Software Requirements Specification

for

Clicker Clicker Game

Version 1.0 approved

Prepared by Hannah Milton

04/04/2019

Table of Contents

Revision History

Name	Date	Reason for Changes	Version
Hannah Milton	03/02/2018	Initial Creation	1.0
Hannah Milton	05/04/2019	Updated system features	1.1

1. Introduction

1.1 Purpose

This SRS details the function and nonfunctional requirements of Clicker Clicker 1.0. Clicker Clicker is a simple game in which users will aim to increase their score via clicking on various buttons.

1.2 Document Conventions

No document conventions are being used at this time.

1.3 Intended Audience and Reading Suggestions

This document is intended to be read by the developers and testers of the project.
A suggested sequence of reading is to follow the document in chronological order. Developers may want to focus on section 4 which details the system requirements.

1.4 Product Scope

Clicker Clicker is a mobile and desktop game.
The user will aim to increase their score and earn available achievements by clicking on various buttons.
Clicker Clicker will be available on iOS, Android, Windows, Linux and MacOS.

1.5 References

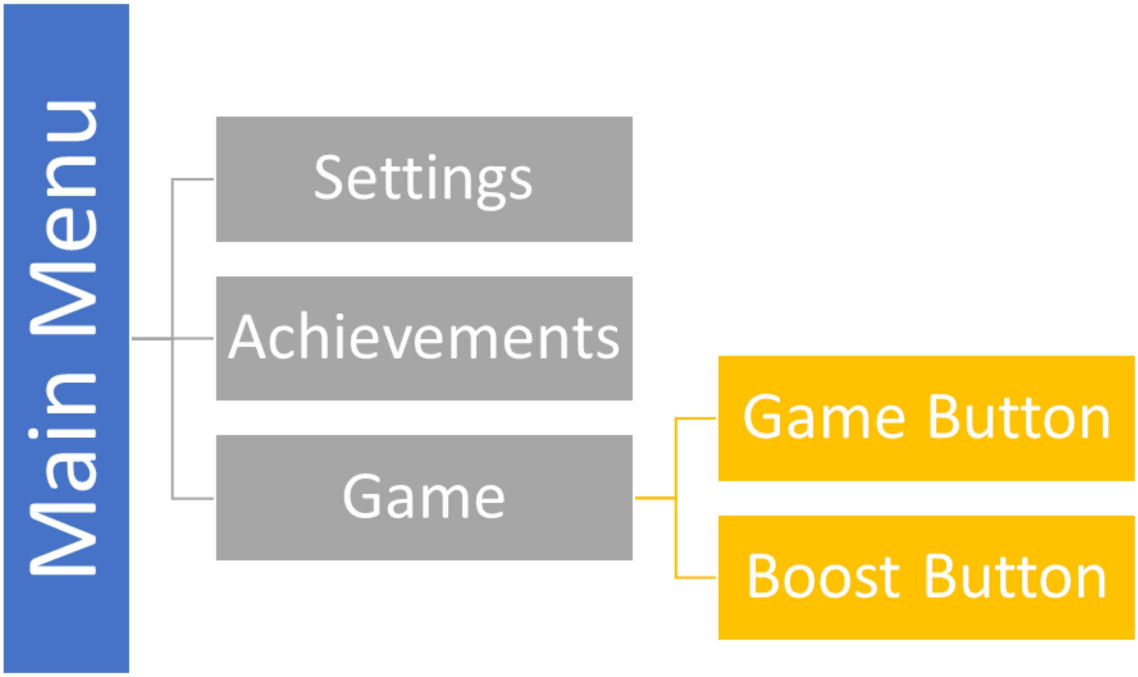
[1] IEEE Software Engineering Standards Committee, “IEEE Std 830 - 1998, IEEE Recommended Practice for Software Requirements Specifications”, October 20, 1998.

[2] *Orteil.dashnet.org. (2018).* Cookie Clicker. *[online] Available at: http://orteil.dashnet.org/cookieclicker/ [Accessed 4 Feb. 2018].*

[3] *Hyper Hippo Game Studio - Kelowna B.C. (2018).* Hyper Hippo Game Studio - Kelowna B.C.. *[online] Available at: https://www.hyperhippo.ca/adventure-capitalist [Accessed 4 Feb. 2018].*

1.6 Product Perspective

This project takes inspiration from similar games in the idle genre such as Cookie Clicker [2] and Adventure Capitalist [3]. The main aim of the game is to accumulate score via clicking buttons. Score can also be accumulated over time with the option to automate the clicking of buttons.



