a paragraph about your role in each of the following phases:

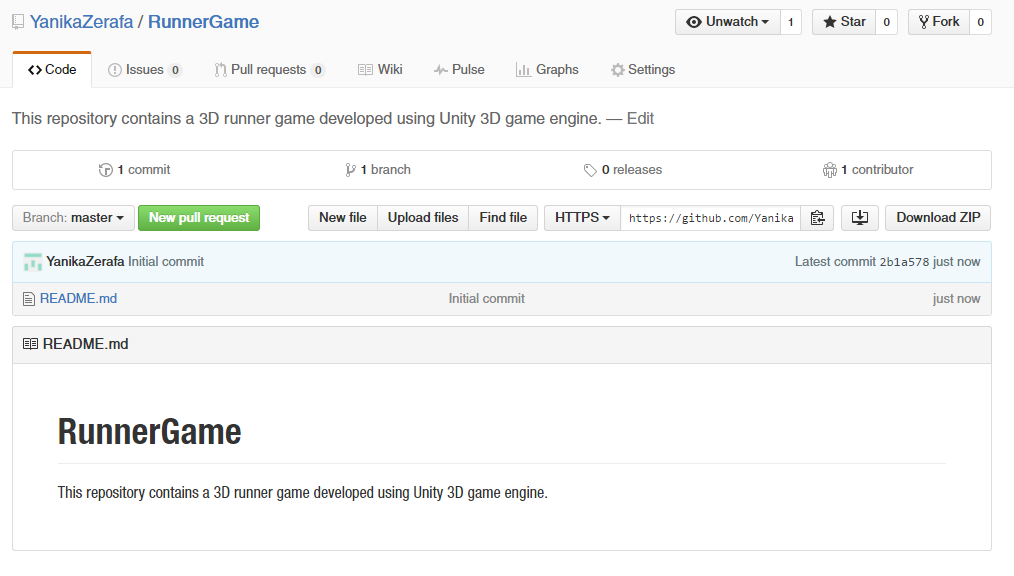
* Idea generation.
* Storyboarding and game design.
* Creation of the game design document.
* Implementation of functionalities.
* Deployment and support of the game.

During the idea generation process, a UI/UX designer would start coming up with various possible solutions for design styles, that depends on the game approach that the team is going for. This is achieved after some intense research and brainstorming sessions. Later comes the part where the designer starts mapping out the navigational system of the game and how various assets have to be shown on screen. This is a crucial part as this sets a base later upon which developers later have to build on. Afterwards, the designer should start creating the assets for the UI according to the graphic style chosen for the game. Once the assets and the storyboarding are finished, the elements should pass to a developer for him/her to implement them in the actual game. Once they are implemented, the UI designer should pay attention to the testing phase results carried out on the game, especially the UI navigational part. If required, any changes should be implemented during this stage. Assets created should also be shared with any graphic designers and web developers so as to create the appropriate promotional material for the game.

# Task 6 – (P3.1)

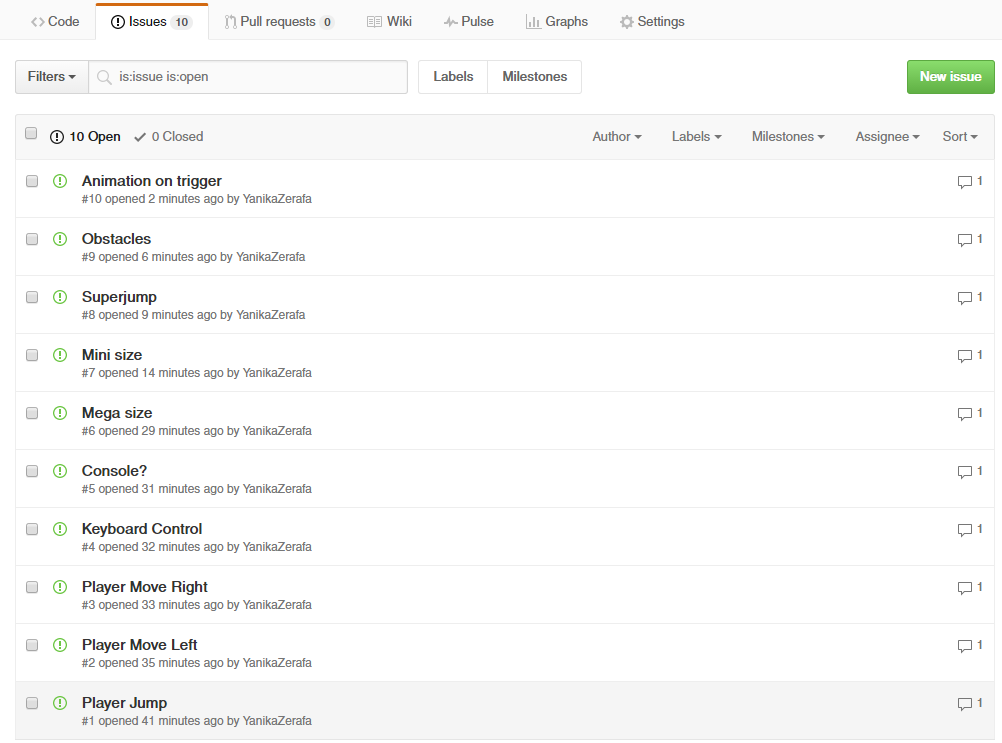
Produce preliminary concepts for an initial prototype by forking a new project from the following repository: <https://github.com/TheGer/IMTAssignment2016> on http://www.github.com, and writing a full description of your intentions for the project in the project description screen as shown below:

