Ryerson University

Introduction to Software Engineering

(CCPS406 – Fall 2018)

Project Status Report#5

Project Analysis

Prepared for: Professor Nawar Al Hakeem

Prepared by: Fahad Jamil

Yashodhar Dave

Meet Pandya

Benjamin Travaglini

Contents

[Chapter A : SDLC Activities 5](#_Toc529556309)

[1. Proposal 5](#_Toc529556310)

[2. Background 5](#_Toc529556311)

[3. Time Schedule 5](#_Toc529556312)

[A. Work Breakdown Structure 5](#_Toc529556313)

[B. Initial Timeline at the start on Sep 28th, 2018: 7](#_Toc529556314)

[C. Timeline on Nov 9, 2018 9](#_Toc529556315)

[4. Business Case 10](#_Toc529556316)

[8. Reasons and Motivation behind the proposed project: 10](#_Toc529556317)

[D. Value Added by this product 10](#_Toc529556318)

[E. Intended Audience 10](#_Toc529556319)

[F. Targeted Market 11](#_Toc529556320)

[G. Other features: Optionally, we would paint/draw the snake objects as opposed to use the ASCII block characters to form the snake shape. 11](#_Toc529556321)

[9. Market Survey of competition for similar products 12](#_Toc529556322)

[A. Competitive Products Reviewed 13](#_Toc529556323)

[i) **Lego Pac-Man Snake Game :** 13](#_Toc529556324)

[ Pros/Features: Colorful 13](#_Toc529556325)

[ Cons/Issues: Initial access to the internet must be available 13](#_Toc529556326)

[ Popularity/Market Share: There is a Silly Snake 2.0 but does not seem appealing. There is an alternate version named Silly Snake.io 13](#_Toc529556327)

[ii) **Google Snake Game:** 13](#_Toc529556328)

[ Pros/Features: A decent game with time limit of each gaming session 13](#_Toc529556329)

[ Cons/Issues: Initial access to the internet must be available 14](#_Toc529556330)

[ Popularity/Market Share: Seems to have been left in the past as a historical landmark. 14](#_Toc529556331)

[iii) **Slither.io** : 14](#_Toc529556332)

[ Cons/Issues: Initial access to the internet must be available 14](#_Toc529556333)

[ Popularity/Market Share: This game seems to be very popular as we could see many students playing this in the Learning Commons with full screen window. 14](#_Toc529556334)

[5. Requirements Analysis 15](#_Toc529556335)

[1. Scope 15](#_Toc529556336)

[a. Boundary of the Topic: 15](#_Toc529556337)

[b. Main Components/Modules: 15](#_Toc529556338)

[c. Inputs/Outputs: 15](#_Toc529556339)

[d. Functions it will accomplish: 15](#_Toc529556340)

[e. Resources and constraints: 15](#_Toc529556341)

[10. Project Success Criteria 15](#_Toc529556342)

[A. The product will work online as well as offline 15](#_Toc529556343)

[B. The product will exhibit one push notification 15](#_Toc529556344)

[C. The product will function for the basic movements, defined key operations, user exit detection for graceful shutdown. 15](#_Toc529556345)

[D. The product will make use of high resolution graphics with colors. 15](#_Toc529556346)

[6. Design 15](#_Toc529556347)

[11. System Specifications 15](#_Toc529556348)

[A. System and sub-system Functional specifications: 17](#_Toc529556349)

[i) Non-Functional Requirements 18](#_Toc529556350)

[13. Use Case Model 18](#_Toc529556351)

[14. Activity Diagram 19](#_Toc529556352)

[7. Testing 19](#_Toc529556353)

[Quality assurance 19](#_Toc529556354)

[8. Final Product (Abilities / Constraints) 20](#_Toc529556355)

[ The product will work online as well as offline 20](#_Toc529556356)

[ The product will exhibit one push notification 20](#_Toc529556357)

[ The product will function for the basic movements, defined key operations, user exit detection for graceful shutdown. 20](#_Toc529556358)

[ The product will make use of high resolution graphics with colors. 20](#_Toc529556359)

[Chapter B : Umbrella / Support Activities 21](#_Toc529556360)

[1. HR Effort 21](#_Toc529556361)

[2. Risks 22](#_Toc529556362)

[3. Software Engineering models and principles 23](#_Toc529556363)

[4. Integration 24](#_Toc529556364)

[5. Reflective Assessment 24](#_Toc529556365)

[Appendices 24](#_Toc529556366)

[2. Technical Specifications 24](#_Toc529556367)

[1. 12. Technical Requirements 24](#_Toc529556368)

# Chapter A : SDLC Activities

## Proposal

Brief Explanation of Concept

A minimalistic web-based Arcade/Retro Snake game with a clean, and friendly UI integrating high scores. This game will be mainly client side with an addition of Service workers to implement a progressive web application. A progressive web application caches its results upon the very first request of the application made by any browser, which will assist in providing access to users who do not have internet readily available.

