Contents

1. Introduction ............................................................................................................................................... 5
   1.1 Game Summary ...................................................................................................................................... 5
       1.1.1 Characters .................................................................................................................................. 5
       1.1.2 Game Play .................................................................................................................................. 5

2. Purpose .................................................................................................................................................... 6

3. Functional Requirements .......................................................................................................................... 7
   3.1 Single-Player Mode .............................................................................................................................. 7
       3.1.1 Overview .................................................................................................................................. 7
       3.1.2 Game Play .................................................................................................................................. 7
   3.2 Multi-Player Mode .................................................................................................................................. 8
       3.2.1 Overview .................................................................................................................................. 8
       3.2.2 Usage ....................................................................................................................................... 8
       3.2.3 Game Play Features ................................................................................................................... 9
       3.2.4 Chat Features .......................................................................................................................... 9

4. Non-Functional Requirements .................................................................................................................. 10
   4.1 Description ......................................................................................................................................... 10
   4.2 Purpose ............................................................................................................................................. 10
   4.3 Overview ........................................................................................................................................... 10
   4.4 Usability ............................................................................................................................................ 10
       4.4.1 Graphical User Interface ........................................................................................................ 10
       4.4.2 Game Utilities .......................................................................................................................... 11
       4.4.3 Input Devices ........................................................................................................................... 11
   4.5 Performance ..................................................................................................................................... 11
       4.5.1 Speed ....................................................................................................................................... 11
       4.5.2 Multimedia Elements ............................................................................................................. 11
       4.5.3 Networking ................................................................................................................................ 12
       4.5.4 Security .................................................................................................................................. 12
   4.6 Maintainability ................................................................................................................................... 12
   4.7 Environmental Requirements .............................................................................................................. 12

5. Use Case Model ....................................................................................................................................... 13
   5.1 Description ......................................................................................................................................... 13
   5.2 Purpose ............................................................................................................................................. 13
   5.3 Notation ............................................................................................................................................ 13
   5.4 Example Use Cases ............................................................................................................................ 14
       5.4.1 Player Movement .................................................................................................................... 14
5.4.2 Walking on a Burger ................................................................. 15
5.4.3 Falling Burger Pieces ........................................................... 16
5.4.4 Pepper Throwing ................................................................. 17
5.4.5 Player Dies ......................................................................... 18
5.4.6 Attempt to Join a Multi-Player Game ................................. 19
5.4.7 Multi-Player Game Play ..................................................... 20

6. Expected Change ...................................................................... 21
6.1 Description ............................................................................ 21
1. Introduction

This document describes the requirements for the BurgerTime Clone project, based on CS 320 course guidelines.

Requirements specification is critical to ensure a successful project.

Furthermore, the requirements document is used:

- To define requirements and constraints on the solution. Clear requirements are necessary for a successful project because they define the project's goals. They clarify what is needed and help keep the team focused.
- As a basis for early solution sizing and estimates of cost.
- To assess the viability of the proposed solution.
- To drive design of the operational models. Non-functional requirements are frequently the most important determining factor of the architecture. Two systems with the same use cases but with very different Nonfunctional Requirements need very different architectures.
- As an input to component design.

1.1 Game Summary

We are to build a clone of the popular BurgerTime arcade game.

1.1.1 Characters

The game has an amusing cast of characters. The player character is a chef called Peter Pepper who is pursued by enemies named Mr. Hot Dog, Mr. Pickle and Mr. Egg.

1.1.2 Game Play

The object of the game is for Peter Pepper to assemble four giant burgers without colliding with any pursuing hot dogs, eggs or pickles (who appear in later stages). Peter Pepper has to climb up ladders and walk across ramps to get to each of the burger fixings which include buns, beef patties, lettuce leaves and sometimes tomato slices and cheese slices.

Peter Pepper has to walk across each of the burger fixings, so they will fall downward to the each of the lower ramps until they reach each of the plates below at the bottom of the screen. He must assemble all four burgers to advance to the next stage. As the hot dogs, eggs and pickles pursue him, he can stop them three different ways. He can throw a pinch of pepper on them to stun them and make them temporarily harmless, but his pinches of pepper are limited, so he will have to use it wisely. Additional pinches of pepper can be gained by getting bonus items the will occasionally appear on the screen temporarily including ice cream cones, french fries and cups of coffee.