Include a similar screenshot with the project description filled in and the following settings set up. Your project will be the area where you will be working on your project step by step when it comes to developing the game and adding files to it.



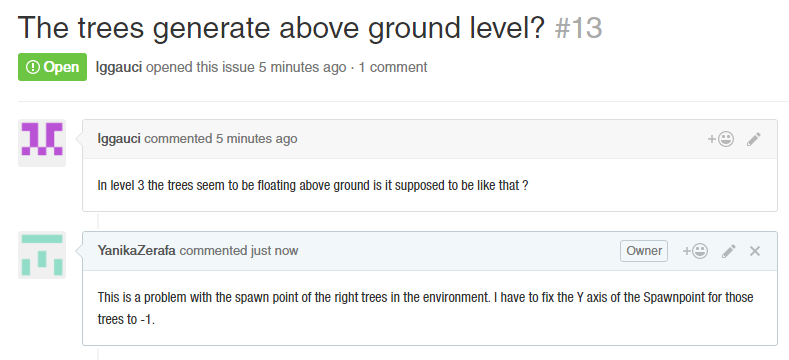
# Task 7 – (P3.2)

Evaluate and confirm the prototype in relation to constraints by listing the client requirements as mentioned in the brief as issues in your github project. Each client requirement will be one issue as may be seen in the following screenshot. Reply to each issue detailing how the issue could be implemented. You are required to fill in 3 issues.



# Task 8 – (P3.3)

Reflect and record on feedback from prototype phases by showing a demo copy of the game/software to

another student in your class. That student needs to post at least 3 questions about your game on your github page, and you need to respond to those questions on the issue tracker. Add a screenshot of the questions and responses to your report.