## Background

Motivation/Reason

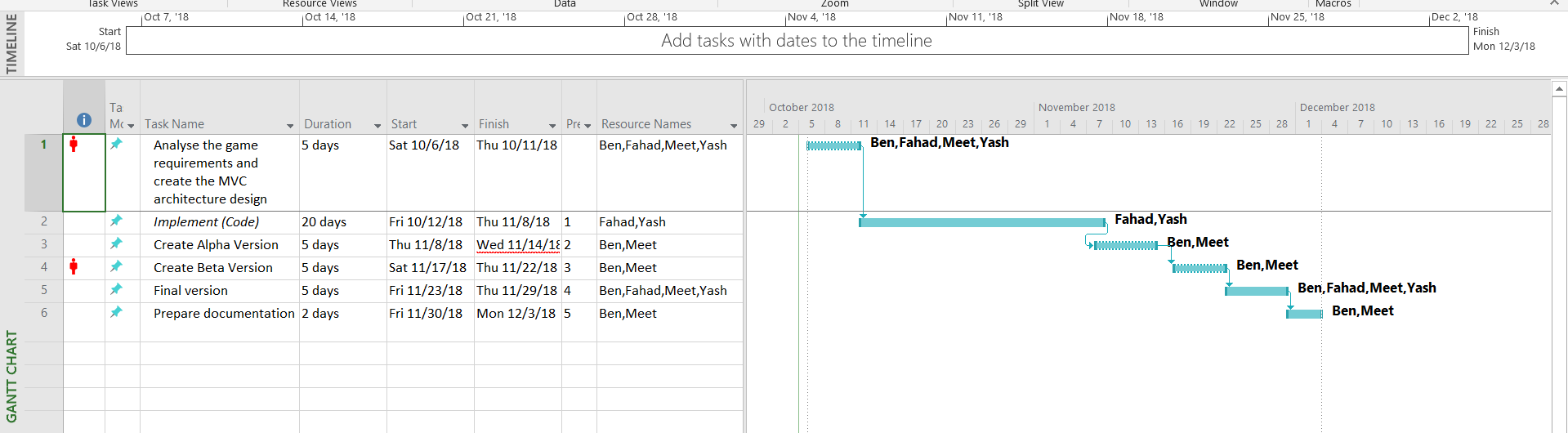
The motivation behind implementing a minimalistic web game that was a fan favorite in the 1990s; is to provide a quick and easy access to the same game but with modern visuals and approach. Internet access is a privilege many take for granted especially in rural areas and third world countries, where internet is either scarce or a privilege; only available to the rich.  Allowing users from these areas to be able to save the game upon first load up dramatically decreases the data spent on requesting for the same application via their browsers.

