

# Quesadilla Ratón

Software Requirements Document

Team Name: Runtime Terrors That Keep You Up At Night

Date: 5/1/2020

Team Members: Anthony Cardenas Barrera, Gregory Dwyer, Erika Sudderth, Hasler Zuniga

Honor Code: I HAVE ABIDED BY THE UNCG Academic Integrity Policy ON THIS ASSIGNMENT.

Student Signatures: Anthony Cardenas Barrera, Gregory Dwyer, Erika Sudderth, Hasler Zuniga

# 1. Introduction

## 1.1. Table of Contents

Introduction	<b>1</b>
Table of Contents	1
Purpose	3
Document Conventions	3
Intended Audience	3
Definitions/Jargon	3
Project Scope	3
Technical Challenges	3
References	3
Overall Description	<b>4</b>
Product Features	4
User Characteristics	4
Operating Environment	4
Design and Implementation Constraints	5
Assumptions and Dependencies	5
Functional Requirements	<b>5</b>
Primary	5
Secondary	6
Technical Requirements	<b>6</b>
Operating Systems/Compatibility	6
Interface Requirements	6
User Interface	6
Hardware Interface	6
Software Interface	6
Communications Interface	6
Nonfunctional Requirements	<b>7</b>
Performance Requirements	7
Safety/Recovery Requirements	7
Security Requirements	7
Policy Requirements	7
Software Quality Attributes	7
Availability	7
Correctness	7

	2
Maintainability	7
Reusability	7
Portability	8
Process Requirements	8
Development Process Used	8
Time Constraints	8
Cost and Delivery Date	8

## 1.2. Purpose

This System Requirements Document (SRD) covers a description of the game and all functional, technical, and nonfunctional requirements for the game Quesadilla Ratón.

## 1.3. Document Conventions

For readability, this document is divided into five major sections. Section 1 contains an introduction and information about this document. Section 2 contains an overall description of the game, while Sections 3, 4, and 5 contain the functional, technical, and nonfunctional requirements of the program respectively. All major sections are further divided into smaller subsections.

## 1.4. Intended Audience

The intended audience of this document is the UNCG CSC 340: Software Engineering Section One students and teacher, Ike Quigley.

## 1.5. Definitions/Jargon

Throughout this document there are references to in-game objects. All in-game objects that can interact with each other are referred to as sprites. The Mouse is the playable character, while the Cat is the enemy, a non-playable character who moves across the game window. Hairballs are damage causing stationary objects. Cheese are stationary objects that the Mouse can collect and deposit into the Door, the destination guarded by the Cat. Player and User refer to anyone controlling the Mouse. This program is referred to as an application, a game, and a project.

## 1.6. Project Scope

This application was created as a submission for the UNCG CSC 340: Software Engineering Section One final project.

## 1.7. Technical Challenges

Technical challenges include learning JavaFX, applying MVC structure, and using the Twitter API.

## 1.8. References

“Social media API, posting to Twitter” Twitter4j.org. 2007.  
This is a guide to how to use Twitter4j twitter bot.

“JavaFX Game Tutorial.” YouTube. JavaCraving. 2018.

This is a general tutorial for game mechanics using JavaFX.

“Creating a Sprite Animation with JavaFX.” Netopyr. Michael Heinrichs. 2012.

This guide was the basis for our sprite animation class.

“JavaFX Basic Key Event Handling.” Youtube. David Burnham. 2018.

This is a YouTube tutorial on Key Pressed and Key Released events in relation to image movement in JavaFX.

JavaFX 8. Oracle Docs. 2015.

This is online documentation for JavaFX 8. It gives a basic overview of how to use everything JavaFX 8 is capable of.

“Learn JavaFX absolute beginners.” Tutorialspoint.

This is an online tutorial of creating an application display with JavaFX.

“C64 TrueType Font” Style64. 2020.

This is the font used in the application.

“Java: Read Text File Easily” Youtube. Alex Lee. 2018

This is a tutorial for learning how to open and read a simple text file.

## 2. Overall Description

### 2.1. Product Features

Quesadilla Ratón is a game in which the user controls a Mouse whose only goal is to deliver pieces of Cheese to its Door so it can make as many quesadillas as it wants. The Mouse must avoid Hairballs and a Cat throwing dangerous Claws in order to place Cheese pieces in the Door and score points. If the user’s health reaches 0, the game is lost. The user can choose to display their score on the project’s twitter account, [twitter.com/runtime\\_terrors](https://twitter.com/runtime_terrors), and see if they made it onto the leaderboard.

### 2.2. User Characteristics

This program is intended for English speaking users of any age.

### 2.3. Operating Environment

The program runs on Windows OS, Mac OS, and Linux OS running Java Version 8.2 with the JavaFX 8 library.

## 2.4. Design and Implementation Constraints

The guidelines for this project require it to be a desktop application implemented using Java. The project has to be developed using no external frameworks. It requires a connection to an external API and saving data to a persistent data store.

## 2.5. Assumptions and Dependencies

Use of Quesadilla Ratón is dependent on the JavaFX 8 library and the Twitter4j API.