1.7 Product Functions

FE-1: Main menu to choose between various scenes
FE-2: Adjust sound, resolution and full screen mode
FE-3: Start a new game or continue with an existing one
FE-4: Clickable game buttons that increase the score
FE-5: Automated option to increase score during idle time
FE-6: Unlockable achievements

1.8 Operating Environment

This application will run on iOS, MacOS, Android, Windows 64bit, Windows 32bit and Linux machines. It will run on the latest versions of these operating systems.

1.9 Design and Implementation Constraints

This application will be built in Qt using QML and C++.

1.10 User Documentation

No user documentation will be provided with this project

1.11 Assumptions and Dependencies

No assumptions or dependencies at this time

2. External Interface Requirements

2.1 Hardware Interfaces

The application will be interacted via keyboard and mouse or via touch.

2.2 Software Interfaces

No software interfaces identified at this time

2.3 Communications Interfaces

No communication interfaces will be required.

3. System Features

3.1 Main Menu Screen

Upon loading the game, the user should be presented with the Main Menu. This screen will include all the actions that the user can perform next.

Priority: High

3.1.1 Stimulus/Response Sequences

Stimulus: User clicks Start Game button
Response: User is taken to the game screen

Stimulus: User clicks Restart Game button
Response: User is prompted to confirm

Stimulus: User confirms restarting the game
Response: Prompt screen closes, user is taken to the game screen

Stimulus: User rejects restarting the game
Response: Prompt screen closes, user remains on the main screen

Stimulus: User clicks Achievements button
Response: User is taken to the achievements screen

Stimulus: User clicks Settings button
Response: User is taken to the settings screen

Stimulus: User clicks Quit button
Response: The application shuts down

3.1.2 Functional Requirements

REQ-1: There should be a button with the text “Start Game” which when clicked, will take the user to the Game Screen
REQ-2: There should be a button with the text “Restart Game” which when clicked, will prompt the user to confirm the action. If there is no saved game available, this button will be disabled.
REQ-3: When “Restart Game” is clicked the user will be prompted to make sure they confirm this action. On confirmation, the user’s last saved game will be overwritten with a fresh state (see Feature 3.3 for more information) and they proceed to the game screen. On rejection, the user remains on the Main Menu screen.
REQ-4: There should be a button with the text “Achievements” which when clicked, will take the user to the Achievements Screen
REQ-5: There should be a button with the text “Settings” which when clicked, will take the user to the Settings Screen
REQ-6: There should be a button with the text “Quit” which when clicked, will close the application
REQ-7: There should be a title with “Clicker Clicker”
REQ-8: There should be text that shows how long your last game session was if this exists
REQ-9: There should be a logo for the game

3.2 Load Game

Upon opening the application, a saved game will try to be located. When successful, the last session play time is formatted into the text described in 3.1 REQ-7. When no game can be located, the text is not shown.

Priority: High

3.2.1 Stimulus/Response Sequences

Stimulus: User opens the application
Response: A saved game is searched for and the text for last session play time is updated on success or hidden on failure.

3.2.2 Functional Requirements

REQ-1: The application should search for an existing saved game
REQ-2: On success, the data for the last session play time should be collected and the text displaying this should be updated.
REQ-3: On failure, the text displaying the last session play time is hidden from the user.

3.3 Restart Game

Upon clicking the restart button and confirming the action, the current saved game is overwritten with a clean slate.

Priority: High

3.3.1 Stimulus/Response Sequences

Stimulus: User clicks to restart the game and confirms the action
Response: The current saved game is overwritten with a clean slate

3.3.2 Functional Requirements

REQ-1: The application should locate the current saved game and overwrite the data with a clean slate

3.4 Saved Game

When playing the game, if the back button is clicked or the application is closed, the game's state will be saved. The game will save:

- Each game button's current score
- Each game button's current amount
- Each game button's current cost
- Each game button's current cooldown
- The length of playtime for the current session
- The total amount of playtime
- The main button's cooldown
- The total score of the game

Priority: High

3.4.1 Stimulus/Response Sequences

Stimulus: User clicks on the back button in the game screen
Response: The state of the game is stored. This consists of each game button's score, amount, cost and cooldown. The main game button's cooldown. The total amount of playtime, the playtime of the current session and the total score of the game.