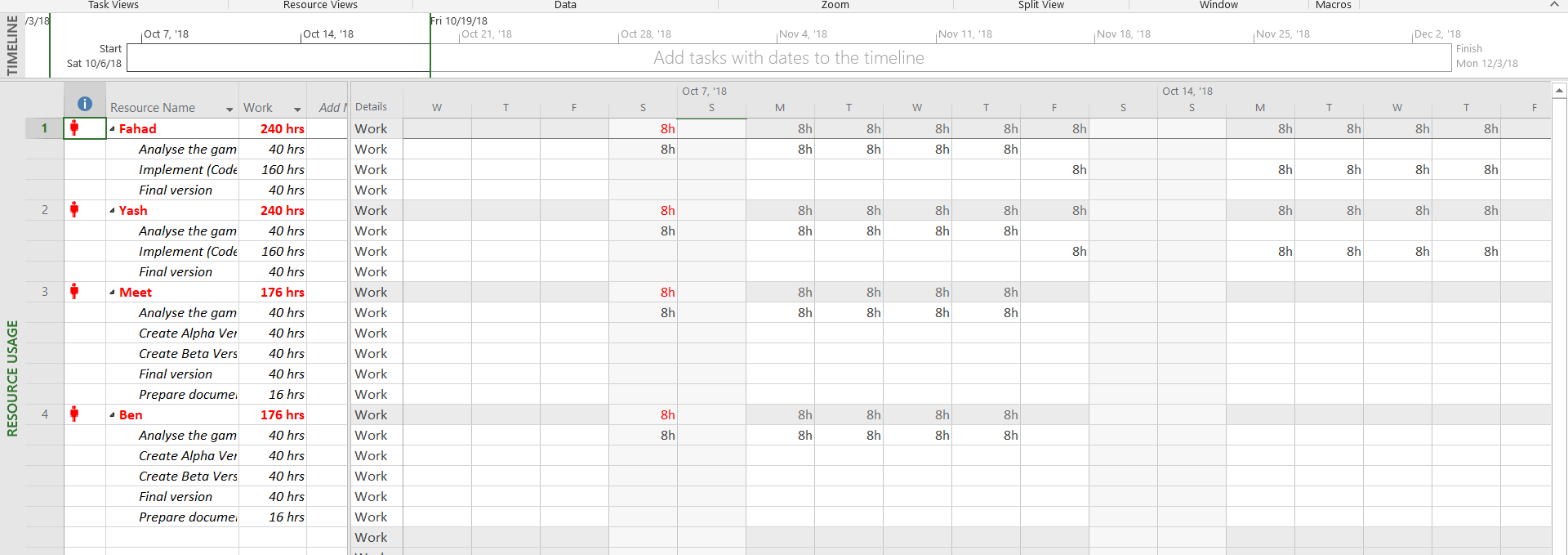
## Time Schedule

## Work Breakdown Structure

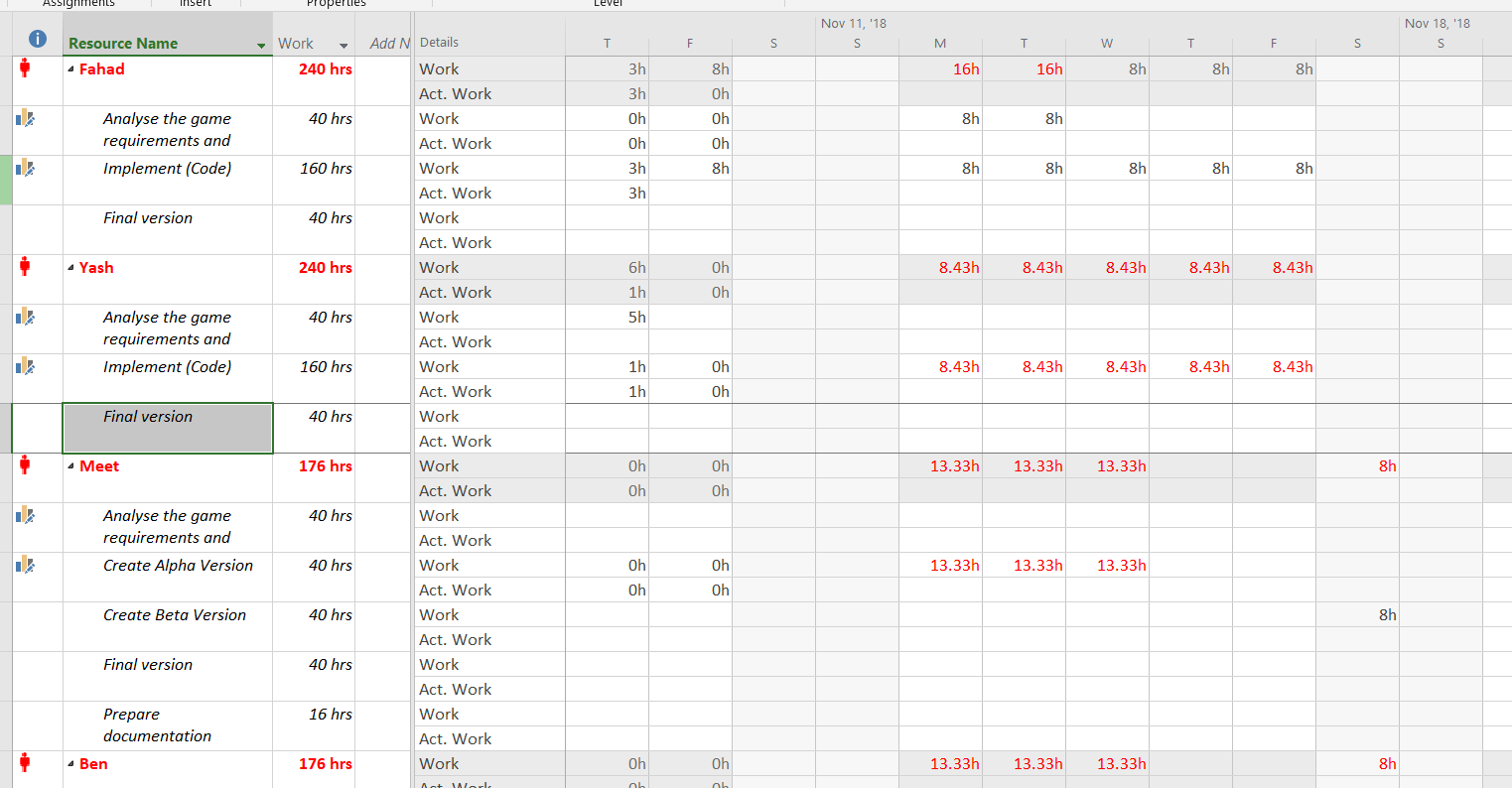
|  |  |  |
| --- | --- | --- |
| **WBS LEVEL 1**   1. Web Based Snake Game | **WBS LEVEL 2** | **WBS LEVEL 3** |
| 1. Analyse the game requirements and create the MVC architecture design   *5 days* | 1. Design Model component of MVC architecture and finalise *1 day* |
| 1. Design View component of MVC architecture and finalise *1 day* |
| 1. Design Controller components of MVC architecture and finalise *1 day* |
| 1. Review the design of the MVC components and finalise *2 days* |
| 1. Implement (Code) *15 days* | 1. Implement (Code) the Model and Unit Test *5 days* |
| 1. Implement (Code) the View and Unit Test *5 days* |
| 1. Implement (Code) the Controller and Unit Test *5 days* |
| 1. Create Alpha Version *5 days* | 1. Integrate MVC components *1 day* |
| 1. Do Integrity Test, debug *3 days* |
| 1. Release *1 day* |
| 1. Create Beta Version *5 days* | 1. Enhance and fix bugs *4 days* |
| 1. Release *1 day* |
| 1. Final version *5 days* | 1. Enhance and fix bugs *4 days* |
| 1. Release *1 day* |
|  | 1.6 Prepare documentation 2 days | 1.6.1 Prepare draft 1 day  1.6.2 Prepare final document 1 day |
|  | Total 37 days (7.5 weeks) | Total: 37 days (7.5 weeks) |