Another way Peter Pepper can stop his pursuers is by crushing them with one of the burger fixings. He does this by dropping it onto them whenever they walk under it. The third way to stop them is by dropping them on one of the burger fixings whenever they walk onto it, this will also allow each of the burger fixings to fall two, three or four floors more depending on how many of them are on it. However the pursuers are put out of action, they will continue to walk around in each stage until all the burgers have been completely assembled.
2. Purpose

The purpose of this document will be to define and document the requirements that must be met by the new solution. The document will be used to gain consensus between the client and Team Beta on the requirements and it will provide the foundation for all future project efforts.

The objective of this deliverable is to outline the requirements that must be met by the new solution. Team Beta will perform the following:

- Identify the different types of intended users for solution and the key characteristics for each user type.
- Identify the planned information exchanges between solution and the external entities that will need to interface with the solution, including the primary inputs and outputs for the exchanges.
- Identify the key requirements for the solution:
- Identify the list of use cases that describe the functional requirements.
- Identify the list of usability requirements.
- Identify the list of service level requirements, system constraints, and other technical properties for the solution.
3. Functional Requirements

This chapter describes the requirements of the BurgerTime Clone for Release 1.0. Each requirement has been assigned a unique identifier.

Any initial estimates in this document are expressed in Man Months. Any initial estimates include only the development effort and should be used to compare the requirements with each other. When the proposal is created, all other project costs will be added (costs for requirements document, proposal, quality assurance, system test, support during user acceptance test).

This document will contain the following:

- A list of the identified service level requirements, system constraints, and other technical properties for the solution. Specific measurements and/or criteria may not be defined for these requirements in this phase.
- The list of identified usability requirements defined for the solution. Specific measurements and/or criteria may not be defined for these requirements in this phase.
- A list of the business rules and/or policies that must be supported by the solution.
- A set of user profiles that define the different types of user groups for the planned solution and the key characteristics of each group.

3.1 Single-Player Mode

3.1.1 Overview

The role of the game is a chef and the goal is to complete burgers to move forward to next level. There are burger pieces arranged all over the map and the chef must get each piece of the burger to fall down to the plate below the lowest reachable platform. Once all of the burgers have been completed, the player moves on to the next level. There are six different levels and after the sixth has been completed, it loops back to the first. The player plays until out of lives.

3.1.2 Game Play

Movement:

The player controls the chef to move left and right on platforms and up and down on ladders. Chef cannot move off of the end of a platform and he may not jump off ladders. The player may pause the game and freeze everything in the game.

Burger Pieces:

The chef makes each burger piece fall by running over it. When the chef has run over a section of a burger piece, that section sinks slightly. When the entire piece has been run over, the piece falls to the next platform. If another burger piece occupies that spot, it makes that piece fall down a level.

Enemies:

Enemies enter the map from the sides when the game begins. There are three types of enemies on the map. There are sausages, eggs and pickles. The sausages are fast and follow set paths. The eggs are slow but follow the chef. The pickles are fast and follow the chef. The chef dies if he comes in contact with any of the enemies.

Enemies can be killed in two ways. If an enemy is under a falling burger piece, he is squished and killed. If an enemy is standing on a falling burger piece, he is killed and the piece falls 3 levels instead of 1. When an enemy dies, a new one of the same type comes to take its place.
**Pepper:**
The chef can defend himself by throwing pepper. The chef can use the pepper to cause an enemy immediately in front of him to go into a 5 second sneezing fit. While the enemy is having his sneezing fit, the chef can pass through it unharmed. The chef throws pepper in the direction he is facing. That means he can throw left and right on platforms, but he cannot throw pepper at all while on ladders. The chef begins with 5 pepper throws and they are not restored upon death.

**Death:**
When the chef dies, he is charged 1 life and returned to the starting point of the level. Enemies are also reset to their starting position but the burger pieces remain where they are, and retain their current “walked on” status. The chef begins with 4 lives and if he runs out, the game is over. If the player’s score reaches the minimum score among the current high-score list, he/she will be prompted to input 3-letter initials upon game over.