# Task 9 – (P4.1)

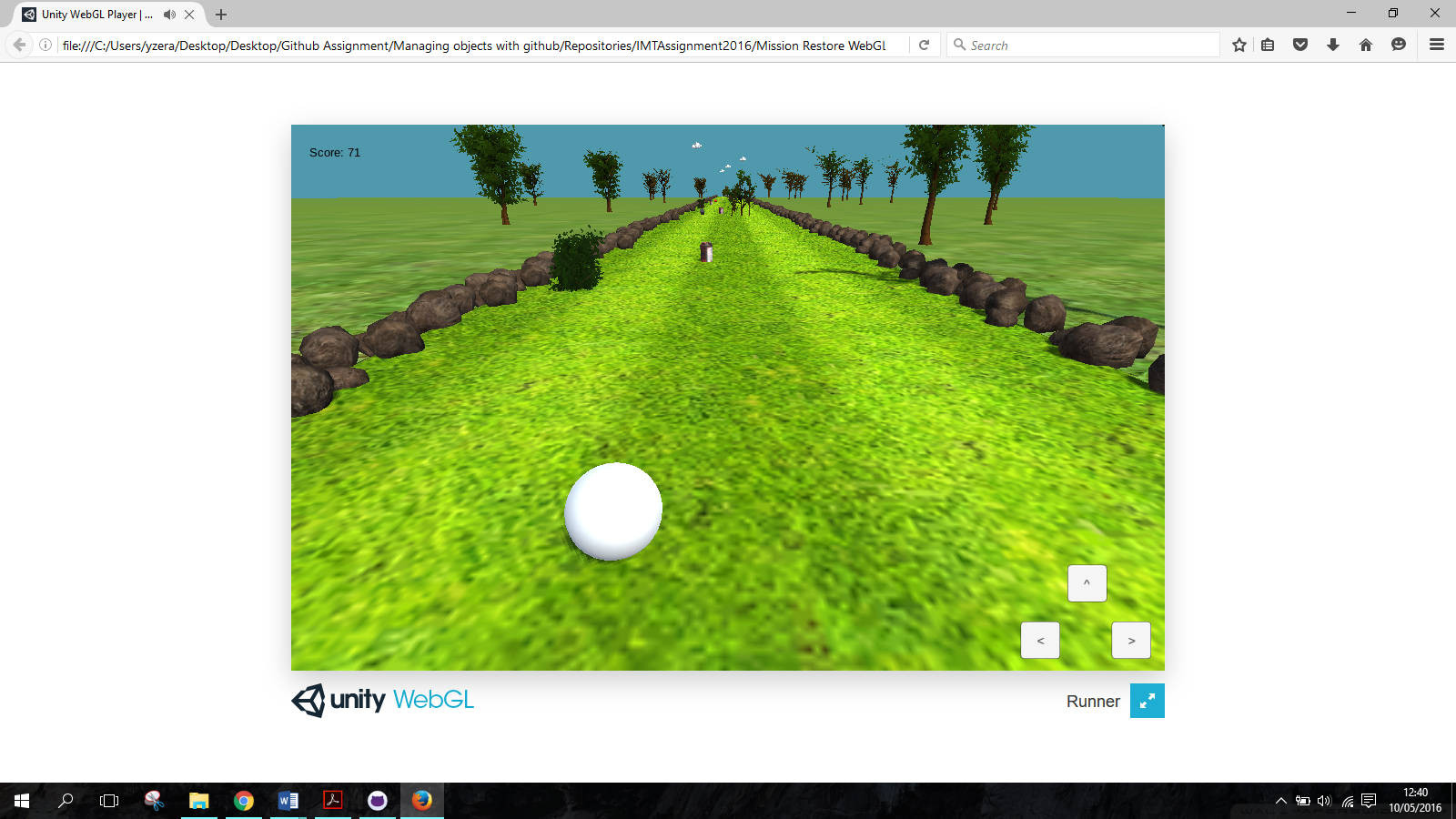
Develop a fully working interactive media product that meets client needs by showing the following step by step process using screenshots:

* Create an issue (bug) as a client
* Describe the issue in full
* Implement a fix in your code
* Upload a new commit on Github

Reply to the issue you posted with reference to your new Git commit.

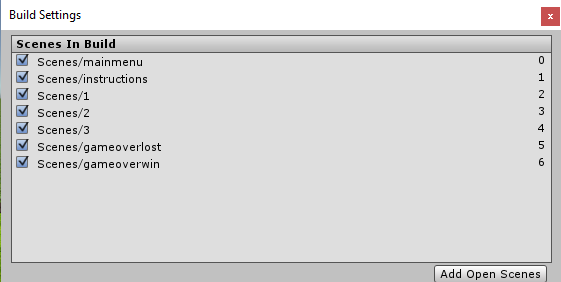


Problem:

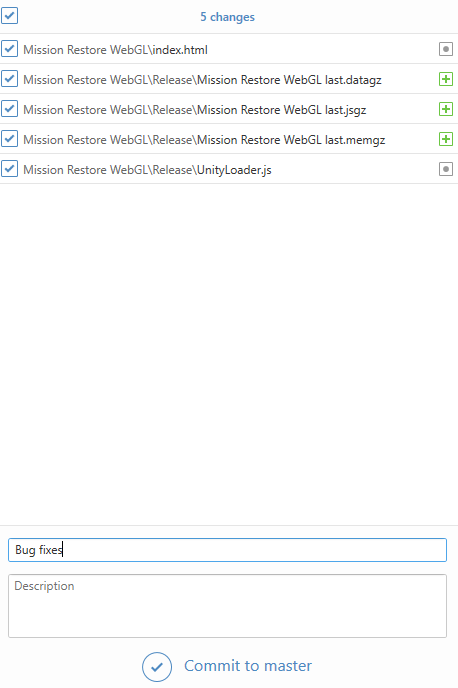
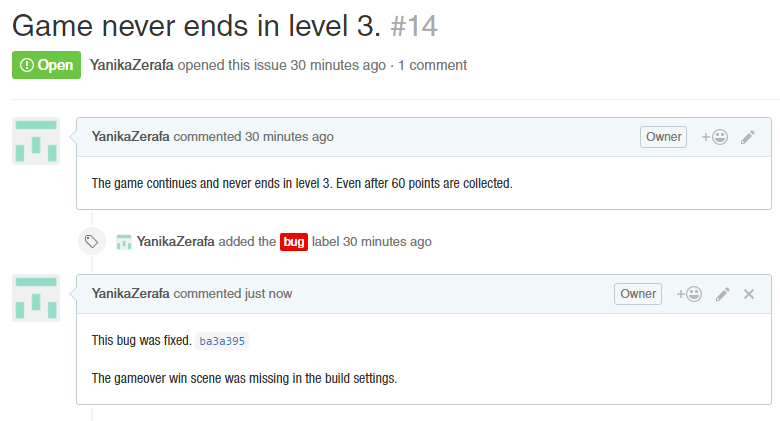


The bug stands in the last level. The game keeps on going and never stops, even when 60 points are exceeded.

Solution:

The code was good and had no errors. The problem was that the scene that should have loaded wasn’t added to the build settings.

Commit:

https://github.com/YanikaZerafa/IMTAssignment2016/commit/ba3a3955bad412e0064d80371a5bb8d8760b9aac

# Task 10 – (P4.2)

Evaluate and record interactive media outcomes against the constraints and requirements of the brief by discussing in a brief paragraph what limitations your interactive media product has in relation to the initial requirements outlined as issues by the client.

All of the requirements requested by the client were met in the game. However, some improvements can still be applied so as to have the game look more professional and complete. Additional animations can be implemented when the player gets a power up, for example an animation to increase the size of the player gradually, not as it is currently in the game. The same counts for the rest of the power ups.

A limitation I encountered during the animation phase was that the 3D models I was importing from Maya, were becoming partially transparent. The problem was from Maya’s side and since I wasn’t very familiar with the settings I had to set when exporting the fbx file, I had to work my way around it. I had to rotate some of the letters or maybe change the words that I used. For example the model for the mega size, originally had to be ‘Super size’.

Another limitation is the console I implemented in the game so that I can be played on smartphones. Playing the game with the console is much more difficult than using the keyboard. Even though it is still playable and the player can still win, the process is much longer and more tedious.

# Task 11 (M1.1)

Show that effective judgments have been made by finding out about systems which are similar to Git. Explain what these systems are and the one basic difference between Git and these other systems. Write a paragraph explaining the basic differences between Git and at least two other concurrent version control systems.

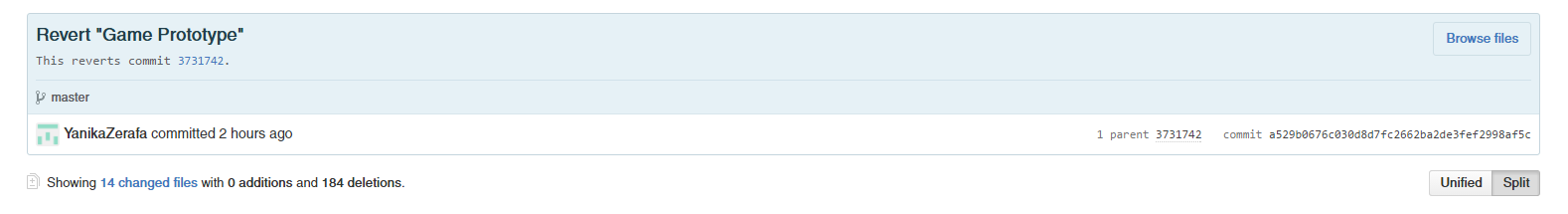
Apart from Github, various other source code hosting websites exist. These include Bitbucket, Sourceforge, Gitlab, Kiln, Codeplane, CodePlex & Beanstalk. However, the major 3 source code hosting applications are Github, Bitbucket and Sourceforge.