# 3. Functional Requirements

## 3.1. Primary

On the game screen, this program takes key presses in as input. In the upper left corner are the user's initial health and the user's initial score of 0. These values are updated throughout the game. An interaction between sprites involves one's bounds overlapping the other's bounds. A Mouse is placed on the lower portion of the game window. Players can use the four arrow keys to move the Mouse around the 2D game window within bounds. There are Hairballs placed throughout the game window; in the event that the Mouse interacts with a Hairball, the user's health decreases. There are pieces of Cheese placed throughout the game window; in the event that the Mouse interacts with a piece of Cheese, the Mouse will carry this Cheese in front of itself. The Mouse may hold only one piece of Cheese, interaction with any other Cheese piece is ignored. A Cat is placed on the upper part of the game window and moves along a horizontal line. This Cat releases a Claw every specified number of seconds. The Claw travels from the Cat's current location in a vertical line to just beyond the borders of the game window. In the event that the Mouse interacts with a Claw, the user's health decreases. In the event that the Mouse interacts with the Cat, the user's health decreases. There is a Door placed in the upper portion of the game window. In the event that the Mouse, holding a piece of Cheese, interacts with the Door, the user's score increases. In the event that the user's health reaches 0, the game is over and an end subscreen is displayed.

The end subscreen gives the user the option of entering their name. If the user gives their name and chooses not to post to Twitter, the user's name and score are recorded. If the user's name is given and the user chooses to post to Twitter, the user's name and score are recorded and the program posts a tweet on the team's twitter: [twitter.com/runtime\\_terrors](https://twitter.com/runtime_terrors). Otherwise, opting out results in the game returning to the main menu and the score being lost.

### 3.2. Secondary

On the initial main menu screen, this program takes mouse clicks in as input. The selection of the Start Game option allows users to begin the game. Selection of the How to Play option provides users with a subscreen containing a description of how the gameplay works. The selection of the Highscores option provides users with a subscreen showing the top 10 user scores on that computer. Selection of the Settings option provides a subscreen where users have the option of toggling the game music on and off, turning the game music volume up or down, and toggling the SFX. There are no future features to be added.

## 4. Technical Requirements

### 4.1. Operating Systems/Compatibility

This program is compatible with Windows OS, Mac OS, and Linux OS running Java Version 8.2 with the JavaFX 8 library.

### 4.2. Interface Requirements

#### 4.2.1. User Interface

The user can interact with this program through both the keyboard and the mouse of a device. In order for the application to be used, the user must be able to press buttons with a mouse, use the four arrow keys of a keyboard, and if the user would like to record their score or post it to twitter, they need to have access to a full keyboard. See included wireframe.

#### 4.2.2. Hardware Interface

Users will need a mouse and keyboard in order for the application to work.

#### 4.2.3. Software Interface

The application will read from and write to a text file containing a list of user names and scores.

#### 4.2.4. Communications Interface

This application is connected to the team member's twitter account where users can choose to post their name and score on our social media account. The API being used is Twitter4j.

## 5. Nonfunctional Requirements

### 5.1. Performance Requirements

This game is capable of being deployed on a single computer and runs in real time.

### 5.2. Safety/Recovery Requirements

In the event that the program crashes, the current user score and health will be lost and no data will be recorded.

### 5.3. Security Requirements

There are no security requirements for this application.

### 5.4. Policy Requirements

In order to avoid injury, stop playing if you experience any of the following symptoms: hands, wrists, arms or eyes becoming sore, numbness, burning, or stiffness. If any of these symptoms persist, contact your doctor.

### 5.5. Software Quality Attributes

#### 5.5.1. Availability

This application is available at all times if it is downloaded onto the user's computer.

#### 5.5.2. Correctness

This application has a high level of correctness.

#### 5.5.3. Maintainability

This application has to be downloaded again in order to receive the latest updates. In the future, the data for scores will get too big and will have to be moved to a database.

#### 5.5.4. Reusability

This application will work as long as the device it runs on has a downloaded version of Java 8.2 containing the JavaFX 8 library. Our application will be usable as long as Java keeps compatibility with 8.2 or 8.2 is available for download. All code, aside from the Twitter4j library, is property of the Runtime Terrors team.



#### 5.5.5. Portability

This application is portable. Anyone can download and play this application on any laptop/desktop running Java 8.2 with the JavaFX 8 library. The application will have different leaderboards specific to each separate device.

### 5.6. Process Requirements

#### 5.6.1. Development Process Used

This program was developed using the Agile Development Process. Meetings were held every week and involved reviewing all progress made and planning the design and progress goals for the next week. Outside of this team members worked individually throughout the week, with collaboration as needed.

#### 5.6.2. Time Constraints

Development time is restricted to the 2020 UNCG spring semester (January 13th, 2020 to May 1st, 2020).

#### 5.6.3. Cost and Delivery Date

Quesadilla Ratón has a \$0.00 budget and will be a free desktop application. The game will be released on May 1st, 2020.