SE 3XA3: Software Requirements Specification
Staroids

Team 20, Staroids
Eoin Lynagh, lynaghe
Jason Nagy, nagyj2
Moziah San Vicente, sanvicem

October 3, 2018

Contents

List of Tables

List of Figures

This document describes the requirements for .... The template for the Software Requirements Specification (SRS) is a subset of the Volere template (**?**). If you make further modifications to the template, you should explicity state what modifications were made.

1 Project Drivers

1.1 The Purpose of the Project

The purpose of the project is to create a fun and dynamic interactive web game, based on the classic 80s arcade game asteriods, complete with keyboard controls and sound. This game will be designed to, hold the attention of the user and provide hours of entertainment as they work towards beating previous high scores.

1.2 The Stakeholders

For the development of Staroids, there are some key shareholders that have impact on what decisions are made and and in effect, have sway in the outcome of the project. The stakeholder's primary role is to ensure that Staroids is developed properly and that all teams involved in developemnt are satisfied with the project. The main stakeholder in Staroids are the developers as well as the client and the customer. In the case of the Staroids project, the Staroids team are the developers. The clients of the project are both the original HTML5 Asertoids developer and the Staroid team, and lastly the customer of the project is once again the developers and any online web game players.

1.2.1 The Client

Staroids is developed for the original creator of HTML5 Asteroids with the purpose of using the proper implementation and documentation techniques. The primary concern of the original creator is Staroids' faithfulness to the original as well as the adaptations of any edits that the creator wanted to make but did not get the chance to do. The Staroids team also takes the part of the client because the team very much wants to complete this project

for themselves. It offers a chance to program in a language that is new and tackle a problem that the team has not attempted yet. The developer's main concern is that the project has a straightforward implementation method.

1.2.2 The Customers

The customers of the project are the developers again and web game users. The developers are the customers as this project is also being created for their sake as a challenge in JavaScript. As such, the developers concern as a customer is that the project provides an insightful and valuable learning opportunity. Web game players are also clients as they will also consume the project once it has been created. Their concerns are where and how the game will be played and how easy it is to get running on someone's machine.

1.2.3 Other Stakeholders

Some other stakeholders that could impact the project are advertisers and web game companies. Advertisers may look to spreak the availability of the Staroid project to new users, so they require the project to be unique, special or different in some way so that they can advertise to others and bring the Staroids project into the attention of new users. Web game companies may also be stakeholders because they may look to advertise or host Staroids on their web sites. In terms of advertising, they would be similar to the advertisers, but for hosting, the companies will need the project to meet a set of techinal requirements. The web game companies may need the project to be in a certain format, only use generic libraries or be written in a certain manner.

1.3 Mandated Constraints

1.4 Naming Conventions and Terminology

As described in the Development Plan, our main naming convention is to make use of camel case. This has a distintive look, as well as being easy to type.To name files, we will be descriptive in the name of the file, making sure that it is obvious to all developers what file contains what information,

to name variables, we will also use camel case, and variable names will be a combination of attribute and object, so for example, if we need to store the velocity of the ship, the variable will be named velocityShip.

1.5 Relevant Facts and Assumptions

User characteristics should go under assumptions.

2 Functional Requirements

2.1 The Scope of the Work and the Product

2.1.1 The Context of the Work

-similar to problem statement

2.1.2 Work Partitioning

2.1.3 Individual Product Use Cases

2.2 Functional Requirements

- Run on Google Chrome, Mozilla Firefox, Microsoft Edge and Apple Safari browsers.
- The game shall contain pre-game, playing, post-game, and paused states.
- When initially ran, the pre-game screen shall show first.
- On the pre-game screen, if the play button is pressed, the playing screen shall show.
- On the press of the pause button during playing, the pause state shall show.
- On the press of the pause button while paused, the playing state shall appear again.

- The playing screen shall always display the player character, score and lives.
- Every time the player character is hit by an enemy, the lives count shall decrease by one.
- When the fire button is pressed, the player character will fire a projectile.
- If a projectile hits an enemy, the enemy will be removed, that enemy's death action will occur and the score will be incremented.
- If zero lives remain and the player character is hit, the game shall enter the post game screen.
- After a set amount of time, an enemy character shall appear. He shall shoot at the player character.
- If a large asteroid is hit, it will separate into 3 medium asteroids and the score shall be incremented.
- If a medium asteroid is hit, it will separate into 3 small asteroids and the score shall be incremented.
- If a small asteroid is hit, it will be removed and the score shall be incremented.
- No player character can spawn if there is an asteroid near by.
- No more than six projectiles can be on the screen at a time.

3 Non-functional Requirements

3.1 Look and Feel Requirements

- Staroids should have visually appealing graphics.
- Staroids should have intuitive controls.
- The player character should move fluently.

