Breakout!

**A picture containing can, drawing, food

Description automatically generated**

By: FreshprinceofSoftwareware

**Table of Contents**

1. **Introduction**

**1.1.        Title**

**1.1.1.        Project Name**

**1.1.2.        Team Name**

**1.1.3.        Date**

**1.1.4.        Team Members**

**1.1.5.        Stakeholders/Company - Honor Code**

**1.2.        Table of Contents**

**1.3.        Purpose**

**1.4.        Document Conventions**

**1.5.        Intended Audience**

**1.6.        Definitions/Jargon**

**1.7.        Project Scope**

**1.8.        Technical Challenges**

**1.9.        References**

**2.            Overall Description**

**2.1.        Product Features**

**2.2.        User Characteristics**

**2.3.        Operating Environment**

**2.4.        Design and Implementation Constraints**

**2.5.        Assumptions and Dependencies**

**3.            Functional Requirements**

**3.1.        Primary (main screen)**

**3.2.        Secondary bell and whistle (volume)**

**4.            Technical Requirements**

**4.1.        Operating Systems/Compatibility**

**4.2.        Interface Requirements**

**4.2.1.        User Interface**

**4.2.2.        Hardware Interface**

**4.2.3.        Software Interface**

**4.2.4.        Communications Interface**

**5.            Nonfunctional Requirements**

**5.1.        Performance Requirements**

**5.2.        Safety/Recovery Requirements**

**5.3.        Security Requirements**

**5.4.        Policy Requirements**

**5.5.        Software Quality Attributes**

**5.5.1.        Availability**

**5.5.2.        Correctness**

**5.5.3.        Maintainability**

**5.5.4.        Reusability**

**5.5.5.        Portability**

**5.6.        Process Requirements**

**5.6.1.        Development Process Used**

**5.6.2.        Time Constraints**

**5.6.3.        Cost and Delivery Date**

**1.     Introduction**

**1.1        Project Name: Break Out!**

This is a System Requirements Document (SRD) for the class CSC340-01

The requirements covered in this documentation explains how you the user can experience our game with full content

**1.1.2      Team Name**

The Team Name that was constructed is the “Fresh Prince of Software**”**

**1.1.3       Date**

**The Date that this Project was Established is February 2, 2020**

**1.1.4       Team Members**

The Team members in the “Fresh Prince of Software” are Marquis Killett, Tyler Roux, Dymond Deans, and Jerid Mei

**1.1.5       Honor Code / Stakeholders**

The *Academic Integrity Policy* was created by faculty based on the core values and ideals of practice as recommended by the Center for Academic Integrity. The UNCG Faculty Senate reviewed and approved the Academic Integrity Policy. The policy, in its entirety, is available online at Student Policy Handbook: Academic Integrity. The Office of Student Rights and Responsibilities, which handles all student conduct and academic integrity functions, has made it easier to report an incident of academic misconduct to our office. Click on “[Report an Academic Integrity Violation](https://cm.maxient.com/reportingform.php?UNCGreensboro&layout_id=2)“, which will direct you to more detailed information about reporting a violation. You will be able to upload all materials including the Academic Integrity Violation Report Form in one convenient location which will automatically be sent to the Office of Student Rights and Responsibilities for review and action.

**1.3        Purpose**

This is a System Requirements document (SRD) for the Software engineering project for CSC 340-01.

The requirements over the work corresponding to the specific java classes along with technical attributes that will help the user

**1.4        Documentation Conventions**

In this Document if the word is “*italicized*” It means that

**1.5        Intended Audience**

To have access to this SRD Please contact anyone from the “Fresh prince of Software” group. You should use this product to pass the time and have a good time with your friends without the hassle of having complex controls and technical difficulties.