Stimulus: User closes the application
Response: The state of the game is stored. This consists of each game button's score, amount, cost and cooldown. The main game button's cooldown. The total amount of playtime, the playtime of the current session and the total score of the game.

3.4.2 Functional Requirements

REQ-1: When the back button on the game screen is pressed the state of the game is saved
REQ-2: When the application is closed, the state of the game is saved

3.5 Settings Screen

The user should be able to change various settings in the game. These settings will be:

- Sound
- Resolution
- Full screen

The resolution and full screen option should only show on non-mobile device applications.

Priority: Medium

3.5.1 Stimulus/Response Sequences

Stimulus: User drags the progress bar handler on the sound
Response: The game sound is increased/decreased depending on which way the slider is dragged

Stimulus: User clicks on resolution drop down box
Response: The drop-down box is populated with various resolution sizes

Stimulus: User clicks on one of the resolution sizes
Response: The application changes to the resolution size chosen

Stimulus: User clicks on the full screen checkbox
Response: If the checkbox was ticked, the checkbox becomes unticked and the application will exit full screen mode. If the check was not ticked, the checkbox becomes ticked and the application will enter full screen mode.

Stimulus: User clicks on the back button
Response: The user will be returned to the main menu screen

3.5.2 Functional Requirements

REQ-1: There should be a title showing the user they are on the settings screen
REQ-2: There should be a label for the sound option which is accompanied by a slider which is draggable
REQ-3: The slider position should directly affect the sound of the application
REQ-4: There should be a label for the resolutions which is accompanied by the drop box
REQ-5: The user should be able to select a resolution from the drop-down box and the application should adjust to this resolution
REQ-6: The option to change resolution should only exist on desktop applications and not on mobile devices
REQ-7: There should be a label for full screen which is accompanied by a checkbox
REQ-8: The option to choose full screen should only exist on desktop applications and not mobile devices
REQ-9: There should be a button with the text "Back" which when clicked, will take the user to the Main Menu screen
REQ-10: The settings of the application should be remembered each time the application is loaded

3.6 Achievements Screen

The user should be able to access a grid of all achievements accomplished in the game and achievements they have yet to achieve.

Priority: Low

4.1.2 Stimulus/Response Sequences

Stimulus: User clicks on the back button
Response: User is taken back to the Main Menu screen

Stimulus: User clicks on one of the achievements
Response: A popup shows the achievement details (how to acquire the achievement)

4.1.3 Functional Requirements

REQ-1: There should be a title showing the user they are on the achievements screen
REQ-2: There should be a grid of buttons with images of each achievement available
REQ-3: Achievements marked as completed should be full in clarity
REQ-4: Achievements not currently achieved should be slightly greyed out
REQ-5: Clicking on an achievement should create a popup. This popup should provide further details on the achievement
REQ-6: There should be a button with the text "Back"

3.6.1 100 Amount Achievement

There should be an achievement for having 100 of a game button

Priority: Low

3.6.1.1 Stimulus/Response Sequences

Stimulus: User purchases a game button and the total amount goes to 100 or above
Response: A popup on the screen with an image of the achievement and the name of the achievement

3.6.1.2 Functional Requirements

REQ-1: Be able to detect when a game button has an amount of 100
REQ-2: Pop up achievement image and achievement name when game button amount is 100
REQ-3: When achievement pops up, the achievement is marked as complete in the achievement screen

3.6.2 4 Hours Game Time Achievement

There should be an achievement for having played 4 hours of game time

Priority: Low

3.6.2.1 Stimulus/Response Sequences

Stimulus: User reaches 4 hours of game time
Response: A popup on the screen with an image of the achievement and the name of the achievement

3.6.2.2 Functional Requirements

REQ-1: Be able to detect when game time has exceeded 4 hours
REQ-2: Pop up achievement image and achievement name when game time exceeds 4 hours
REQ-3: When achievement pops up, the achievement is marked as complete in the achievement screen