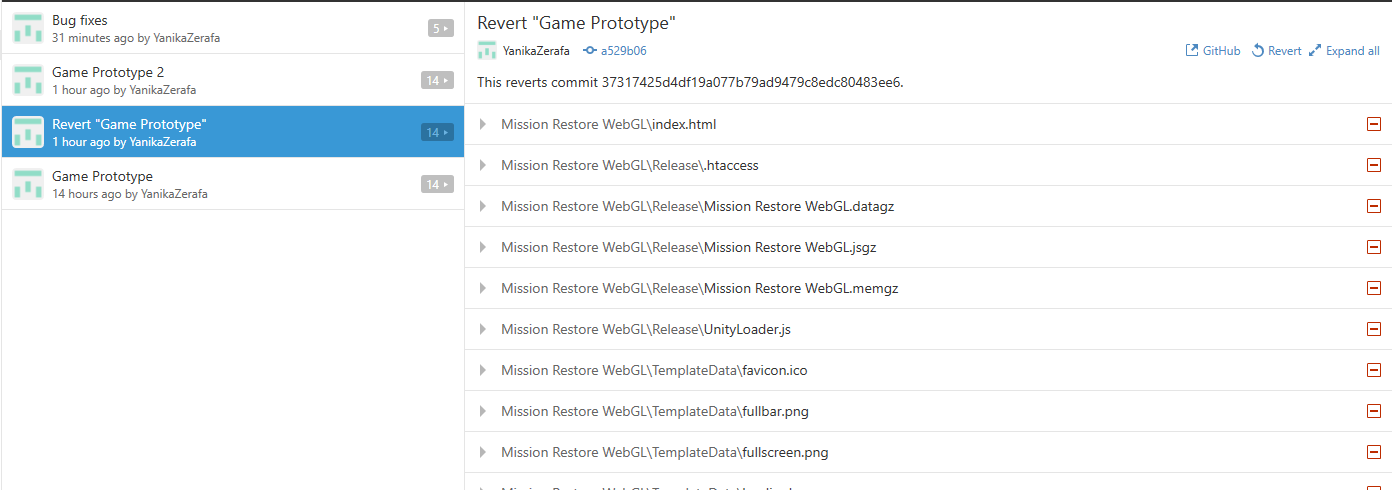
All of the three are free to use, or at least there are free versions accessible to the public. Github and Bitbucket offer unlimited space for public repositories but Sourceforge isn’t unlimited. Repositories in Github have to be up to 1gb but Bitbucket allows up to 2gb. Ultimately Sourceforge is totally unlimited and has no storage limit per repository. Furthermore, for private repositories, you have to pay on Github but not on Bitbucket. Pricing systems on the Github and Bitbucket are different as Github charges their clients by every private repository while Bitbucket charges the client by collaborator on a project. Therefore, the two cannot be compared when it comes to this aspect. An advantage Sourceforge has over the other two applications is that it has discussion forums. This enables the users to discuss any possible problems in an easy user friendly way. On the other hand, you can’t use the fork facility that Github has. Among the three, only Github offers network graphs.

# Task 12 – (M2.1)

Show that relevant theories and techniques have been applied by explaining the concept of rolling back a commit. Explain what happens when a commit is rolled back and why this would happen in detail with screenshots of an example rollback and the effect on the saved code. At least one commit must be rolled back and reverted.

Rolling back a commit means that you make use of the revert function in Github desktop. What Github does when a commit is reverted is to restore any changes that had been implemented in the selected commit. For example, I had committed an early prototype of the webgl version of the game and I needed to take it back. This meant that all of the files had to be deleted since before that commit, there was nothing in the repository.

Reverted Commit: https://github.com/YanikaZerafa/IMTAssignment2016/commit/a529b0676c030d8d7fc2662ba2de3fef2998af5c



# Task 13 – (M3.1)

Show that the appropriate structure has been used by explaining how best to maintain multiple versions of the same code in git branches. Explain how the fork that you took from the initial project is a branch of the initial project.

Branches in Github are used to organize your workflow. For example, you want to try out something and change a part of the project, you can create a new branch. This way the master branch won’t be affected by the changes and thus you will still have an original copy. This is an ideal method when working in a team. Each team member can have a branch of his/her own on which he or she can work on without modifying the master branch. Then, a pull request can be submitted to the master so that then the branches can be merged if the request is accepted.

For this assignment, I forked the repository found at the following location:

<https://github.com/TheGer/IMTAssignment2016>

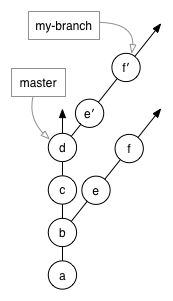
I started applying changes to the project by adding the prototype of the game and the eventual issues and commits mentioned previously in this documentation. Ultimately, I will submit a pull request when I’m finished and I’m pleased with the project. This way, the teacher will be able to accept or deny the request and maybe merge the branches.