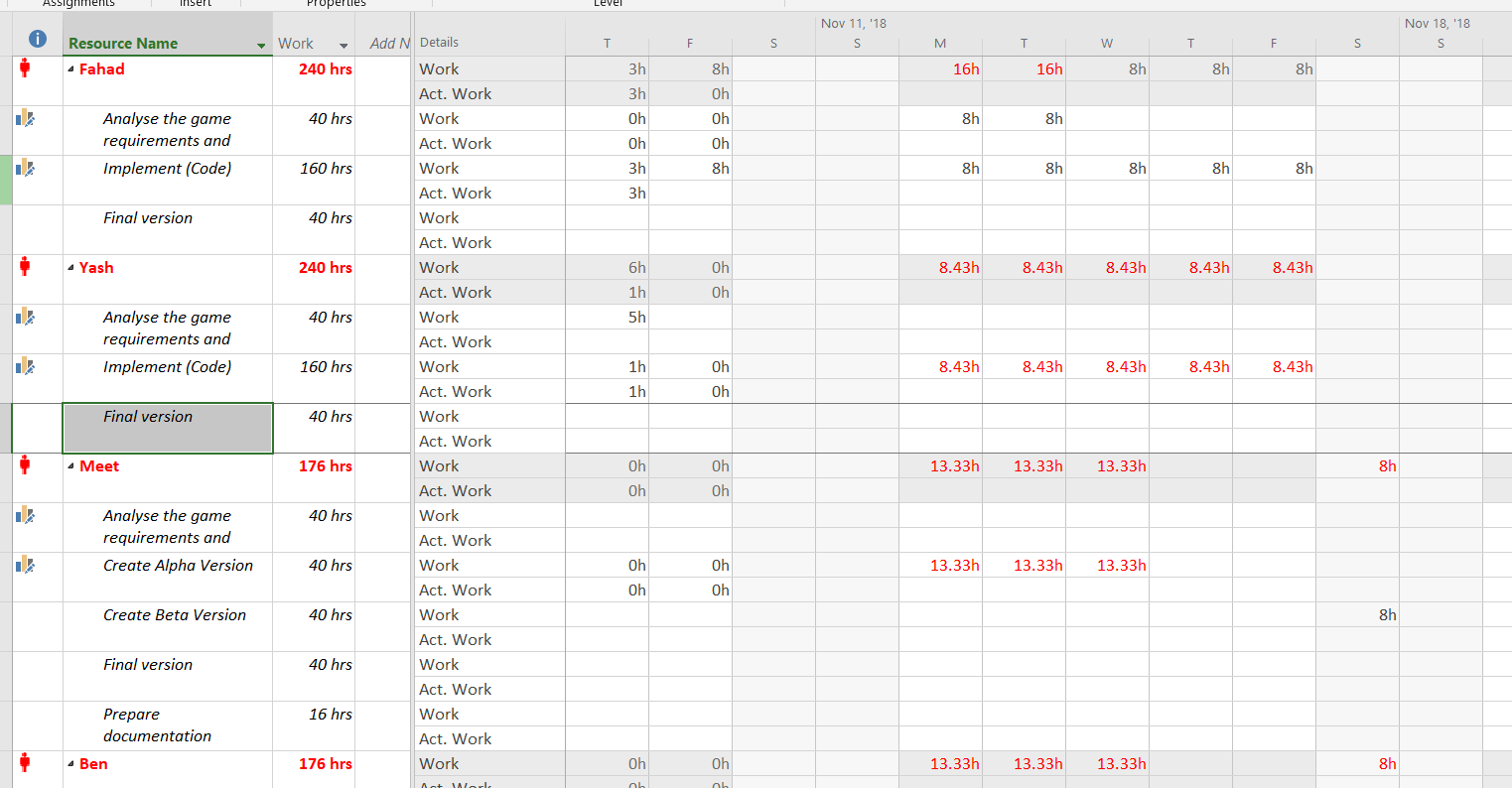
## Initial Timeline at the start on Sep 28th, 2018:





## Timeline on Nov 9, 2018





## Business Case

### Reasons and Motivation behind the proposed project:

We firmly believe that the snake movement simulation game that we plan to develop as part this project will provide following benefits to the gaming community and to the general audience:

## Value Added by this product

The ability to play a game while you are on a computer with the internet or while on a mobile phone without a data plan is the key advantage we want to offer this product. This type of product is relatively new in the market, because by choosing either one of the two alternatives makes the game development less complicated, but at the cost of limitation to the users. We would like to develop this game to present that it is possible to have a game that could do both.

The value added by the snake game project is purely entertainment based as it’s quick, easy, and accessible to anyone with an internet connection or without (Having visited the application once). Even if users don’t have a stable internet connection they can still receive entertainment from this platform-based game. Aside from it being accessible it’s user friendly and provides a classic game in a modern time, with enhancements done for the game to be able to survive and thrive in a modern environment.

## Intended Audience

By making the game available online as well as making it compact and auto deployable to the user’s workstation, we intend reach the online gaming users as well as the users at home who are restricted to the limited internet bandwidth. Even though the major internet service providers have started to provide internet speeds in the upwards of 25 Mbps – 50 Mbps, the same service providers have either started to restrict the amount of download a retail user could have. In many cases, the service providers which used to provide the large amount of download capability, have either increased the charges for the amount users could consume or have stopped allowing the large amounts of download altogether.

In the case of young gaming enthusiasts who are restricted from playing online games by means of limited or timebound internet access, the game that we are developing would provide these users one more alternative to play the game offline after the game is installed to their workstation initially. This will allow them to be entertained.

By making this game offline, the users who experience frequent internet service interruptions would also be at advantage.

## Targeted Market

Primarily the game will be hosted on a North American Web Hosting service, which will be accessible to all countries allowing access to the North American content.

The target market however could be remote areas of Asia and Africa where the internet access is still relatively expensive and prone to interruptions.

Alternative target market would also be the remote North American Cottages and Campgrounds where internet service is pay-per-use.

Lastly, for the gamers in the modern neighborhoods where the internet is available, this would still be applicable to allow them to be mobile by continuing the use of this game on their mobile device such as laptop, tablet and cell phone. They don’t need to be in the same wireless range anymore.

## Other features: Optionally, we would paint/draw the snake objects as opposed to use the ASCII block characters to form the snake shape.

