Software Requirements Specification

for

Burger Breakout

**Version 1.0 approved**

**Prepared by Cooper Dahlberg, David Sincyr, Ethan Esber, Kevin Finley, Michael Rumohr, and Michael Taylor**

**Six Guys**

**University of Maine**

**February 25, 2020**

**Table of Contents**

[1. Introduction 1](#_Toc33340885)

[1.1 Purpose 1](#_Toc33340886)

[1.2 Document Conventions 1](#_Toc33340887)

[1.3 Intended Audience and Reading Suggestions 1](#_Toc33340888)

[1.4 Product Scope 1](#_Toc33340889)

[1.5 References 2](#_Toc33340890)

[2. Overall Description 2](#_Toc33340891)

[2.1 Product Perspective 2](#_Toc33340892)

[2.2 Product Functions 3](#_Toc33340893)

[2.3 User Classes and Characteristics 3](#_Toc33340894)

[2.4 Operating Environment 3](#_Toc33340895)

[2.5 Design and Implementation Constraints 3](#_Toc33340896)

[2.6 User Documentation 4](#_Toc33340897)

[2.7 Assumptions and Dependencies 4](#_Toc33340898)

[3. External Interface Requirements 4](#_Toc33340899)

[3.1 User Interfaces 4](#_Toc33340900)

[3.2 Hardware Interfaces 5](#_Toc33340901)

[3.3 Software Interfaces 5](#_Toc33340902)

[3.4 Communications Interfaces 5](#_Toc33340903)

[4. System Features 5](#_Toc33340904)

[4.1 System Feature 1 5](#_Toc33340905)

[4.2 System Feature 1 6](#_Toc33340906)

[4.3 Other Nonfunctional Requirements 7](#_Toc33340907)

[4.4 Performance Requirements 7](#_Toc33340908)

[4.5 Safety Requirements 7](#_Toc33340909)

[4.6 Security Requirements 7](#_Toc33340910)

[4.7 Software Quality Attributes 7](#_Toc33340911)

[4.8 Business Rules 8](#_Toc33340912)

[5. Other Requirements 8](#_Toc33340913)

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| David Cincyr/  Michael Rumohr | 02/25/20 | Initial Creation | 1.0 |
|  |  |  |  |

# Introduction

## Purpose

The purpose of this Software Requirements Specification (SRS) document is to specify the software requirements of Burger Breakout version 1.0, a gaming application playable on a computer. Burger Breakout plans to allow users to play a fun burger-and-fries-themed gaming application on their personal computer and this is a standalone system described in this Software Requirements Specifications document.

*<Identify the product whose software requirements are specified in this document, including the revision or release number. Describe the scope of the product that is covered by this SRS, particularly if this SRS describes only part of the system or a single subsystem.>*

## Document Conventions

Large size font and bold text are representative of significant importance. Requirements are assumed to have their own priority unless specified otherwise.

*<Describe any standards or typographical conventions that were followed when writing this SRS, such as fonts or highlighting that have special significance. For example, state whether priorities for higher-level requirements are assumed to be inherited by detailed requirements, or whether every requirement statement is to have its own priority.>*

## Intended Audience and Reading Suggestions

This intended Audience is intended for marketing staff, investors, and professor Ghanavati of COS 420. The SRS is organized in compliance with IEEE Recommended Practice for Software Requirements Specifications with this introduction, descriptions and features, followed by interfaces and requirements, and appendices can be found at the end of this document. It is suggested to read through the first three sections which are most pertinent to marketing staff and investors.

*<Describe the different types of reader that the document is intended for, such as developers, project managers, marketing staff, users, testers, and documentation writers. Describe what the rest of this SRS contains and how it is organized. Suggest a sequence for reading the document, beginning with the overview sections and proceeding through the sections that are most pertinent to each reader type.>*

## Product Scope

Burger Breakout is a standalone gaming application on a personal computer designed for all. The aim of the project is to develop a gaming application for children but can be enjoyed by all ages that is centered around a Five Guys® theme. The 2-D levels would be designed around different environments of the restaurant, such as the tables where patrons eat and the kitchen where the food is prepared. The user will guide the playable character through various levels by running, jumping, fighting, and climbing. There will also be other non-playable characters that will act as allies or enemies towards the playable character. Some levels may culminate in a boss fight or even encounter mini-bosses’ in the middle of them. These non-player characters will have increased stats as well as move sets that will add a challenge for the user.

<Provide a short description of the software being specified and its purpose, including relevant benefits, objectives, and goals. Relate the software to corporate goals or business strategies. If a separate vision and scope document is available, refer to it rather than duplicating its contents here.>

## References

* Unity Scripting API: <https://docs.unity3d.com/ScriptReference/>
* IEEE Recommended Practice for Software Requirements Specifications: <https://standards.ieee.org/standard/830-1998.html>
* https://bb.courses.maine.edu/bbcswebdav/pid-6227278-dt-content-rid-17503191\_2/courses/2020.UMS05-C.0017.1/IEEE%20Recommended%20Practice%20for%20Software%20Requirements%20Specifications.pdf

*<List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.>*

# Overall Description

## Product Perspective

Burger Breakout is a new, self-contained, standalone gaming application and is designed to run on personal computers. The only requirements for the user are to have a personal and functional keyboard.