**1.6        Definitions**

|  |  |
| --- | --- |
| **Tick()** | Runs each part of the game simultaneously. |
| **.sleep()** | Manipulates the speed of the game/frame rate. |
| ***BufferStrategy()*** | Positions the canvas/frame of the game. |
| .addkeyListener | Registers the controls for the game, like the arrow buttons for movement. |
| run | Runs the game. |
| .setfont | Registers the text font. |
| .setColor | Registers the color of the paddle and bricks. |
| paddlePlacementWidth() | Sets the paddle width size. |
| PaddlePlacementHeight: int | Sets the paddle height size. |
| gameSizeWidth | Sets the width size of the game screen. |
| gameSizeHeigh | Sets the height size of the game screen. |

**1.7        Project Scope**

 This document is produced as part of the Technical Specification that shall be reviewed by Prof. IKE Quigley. Therefore, making it applicable for the submission of the project

**1.8        Technical Challenges**

Getting the game to move to the next level, importing sprites into the game, connecting our api, and running the game with our menu.

**1.9        References**

For this Application what inspired us is the retrogamer called Pong

**2.  Overall Description**

**2.1  Product Features**

Within this product you will have a Starting game menu and within the game play itself you will have a high score along with some power ups that will help you get the highest high score

**2.2  Operating Environment**

Netbeans/Intelij

**2.3  User Characteristics**

Individuals who want to occupy their minds, and want to pass the time with a little challenge

**2.4 Design Implementation and Constraints**

Some of the constraints that we had during creation of this game is both the “Twitter Api” along with the game menu. With the Twitter Api it was hard  for us to be able to get what we have, which is the score that you get from the game by letting the ball pass your paddle at the bottom of the screen, or by breaking all the bricks on the screen.

**2.5 Assumptions and Dependencies**

Using the Facebook api would have destabilized the program, so we had to use the Twitter Api instead.

**3.  System Features and Requirements**

**3.1 Primary**

The main menu of the game appears as the game runs, then proceeds the game itself. After the game is completed the “Congratulations you won ” text will appear followed by the score the user has gotten. If the user loses the game the text “You lose”, followed by the score will appear. The score the user achieves will be recorded and uploaded to Twitter through the Twitter api.

**3.2** **Secondary**

The user is able to interact with the GUI, being able to start the game from the menu, and being able to close the game after either winning the game or losing it.

**4.Technical Requirements**

**4.1 Operating Systems/Compatibility**

This game is compatible with the modern-day operating systems. These include Microsoft Windows 10, MacOS X, etc. There is a possibility however that the game will have different running speeds on some operating systems compared to others.

**4.2 Interface Requirements**

**4.2.1User Interface**

The mouse and Keyboard is the main form of interaction for the game from the user. The user interacts with the game through GUI frames that run during the game’s runtime.

**4.2.2 Hardware Interface**

The game requires the typical computer utilities such as a working screen, mouse and keyboard, and Internet connection.

**4.2.3 Software Interface**

Installation of Java 8 is a requirement for this product. OpenJDK 8, or Amazon Cornetto’s JDK 8 are a few recommendations.

**4.2.4 Communications Interface**

Dialog box notification about game over and high score indication.

**5. Nonfunctional Requirements**

**5.1 Performance Requirements**

OS: Windows 7, 8/8.1, 10; MacOS X; Linux Mint Processor: Dual Core 3.0 Ghz or Greater Memory: 4GB or Greater Hard Disk Space: 10 GB Video Card: 256mb Video Memory, capable of Shader Model 2.0+ JRE: 8.0 Release; OpenJDK 8, or Amazon Corretto’s JDK 8.

**5.2 Safety/Recovery Requirements**

If the game does not run properly on the system of the user, a recommended solution would be to restart the computer.

**5.3 Security Requirements**

Security consist of the founders/owners of this application on GitHub account with ownership of the repository.

**Application requires no personal data/documents**

**5.4 Policy Requirements**

Private project owned by UNCG© ( University of Greensboro North Carolina)

**5.5 Software quality Attributes**

**5.5.1  Availability**

People of UNCG can access the information and rights of this Program Due to the Owners of this application with a GitHub account

**5.5.2  Correctness**

In this SRD all requirements Are meet in the fullest in what is contributed with in this documentation.

**5.5.3  Maintainability**

Test will be designed and implemented to the program to test for each and every single component that has problems and/or has interactions that are present.

**5.5.4  Reusability**

Easy to use you need to run the application to boot it up.

**5.5.5  Portability**

The portability System requirements comprise of the user having access to a GitHub account.

**5.6**