### Market Survey of competition for similar products

A simple google search for “snake game” will yield results about the products that currently exist in this market space. The market study will include our analysis of each top product with include strengths and weaknesses with a brief explanation in similarities and differences.

[patorjk.com/games/snake/](http://patorjk.com/games/snake/) -> Contains: rugged UI, Minimalistic graphics, No Progressive web(PWA) application support, Small web application that loads relatively fast. 0.9 KB. The strengths are that this game is very small in size, quick to load and get playing. The weaknesses are that this application is poorly built that showcases this application was not built with modern software engineering practices, it does not incorporate a PWA approach and does not implement high scores and different levels within the game. It is at best a showcase of how a game can be transmitted to the end user.

<https://playsnake.org/> -> Contains: Decent friendly UI, Decent graphics, No Progressive web application support(PWA), Double the download size of the first analyzed product. 1.8 KB. The strengths include a friendly UI, Decent graphics and relatively small download size. The weakness is that it does not incorporate a PWA approach and does not implement high scores and different levels within the game.

<https://www.coolmathgames.com/0-snake> -> Contains: Cluttered UI, Game is just one amongst hundreds, Decent graphics, No Progressive web application support(PWA), almost 25 times the size of the second analyzed product. 25 KB. The strengths include that this game can be played amongst hundred others. The weaknesses include the fact that there is no support for users with limited internet connection as the size of the download is large. It does not incorporate a PWA approach and does not implement high scores and different levels within the game. It is infested with advertisements that limit the end user from enjoying the game for what it’s built for.

These products are top 3 results that you receive for searching the above keyword. Now, we have analyzed their source code via the browser and the results are produced on the side of the hyperlink to each product. Looking at the result of these said products we can easily deduce that these products are similar in terms of what they provide for the end user; which is entertainment. But if you dive deeper on each product they are different in every way from their strengths and weaknesses described in the analysis.

What our product will achieve is the perfect balance between the strengths and no weaknesses from the analysis we conducted on other products in the market. We intend to include a friendly, uncluttered UI that goes well with decent graphics. Add progressive web application support for our end users to be able to retain the downloaded application and use it in times of no internet access. Implementation of high scores and different levels within the game. A small download size which will not exceed the current market expectations of about 2 KB and most importantly it will be built following modern software engineering practices. Analyzing the failure of each product how these products are positioned in this market we can easily be recognized as a superset of each top product available in the market and by that we mean, a “perfect” product.

### Competitive Products Reviewed

#### Lego Pac-Man Snake Game :

<http://www.gahe.com/s/Lego-Ninjago-Rise-Of-The-Snakes-games>, <http://www.gahe.com/Silly-Snakes>

Silly Snakes is a colorful game with selection for snake skin color. Defined window with other online players for interactive play.

### Pros/Features: Colorful

Online

Control for Left and Right edge boundaries with Lower and Upper Edge screen scrolling.

High Score board

A small window showing potentially lucrative area to find more fruits to gobble to score more.

### Cons/Issues: Initial access to the internet must be available

As soon as the game detects that the internet is not available for advertisement ingestion, the game exits.

Full of advertisements and distractions

### Popularity/Market Share: There is a Silly Snake 2.0 but does not seem appealing. There is an alternate version named Silly Snake.io

#### Google Snake Game:

<https://elgoog.im/snake/>

A decent game with time limit and various objects to consume. Also available offline but the initial access to internet must be available for the game to start. Not available offline. Only after the initial internet access, the game is accessible to be played while internet connection is not interrupted.

### Pros/Features: A decent game with time limit of each gaming session

Available offline after initial internet access

Background music with on/off toggle

Inspired by Chinese year 2013, the year of the snake.

Score for current play

### Cons/Issues: Initial access to the internet must be available

Fixed time limit for each game.

Fixed snake color.

Fixed background music which could get annoying after a few minutes.

No ability to collaborate with other gamers, only a single user play.

### Popularity/Market Share: Seems to have been left in the past as a historical landmark.

#### Slither.io :

<http://slither.io/>

A great game with colorful objects to consume. Not available offline.

Pros/Features: A decent game with time limit of each gaming session

Makes great use of colors

Known in the social media

Ability to name the snake and skin selection for paid members

High Score Leader board

Ability to play with other online gamers.

### Cons/Issues: Initial access to the internet must be available

Some advertisements.

No music!!!

Could cause addition

### Popularity/Market Share: This game seems to be very popular as we could see many students playing this in the Learning Commons with full screen window.

## Requirements Analysis

### Scope

1. Boundary of the Topic: A web game application
2. Main Components/Modules:MVC/MVP style architecture/design of the application with service works for implementation of PWA, Service workers differ for each browser. Main modules on the client side are hosted inside an app shell.
3. Inputs/Outputs:Inputs will be user interaction via a mobile device or a Desktop. Output will be directly to a user’s screen.
4. Functions it will accomplish: Refer to main components/modules section (b) above
5. Resources and constraints:Can be accomplished within the allotted time. 1-2 Developers required for implementation and one of designer/analyst/scrum master/documentation writer.

### B. Project Success Criteria

1. The product will work online as well as offline
2. The product will exhibit one push notification
3. The product will function for the basic movements, defined key operations and user exit detection for graceful shutdown.
4. The product will make use of high resolution graphics with colors.

## Design

### A. System Specifications

When a user launches a webpage, the game board canvas will be painted with the welcome splash screen and brief instructions on how to run the game.

The user will have “Start Game”, “Pause Game”, “Stop Game” buttons control the game operations.

When the “Start Game” button is pushed.