*<Describe the context and origin of the product being specified in this SRS. For example, state whether this product is a follow-on member of a product family, a replacement for certain existing systems, or a new, self-contained product. If the SRS defines a component of a larger system, relate the requirements of the larger system to the functionality of this software and identify interfaces between the two. A simple diagram that shows the major components of the overall system, subsystem interconnections, and external interfaces can be helpful.>*

## Product Functions

* Burger Breakout shall be designed as a two-dimensional side scroller personal computer gaming application.
* Burger Breakout shall have the user navigate linearly through different levels of difficulty.
* Burger Breakout shall allow the user to guide the game character using the arrow keys or WASD keys.

*<Summarize the major functions the product must perform or must let the user perform. Details will be provided in Section 3, so only a high level summary (such as a bullet list) is needed here. Organize the functions to make them understandable to any reader of the SRS. A picture of the major groups of related requirements and how they relate, such as a top level data flow diagram or object class diagram, is often effective.>*

## User Classes and Characteristics

The class of users for Burger Breakout is children. This class of children has been divided into two subclasses of novice and frequent players. The novice players do not have little to no experience and will require more guidance than the frequent players with hints and suggestions. The frequent players will have more insight and experience into gameplay. It is important to satisfy the novice user class so that they are more likely to continue playing Burger Breakout if they get frustrated during a certain part of the game.

*<Identify the various user classes that you anticipate will use this product. User classes may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the most important user classes for this product from those who are less important to satisfy.>*

## Operating Environment

The environment which the software will operate in is Windows OS. Specifically, the software will target Windows 10, but will also function on Windows 7. Currently, the software will not operate in a Linux or Mac OS environment.

*<Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.>*

## Design and Implementation Constraints

There are currently no design or implementation restraints.

*<Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer’s organization will be responsible for maintaining the delivered software).>*

## User Documentation

User documentation will include a help section for the user providing complete instructions of Burger Breakout gameplay and suggestions and hints for beating certain bosses at the end of levels.

*<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.>*

## Assumptions and Dependencies

Burger Breakout assumes and depends on the existence and reliability of a functioning Windows OS and functioning keyboard attached to that OS.

*<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan).>*

# External Interface Requirements

## User Interfaces

Burger Breakout will utilize Unity for the creation of user interface between the software product and user. See “Unity Scripting API” in the References section of the SRS for more details.

*<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>*

## Hardware Interfaces

Burger Breakout is a computer-based gaming application and will run on any platform running Windows OS that has an attached functional keyboard (see section 2.4).

*<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>*

## Software Interfaces

Burger Breakout will be written with Unity 2018.4.16f1 software. Unity is a closed-source, cross-platform game development application. Users of Burger Breakout will not need to interface with Unity once it has been created.

*<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>*

## Communications Interfaces

Burger Breakout does not require any communication interface functions since it is a standalone product that does not communicate with any other systems.

*<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>*

# System Features

*<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>*

## System Feature 1

*<Don’t really say “System Feature 1.” State the feature name in just a few words.>*

4.1.1 Description and Priority

*<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>*

4.1.2 Stimulus/Response Sequences

*<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>*

4.1.3 Functional Requirements

*<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use “TBD” as a placeholder to indicate when necessary information is not yet available.>*

*<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>*

REQ-1:

REQ-2:

## System Feature 1

*<Don’t really say “System Feature 1.” State the feature name in just a few words.>*

4.2.1 Description and Priority

*<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>*

4.2.2 Stimulus/Response Sequences

*<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>*

4.2.3 Functional Requirements

*<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use “TBD” as a placeholder to indicate when necessary information is not yet available.>*

*<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>*

REQ-1:

REQ-2:

# Other Nonfunctional Requirements

REQ-X:

## Performance Requirements

Burger Breakout

*<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>*

## Safety Requirements

Burger Breakout does not currently have requirements concerned will the possible loss, damage, or harm that could results from the its use.

*<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product’s design or use. Define any safety certifications that must be satisfied.>*

## Security Requirements

Burger Breakout does not use data from the user so there are not any requirements regarding security or privacy issues surrounding its use.

*<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>*

## Software Quality Attributes

*<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>*

## Business Rules

*<List any operating principles about the product, such as which individuals or roles can perform which functions under specific circumstances. These are not functional requirements in themselves, but they may imply certain functional requirements to enforce the rules.>*

# Other Requirements

Burger Breakout does not have any additional requirements, and all have been detailed in this SRS.

*<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>*

**Appendix A: Glossary**

OS – Operating System

WASD – W, A, S, and D keys used to move up, left, down, and right respectively

*<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>*

**Appendix B: Analysis Models**

*<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams*.>

**Appendix C: To Be Determined List**

*<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>*