3.2 Usability and Humanity Requirements

- On the pre-game screen, game controls should be shown.
- In pre-game, post-game and pause states, relevant prompts should be shown.
- All text should be written in English in a readably sized font.

3.3 Performance Requirements

- The playing state should not stutter or freeze.
- The project should always run above 60 frames per second.
- Staroids should load to the pre-game state in a few seconds or less.
- Game states should change immediately after the relevant button is pressed.

3.4 Operational and Environmental Requirements

- Users should not need to install any external software to run Staroids as long as there is a web browser present.
- Staroids should not use so many computer resources as to impact other running programs.

3.5 Maintainability and Support Requirements

- Staroids should be sufficiently modularized so that edits to a specific aspect of the game are quick.
- Documentation should be kept up to date.
- Each internal function, object and complicated component should be documented and commented to allow an outsider to understand that component.

3.6 Security Requirements

- Staroids should not have an accessor methods to any of the internal variables or game state objects and variables.
- Staroids should not transmit any malicious code or programs to a user's machine
- Staroids should not take any personal information from the user.

3.7 Cultural Requirements

- Staroids should not use any symbols that may be considered offensive or rude in its target demographics.

3.8 Legal Requirements

- Staroids should not use logos, charcters or symbols that are owned by or licenced to people other than the creators of Staroids.

3.9 Health and Safety Requirements

- The controls of Staroids should not put strain on the hand.
- No screens containing flashing colours should be shown.

This section is not in the original Volere template, but health and safety are issues that should be considered for every engineering project.

4 Project Issues

4.1 Open Issues

4.2 Off-the-Shelf Solutions

There are many implementations of asteroids in a variety of languages that could offer inspiration to Staroids. The core game that underlines Staroids and the others is the same, but implementation methods and add-ons are what separates them. Some games offer a more realistic physics simulation

while others focus on the visual experience. The other implementations also are written in different languages for different platforms, so if a user cannot run Staroids with its JavaScript and HTML implementation, those users have alternatives. Imput methods also vary amongst the implementations.

4.3 New Problems

4.4 Tasks

The steps to deliver the project are all covered in the dynamic Gantt chart Gantt Chart that is being used and updated throughout the project. It covers all tasks and deliverables throughout the project with a breakdown of what goes into each as well as who is working on it, plus the amount of their efforts that are going into it. The Development phases of the project are also shown on their and are color coded to easily distinguish the different ones.

4.5 Migration to the New Product

4.6 Risks

As this is an online game, there is very little physical risk or ability to cause injury. The software itself is not running any processses that could be vital to the safety of a person. However, they may be other risks, not involving the safety of people. For example, if our code is poorly optimized, it could result in slowdown or overheating of the computer, which can cause probems.

4.7 Costs

The three main types of costs associated with the project are monetary, storage and CPU. The goal is for the project have a monetary cost of $0 CAD. In terms of storage take up a minimal amount of space ideally either less than the original implementation, or if greater in the maximum range of double the originals size. Finally the CPU usage is not very concerning for this project due to how small and simple the project is as well as it being run through a web browser. Any modern computer with a web browser installed will be able to easily run Staroids.

4.8 User Documentation and Training

Staroids is to be documented using the JSDoc 3 documentation solution. JSDoc 3 allows for inline commenting in JavaScript documents to be converted into a documentation website. Since JSDoc documentation is written in the JavaScript source code, the Staroids team can modify it simultaneously with any edits to any function. This allows for completely up to date documentation while making it easier on the developers. All functions, classes and states are to be documented using JSDoc as soon as they are created, and any edits made to the them must be reflected in the JSDoc comments. A special documentation file will be created for training people on how to use Staroids. The document will have a table of what each keyboard key will do and how they can be used. There will also be a short explanation of the rules and goals of the game.

4.9 Waiting Room

4.10 Ideas for Solutions

References

5 Appendix

This section has been added to the Volere template. This is where you can place additional information.

5.1 Symbolic Parameters

The definition of the requirements will likely call for SYMBOLIC_CONSTANTS. Their values are defined in this section for easy maintenance.

Table 1: **Revision History**

Date	**Version**	**Notes**
Sept 26	0.1	Added team and project info
Sept 28	0.11	Divided document work
Sept 28	0.12	Added basis of functional requirements
Oct 2	0.13	Added stakeholders
Oct 2	0.135	Added Tasks with link to Gantt Chart
Oct 3	0.14	Added some non functional requirements, off the shelf solutions and documentation
Oct 3	0.145	Added Purpose of Project, Naming Conventions and Terminology and Risks
Oct 3	0.15	Added non functional requirements and basis of context of the work