The game board canvas will clear, and the colorful snake will be drawn in the centre of the screen with a randomly selected direction to start the snake movement in left, right, up or down direction slowly.

The user will then be required to control the snake’s movement maneuver using the keyboard arrow keys.

At the random time during the game play, a food object with distinct colour other than the snake colour would appear on the screen. It will stay on the screen until the user directs the snake to the food object at which point, the food object would be consumed by the snake and the food object would disappear from the screen. Optionally the snake’s length may increase.

* User should be able to control the snake using the keyboard arrow keys to move Up, Down, Left or Right
* A food object should be there on the game board canvas for the snake to eat and grow
* The snake should grow from the initial size after eating the food object
* The food object should be reachable by snake. In other words, the food object should be drawn a few pixels inside the game board canvas.
* The score should be displayed for the food consumed by the snake or the distance travelled.

### System and sub-system Functional specifications:

* Game board canvas attributes: will be maintained to set the initial size of the game board. The width, length and the colour will need to be maintained.

The size attributes may change depending upon the user action.

The game board canvas colour may be controlled internally depending upon game events.

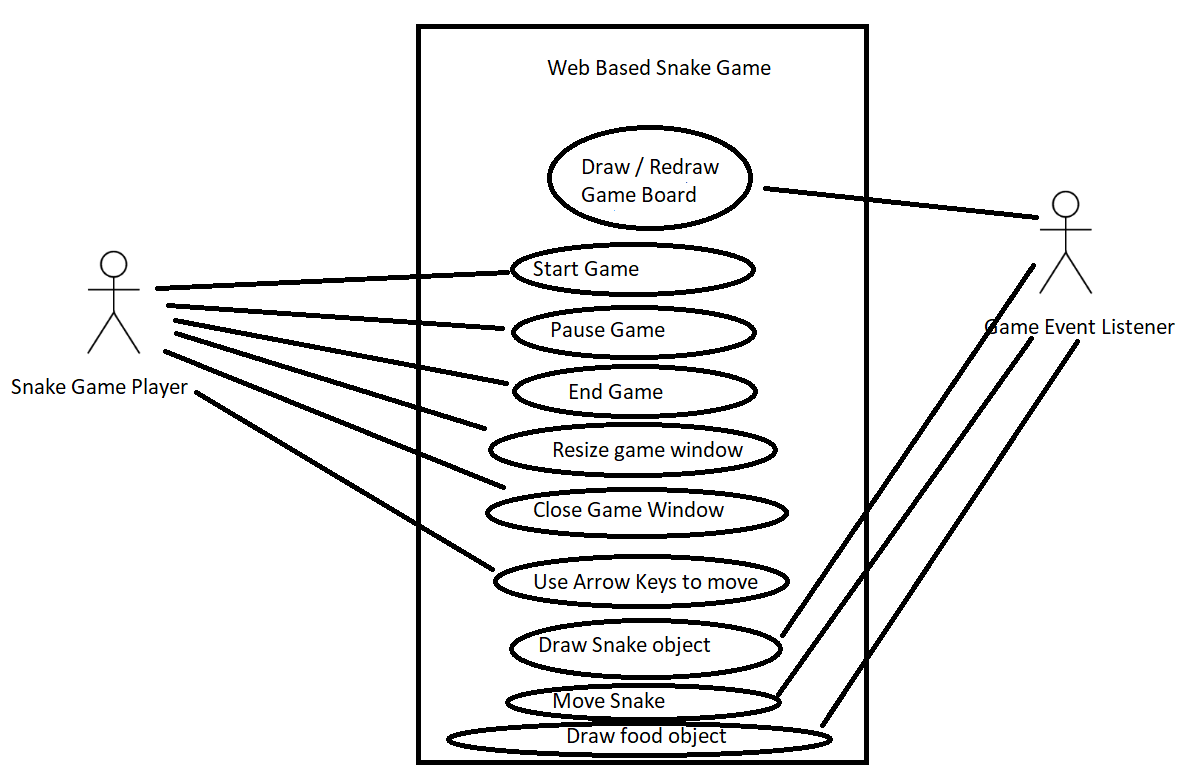
* Snake Object attributes: will be maintained using an array object of variable length changing dynamically as game progresses.
  + - * Length
      * Snake head = First element
      * Snake tail = Last element
      * Snake body pieces = elements between head and tail. Initial number would start at 2, but can increase or decrease as game progresses
* Window event listener object: will be responsible for responding to following:
  + - * game board window size change,
      * game board window closing,
      * user pressing the arrow keys,
      * user selecting to pause the game
      * user selecting to restart the game
* Food object generation: This object should be generated at a random time to draw a food object on the canvas
  + - * One food object can be present at a time on the board canvas
      * Once the food object is consumed, the snake size can be adjusted, and the food object count should be reset to indicate that the new food object can be generated.
* Score object:
  + - * Current game score and
      * the current high score object.

Above will be used to display the score of the user in terms of food objects consumed by user’s snake. It may optionally display the distance travelled in the current game or the time of play in the current game.