3.6.3 Use the Boost Achievement

There should be an achievement for the first time having used the Boost button

Priority: Low

3.6.3.1 Stimulus/Response Sequences

Stimulus: User clicks on the boost button for the first time
Response: A popup on the screen with an image of the achievement and the name of the achievement

3.6.3.2 Functional Requirements

REQ-1: Be able to detect the first time the boost button is clicked
REQ-2: Pop up achievement image and achievement name when boost button is clicked for the first time
REQ-3: When achievement pops up, the achievement is marked as complete in the achievement screen

3.6.4 100 of Each Button Achievement

There should be an achievement for having 100 of each game button

Priority: Low

3.6.4.1 Stimulus/Response Sequences

Stimulus: User has 100 of all buttons par 1, they then purchase more of that button to bring its total to 100 or over
Response: A popup on the screen with an image of the achievement and the name of the achievement

3.6.4.2 Functional Requirements

REQ-1: Be able to detect when each game button has 100 or over in quantity
REQ-2: Pop up achievement image and achievement name when last remaining game button reaches 100 or more in quantity
REQ-3: When achievement pops up, the achievement is marked as complete in the achievement screen

3.7 Game Play

On starting/resuming a game, the user will then be able to play the game. The game consists of 7 buttons in total. 6 of the buttons will provide the user with a variety of scores depending on which button is clicked. The 7th button will be a boost button and can only be clicked at certain times (after 4 hours of gameplay – if the boost button has already been clicked, the four hours will be from the moment the button cooldown has ended). This boost button will provide a boost of x10 to the game buttons. Once the timer has counted down on the boost button, the boost will end.

The user should be able to interact with all 7 buttons (when off cooldown) and see the total score.

On leaving the game, the current state of the game should be saved. The user should be able to continue from this state when resuming the game.

Priority: High overall but individual priorities can be seen in the below sections

3.7.1 Game Screen

The game screen will consist of game buttons, a boost button, the total score and a back button.

Priority: High

3.7.1.1 Stimulus/Response Sequences

Stimulus: User clicks on one of the game buttons
Response: See below section on game buttons

Stimulus: User clicks on boost button
Response: See below section on game buttons

Stimulus: User clicks on back button
Response: The state of the game is saved and the user is taken back to the Main Menu screen

3.7.1.2 Functional Requirements

REQ-1: The screen will display 6 game buttons
REQ-2: The screen will have one boost button
REQ-3: The screen will have a button with the text “Back” which when clicked will go back to the Main Menu screen
REQ-4: Text will be shown on the screen which shows the current score

3.7.2 Game Buttons

There should be 6 buttons in the game. Each button will have:

- Name
- Score
- Cost
- Amount
- Cooldown
- Cost to automate the button

When the user clicks a button, its score will be added to the total score. A cooldown will then begin and during this cooldown, the button cannot be clicked.

Priority: High

3.7.2.1 Stimulus/Response Sequences

Stimulus: User clicks on the game button
Response: The score of the button is added to the total score of the game. A timer will then start and the length of this timer is dictated by the button’s cooldown. During the cooldown the button cannot be clicked. After the cooldown has finished, the button can then be clicked

3.7.2.2 Functional Requirements

REQ-1: When a button is clicked the score of the button is added to the total score

REQ-2: When a button is clicked, the cooldown of the button is started and the button cannot be clicked during this period

REQ-3: When the button is in its cooldown period, the button cannot be clicked and the visual of the button will change to show this

REQ-4: The button's text will show the name of the button

REQ-5: The button's text will also show the cost of the button

REQ-6: The button's text will also show the amount of the button

REQ-7: The button's text will also show the score of the button

3.7.3 Automate Game Button

Each game button has the chance to be automated. A button will be attached to the game button that allows the user to buy this automation. The user will deduct a certain amount from their score to buy this automation option. This will then allow the button to be automatically clicked every time it is off of cooldown.

Priority: High

3.7.3.1 Stimulus/Response Sequences

Stimulus: User clicks to buy automate button option
Response: The cost of this automation is deducted from the total score. The button will click itself every time its cooldown is at 0.