# Task 14 – (D1.1)

Show that conclusions have been arrived at through synthesis of ideas and have been justified by explaining the concept of rebasing in Git. Explain how doing a rebase off a different commit can be used to update the current branch to the changes in the initial branch.

Git rebase is a command that permits the user to modify the history of commits a project has. This function is useful when it comes to forking a project and updating it. Here is the breakdown.

First of all, a master branch can be forked at a certain stage. Now let’s say that after the fork, the master branch has a couple of new commits. The forked branch wouldn’t be up to date with these new commits, hence the rebase function. It finds all the commits done between the forking stage and now and updates the forked branch and creates new commits based on the last updated master branch.



First Original fork.

Master branch. Every letter represents a commit. A is the first commit, D is the last one.

New commits after rebase.

Last two commits that were not part of the first fork.

Image taken from: <https://github.com/edx/edx-platform/wiki/How-to-Rebase-a-Pull-Request>

# Task 15 – (D3.1)

Show that effective thinking has been used in unfamiliar contexts by forking and modifying an existing Unity project on Github. Find a project which has code that you can understand, fork and modify the code and comment your modifications. Show screenshots of the modified project with your additional commit and explain what changes you carried out to the project (eg. Change of button text)

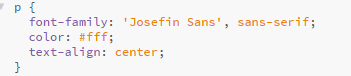
I forked the following project: <https://github.com/YanikaZerafa/0hh1>

Game playable at: <http://0hh1.com/>

Original vs Modified:



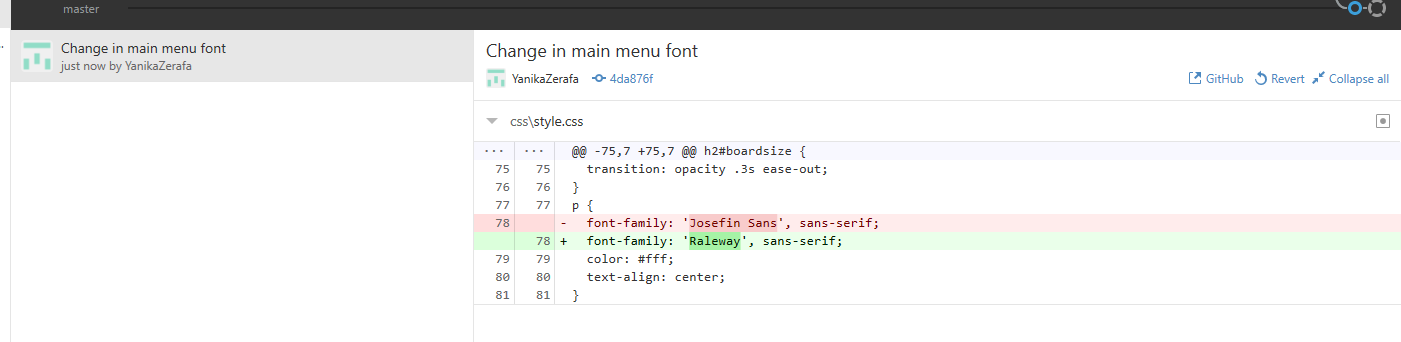
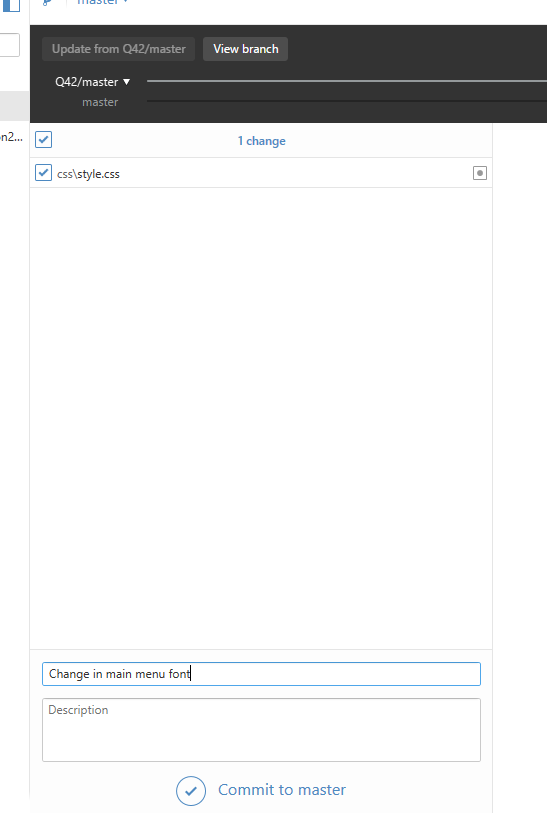
Original vs Modified code:



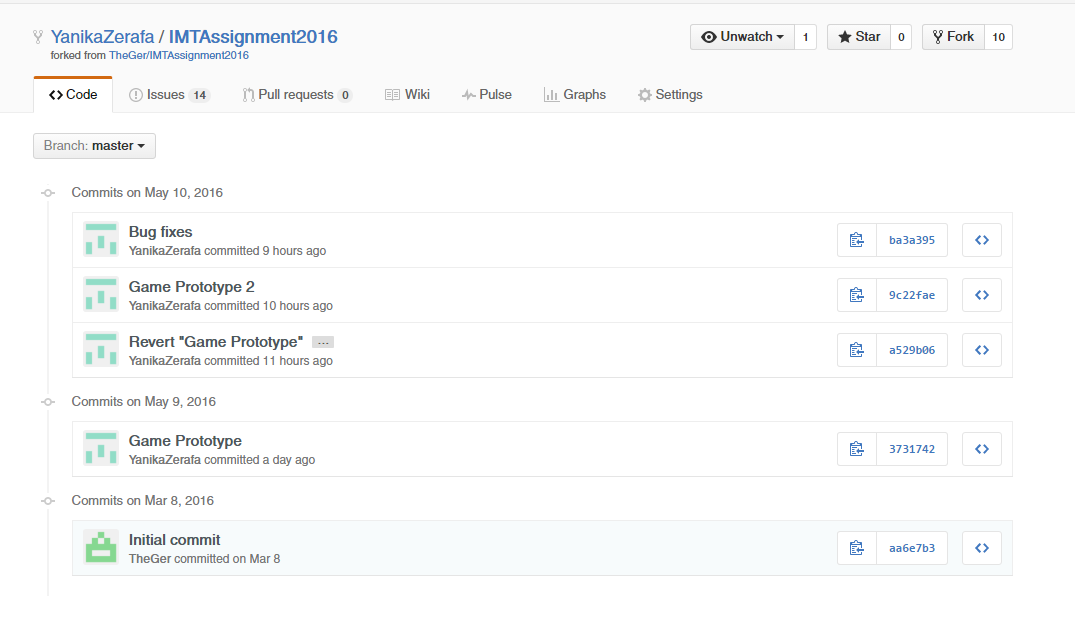
I modified the typeface of the mainmenu.

Evidence of commit:

<https://github.com/YanikaZerafa/0hh1/commit/4da876faa48f089c3d3d977e9308d72bd1a65872>



# Task 16 – (D2.1)

Show that substantial activities have been planned, managed and organized by including a screenshot of your git project with a clear timeline of commits from the beginning of the assignment until the deadline date.

# Bibliography

### Websites:

Guides.github.com. (2016). Mastering Issues · GitHub Guides. [online] Available at: https://guides.github.com/features/issues/ [Accessed 18 Apr. 2016].

GitHub. (2016). *How to Rebase a Pull Request*. [online] Available at: https://github.com/edx/edx-platform/wiki/How-to-Rebase-a-Pull-Request [Accessed 11 May 2016].

Mysore, A. (2015). *GitHub Alternative: Top 7 Best Source Code Hosting Sites*. [online] Beebom. Available at: http://beebom.com/github-alternatives/ [Accessed 10 May 2016].

Upguard.com. (2016). *Github vs Bitbucket*. [online] Available at: https://www.upguard.com/articles/github-vs-bitbucket [Accessed 12 May 2016].

Anon, (2016). [online] Available at: https://www.quora.com/What-are-the-differences-between-BitBucket-and-GitHub [Accessed 12 May 2016].

Gist. (2016). *Git Wars: GitHub vs Bitbucket*. [online] Available at: https://gist.github.com/foogit/8410710 [Accessed 14 May 2016].

GitHub. (2014). *Web games*. [online] Available at: https://github.com/showcases/web-games [Accessed 14 May 2016].