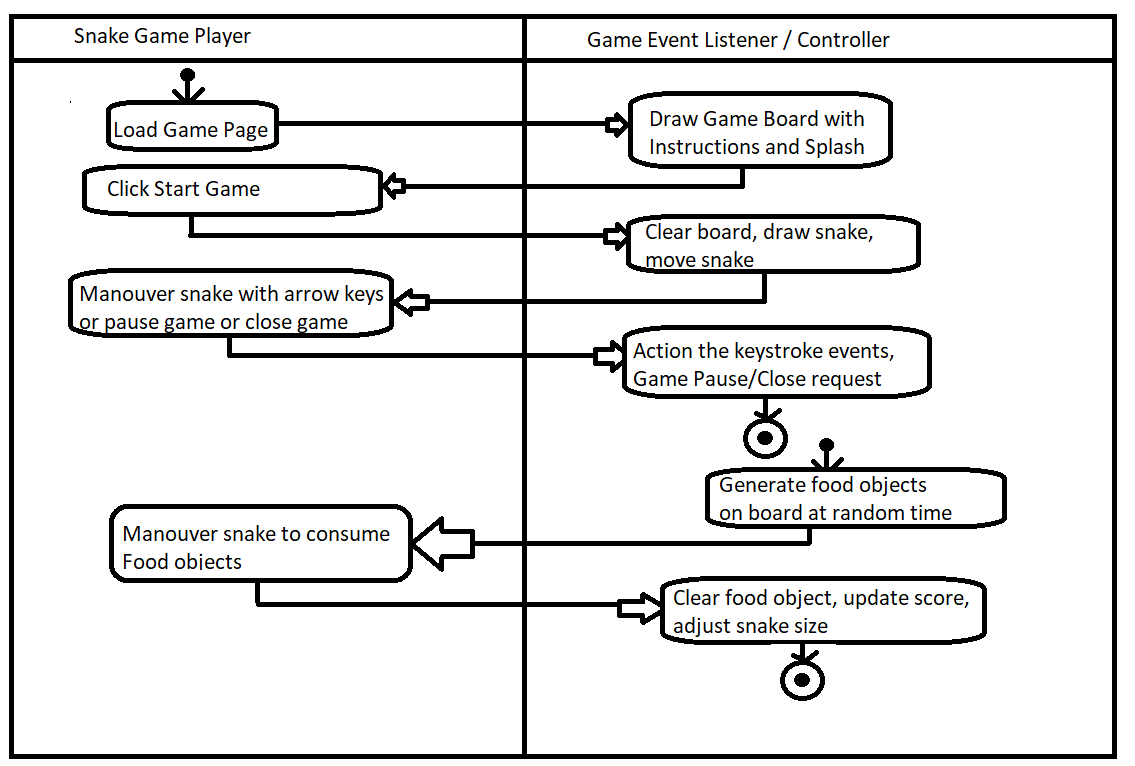
### Non-Functional Requirements

* User should be able to run the game in any web browser
* Game board canvas should not be affected if the user changes the window size
* Snake should be confined to stay within the game board canvas. Alternatively, the snake could exit from right boundary and enter from left boundary or vice versa and could exit from top boundary to enter from the bottom boundary or vice versa.

### D. Use Case Model



### E. Activity Diagram



## Testing

### Quality assurance

In order to make sure our product is being incorporated with quality work, Our team members meet once a week to discuss issues that we have and to provide solutions to problems specifically in developing the product. In order to assure the solution to those problems are of quality we have a brainstorm session of possible solutions others might come up with and the effectiveness of those solutions so we only pick the best solution.

Monitoring these solutions to problems even if they are not discussed with the team must go be subject to a walk through by another team member, who goes over the solution and then proceeds to do a code coverage. Code coverage being that all inputs invoke an action and that any action that is not invoked by either another action or an input must be deleted or if asked to keep - must provide a reason as to why that action cannot be deleted. Team members decide on whether or not an action should stay or be deleted.

This method is effective as the whole team is involved in measuring the quality of the product so that it is up to the standards held by the team members. The time spent doing this exceeds the time allotted but the outcome is always in the favor of a substantially viable product.

## Final Product (Abilities / Constraints)

As stated in the project success criteria, the aim is to develop this product using Progress Web App (PWA) design so that it provides following benefits.

## The product will work online as well as offline

## The product will exhibit one push notification

## The product will function for the basic movements, defined key operations, user exit detection for graceful shutdown.

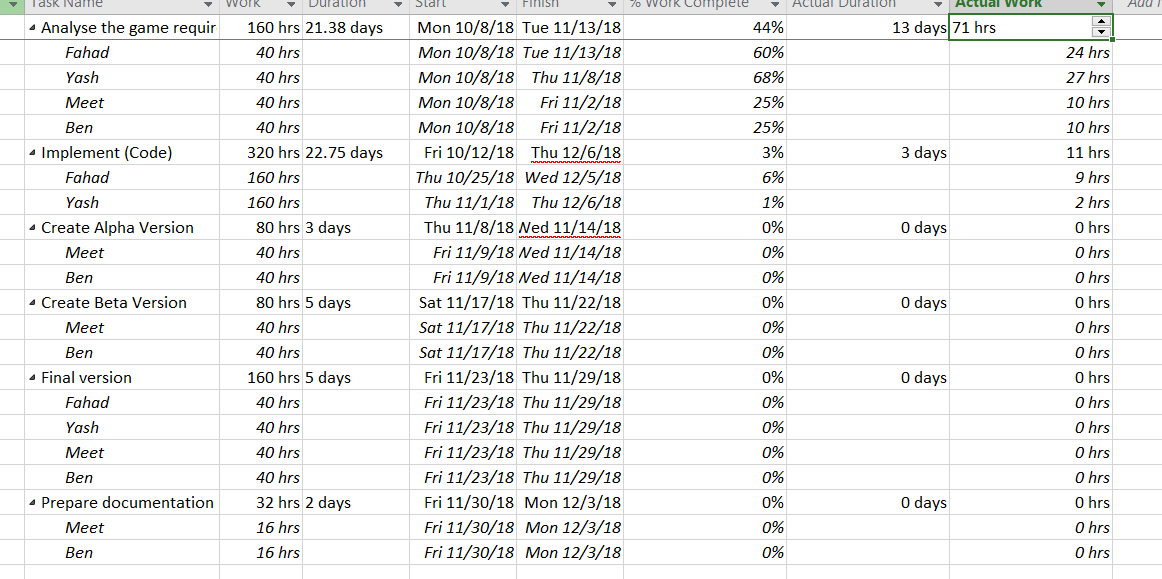
## The product will make use of high resolution graphics with colors.