3.7.3.2 Functional Requirements

REQ-1: A button with the text "Automate" or some graphical icon to represent this

REQ-2: When clicked, the cost of automating the button is reduced from the total score

REQ-3: The button with the automate option will press itself every time it is off cooldown

3.7.4 Buy Game Button

The user can increase the amount of the button by clicking on a purchase option which will be attached to the game button it correlates to. A button will show how many of the game button can be bought with the current score. When purchasing more, the score, the cost and the amount will increase.

The score will increase by a factor of e for each additional button bought. The cost will increase by a factor of 2 for each button bought. The amount of the button will increase by however many were bought.

The total score will deduct the cost of the buttons bought.

Priority: High

3.7.4.1 Stimulus/Response Sequences

Stimulus: User clicks on buy option

Response: The game button's amount will be updated depending on the amount bought. The score of the button will increase by a factor of e times the amount of the button. The cost of the button will increase by a factor of 2 for each button bought. The text on the Buy Game Button will update to reflect how many of the button can now be bought. The total score of the game will deduct the cost of the buttons times how many were bought.

3.7.4.2 Functional Requirements

REQ-1: The Buy Button text will show how many buttons can be bought with the total score

REQ-2: When the buy button is clicked, the amount that can be bought is added to the amount of the button and text showing the amount that can be bought is updated

REQ-3: When the buy button is clicked, the cost of the button times the amount bought is deducted from the total score

REQ-4: The cost of the button will be the amount of buttons there are times by 2

REQ-5: The score of the button will be the button amount times a factor of e

3.7.5 Boost Button

A boost button will be active once every 4 hours of gameplay. If the button has previously been clicked, this 4-hour countdown will start from when the button's action has finished. When clicked, a timer will begin that counts down from 10. While active, the score of each game button will increase by 10. After the timer reaches 0, the boost button will become inactive.

Priority: Medium

3.7.5.1 Stimulus/Response Sequences

Stimulus: User clicks on the boost button
Response: Every other game button will have their score increased by 10. To show that the button is active, a glow will be shown around the button. A timer begins on the boost button which counts down from 10. Once this timer reaches 0, the glow on the button disappears, the other game button's scores will go back to normal and this button will not be able to be clicked for another 4 hours of game time.

3.7.5.2 Functional Requirements

REQ-1: When this button is clicked, every other game button has their score increased by 10

REQ-2: When this button is clicked, every button in the game will have a glow around them

REQ-3: When this button is clicked, a timer counting down from 10 will start

REQ-4: After the timer of the button reaches 0, the button will not be able to be clicked for 4 hours of game time

REQ-5: This game button will check for the current play time and will determine if the button can be clicked or not – this will depend on if the game has 4 hours of gameplay

3.8 Game Time

A timer continually keeps track of how long the user has been playing their current game for.

Priority: Medium

3.8.1.1 Functional Requirements

REQ-1: When the user plays a game, the time they spend in the game is stored

REQ-2: On closing the application or going back to the main menu, the timer stops counting and resumes when going back to the game screen

4. Other Nonfunctional Requirements

4.1 Performance Requirements

4.1.1 Windows

Operating System: Windows 10
Memory: 512 MB RAM
DirectX: Version 9.0
Storage: 60 MB available space
32 bit and 64 bit will be supported
Game should run at a steady 60 frames per second (fps) during gameplay.

4.1.2 Mac OS X

Operating System: Mac OS X 10.6
Memory: 512 MB RAM
Storage: 60 MB available space
Game should run at a steady 60 frames per second (fps) during gameplay.

4.1.3 Linux

Operating System: Ubuntu 10.10+, Mint 17.1+
Memory: 512 MB RAM
Storage: 60 MB available space
32-bit and 64-bit will be supported
Game should run at a steady 60 frames per second (fps) during gameplay.

4.1.4 iOS

Operating System: Requires iOS 8.0 or later. Compatible with iPhone and iPad.
Storage space: 100mb available space

4.1.5 Android

Operating System: Requires 8.1.0 “Oreo” or later.
Storage: 400mb available space
32-bit and 64-bit ARM

4.2 Safety Requirements

No safety requirements are needed for this project as little impact will occur on crashes.

4.3 Security Requirements

No security requirements are needed for this project as no personal data is stored.

4.4 Software Quality Attributes

Adaptability & Maintainability: It is likely that the game will continually include new features. Therefore, the design of this project should take this into account. The design should allow for flexibility in adding new features to the project.