* User should be able to run the game in any web browser

# Chapter B : Umbrella / Support Activities

## HR Effort

Following table shows the actual work done on the project.



Team = 4 developers

Duration = 12 weeks in part-time capacity

Each member able to put 40 hours per week

4 members x 40 hours / week by each member

= 160 hours / week spend by the team on the project

Budgeted At completion BAC = Total effort size = 12 x 160 = 1920 hours

Full time developers is expected to work 8 hours / day = 40 hours / week = 160 hours / month

Man months = 1920 / 160 = 12 man months

Up to Nov 9th, 2018

BCWS = 160 + 320 hours = 480 hours

BCWP = 71 + 11 hours = 82 hours.

% scheduled for completion BCWS / BAC = (160 + 320) / 1920 = 480 / 1920 = 25%

% complete = BCWP / BAC = (71 + 11 hours)) / (160 + 320 hours) = 82 hours / 1920 = 4.27%

SPI = BCWP / BCWS = 82 / 480 hours = 17.08%

SV = BCWP – BCWS = 82 – 480 hours = -398 hours = Project is behind schedule.

## Risks

|  |  |  |  |
| --- | --- | --- | --- |
| Risks | Impact | Probability | Total Risk |
| Delivery deadline will be tightened | 9 | 6 | 54 |
| Customer will change requirements | 5 | 8 | 40 |
| Lack of training on tools | 6 | 9 | 54 |
| Staff inexperienced | 4 | 7 | 28 |
| Productivity issues | 7 | 7 | 49 |
| Communication difficulties | 5 | 7 | 35 |
| Size of final product being too large | 7 | 5 | 35 |
| Cross-platform compatibility | 8 | 5 | 40 |

To mitigate and manage these said risks, certain steps will be taken. For all risks there will be early indicators as to whether or not the risk impact/probability is increased or decreased.

**Delivery deadline will be tightened** - Mitigating this risk will require that we meet as a team and identify where the bottleneck is and the different ways we can allocate resources to reduce the bottleneck so tasks can be completed.

**Customer will change requirements** - Mitigating this risk will require that we as a team meet with the customer to discuss what changes they would like added and in what iteration of our development process would they like to see this change implemented in.

**Lack of training on tools** - Mitigating this risk will require the team members that lack training in tools needed to perform their tasks be trained by staff that already know how to use those tools. Allocating some project hours to training team members on such tools is beneficial even though it might tighten project deadlines.

**Staff inexperienced** - Mitigating this risk will require that we either tighten the project deadline by training the inexperienced staff or we allocate resources to acquire experienced staff. The decision comes from the probability of delivery deadline of project.

**Productivity issues** - Mitigating this risk will require that team members alert their scrum master or project manager about their progress day by day so as to forecast when the team member will reach completion of the task given to them.

**Communication difficulties** - To mitigate this risk we, as a team need to have regular discussions that don't involve work so that team members get comfortable with the idea of being able to discuss issues, progress, and status of their work/project. If team members see that others are approachable then they are more likely to approach them.

**Size of final product being too large** - Mitigating this risk will require that team members in charge of developing the product be asked if certain features could be removed after consulting the customer in order to reach the size quota. If customer wants all of the features then team members identify which feature/component is the largest in terms of size and find an alternative to reach the same functionality by redoing the component by removing the excess size.

**Cross-platform compatibility** - Mitigating this risk will require that whichever browser the product isn't compatible with be assessed by a team member and features that are not compatible be highlighted. This will allow for better understanding of why these features aren't compatible and since the technology we are implementing is fairly new then we need to be able to hold those features off for a later date so the updates make them compatible or find a work around.

## Software Engineering models and principles

Currently we are following the waterfall methodology until we reach the Implementation and coding phase.

During the implementation and coding we intend to apply the Agile methodology to work in pairs for the code development and review.

One pair would do code development and review.

The second pair would perform the Quality Assurance Testing.

## Integration

Although the game needs to be deployed on a web server, the development and the actual demonstration of the game would be done using a standalone laptop running a Google Chrome or Microsoft Internet Explorer.

## Reflective Assessment

From the timeline assessment, we have found that we had overallocated the effort hours for the Analysis design phase compared to the functionality needed in this product.

The design phase also had extra hours. Even though our team spent the few hours than budgeted, the project is meeting the timelines up to the design phase.

We also anticipate that for the development and coding of the product, we may have overallocated the effort hours and may not need all the allocated time.

# Appendices

## Technical Specifications

### 12. Technical Requirements

**Languages used:**

* HTML Version 5 Graphic Objects
* JavaScript

**IDE:**

* Notepad++,
* MS-Visual Studio Editor

**SW Patterns:**

* Windows Event Listener for following:
  + listen for Keystroke events
  + Window closing event
  + Window resizing event
  + Button clicked event

**HW:** Any hardware supported by the MS-Windows and Mac Operating systems.

A hardware supported by X-Windows environment on Main-frame or super-minis may also work, but it would be tested in future releases.

-----\*\*\*-----