**Scoring:**
Players earn points by completing burgers, killing enemies, and getting bonuses. There are 3 bonuses that appear on the map randomly: ice cream, French fries, and coffee. The point values are as follows:

<table>
<thead>
<tr>
<th>Event</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Completing a burger</td>
<td>5000</td>
</tr>
<tr>
<td>Killing a Sausage</td>
<td>100</td>
</tr>
<tr>
<td>Killing an Egg</td>
<td>200</td>
</tr>
<tr>
<td>Killing a Pickle</td>
<td>500</td>
</tr>
<tr>
<td>Ice Cream</td>
<td>500</td>
</tr>
<tr>
<td>French Fries:</td>
<td>700 + 1 extra pepper throw</td>
</tr>
<tr>
<td>Coffee:</td>
<td>1000 + 2 extra pepper throws</td>
</tr>
</tbody>
</table>

Accumulating points grants extra rewards:

<table>
<thead>
<tr>
<th>Points</th>
<th>Reward</th>
</tr>
</thead>
<tbody>
<tr>
<td>10,000</td>
<td>1 extra pepper throw</td>
</tr>
<tr>
<td>20,000</td>
<td>1 extra life</td>
</tr>
</tbody>
</table>

The game keeps track of the 20 highest scores, along with 3-letter initials.

### 3.2 Multi-Player Mode

#### 3.2.1 Overview

The player can play this game online with another person by choosing “Play Online” option at the start menu. Players can connect to and disconnect from the online game at anytime. Players may chat in a separate chat window. However, either player may opt to disable the chat.

#### 3.2.2 Usage

When the player chooses the “Play Online” option at the start menu, the game will ask the player either to create a session to wait for people to join, or request to join someone else’s session.
In the first case, the player may choose to start in single player mode immediately, play the game while wait for others to join. Or the player may choose to wait until someone request to join his/her game and start the game together. When another player is trying to connect to the game, the player will receive a prompt asking for his/her confirmation/permission for connection.

In the second case, i.e., the player chooses to join an existing session, he/she will need to enter the other player's IP address manually, and wait for confirmation from the other player. Once become accepted, the player will begin to play the current game level which the other player is playing.

When one player leaves the game or dies, the game turns into an online version of 1-player mode. However, the player can choose to disconnect and continue the game in offline mode. Or, continue the online mode and accept connection request, but not send request to join.

### 3.2.3 Game Play Features

Each player basically plays the game the same way as in single player mode, with the following changes:

1. The player can throw pepper to each other. The player who throws pepper first will make the other player sneeze and freeze for 5 seconds, i.e., same effect as if throws pepper to the computer characters (pickle, egg, sausage).
2. The player can kill the other player the same way he/she kills the computer characters, i.e., drop a piece of burger on the character or is standing on a dropping piece.
3. If one chef dies and that chef still has extra lives, the game will continue its current state, i.e., enemies will not be reset, only the chef will be reset to its entry position and start in a protected mode for the first 5 seconds.
4. In 2-player mode, the player can pause the game for a maximum of 120 seconds. After that the game will continue automatically. (In contrast to single player mode, where the player can pause the game for an arbitrary amount of time.)
5. If one player reaches the records in the high-score list, the player will be prompted to enter his/her 3-letter initials when the game is over. The player may have to wait for the other one to finish the game. The high-score list will be synchronized and merged so that it stores the first 20 highest scores among both players' records.
6. In 2-player mode, the player cannot save the game or load a stored game scenario, or jump to a different level. (In 1-player mode, such operations are possible under some special mode.)
7. The 2 chefs will have different appearance (such as color), so each player can identify which chef is under his/her control.

### 3.2.4 Chat Features

Players can chat in a separate chat window while playing online. The chat window looks like a MSN or Yahoo messenger. The chat window is attached to the main game window such that it can be moved, resized by the player. Either player may choose to disable chat. If chat is disabled after the chat window is open, or if any player disconnect from the game, the chat window will become disabled and can no longer accept input or refresh.
4. Non-Functional Requirements

4.1 Description

The system context represents the entire solution as a single object (black box) or process in the center of the diagram and identifies the interfaces between the solution and external entities. Shown as a diagram, this representation defines the solution and identifies the information and control flows that cross the system boundary.

The system context highlights important characteristics of the solution: users, external systems, batch inputs and outputs, and external devices.

- External events to which the solution must respond
- Events that the solution generates that affect external entities
- Data that the solution receives from the outside world and that must be processed in some way
- Data produced by the solution and sent to the outside world

4.2 Purpose

The purpose of the system context document is:

- To clarify and confirm the environment in which the solution has to operate.
- To provide the details at an adequate level to allow the creation of the relevant technical specification.
- Verify that the information flows between the solution to be installed and external entities are in agreement with any business process or context diagrams.

4.3 Overview

This section address general issues involved with the game design, but may not be particular to the BurgerTime game. This includes various usability, performances, security, maintainability, and environmental requirements for the game.

4.4 Usability

4.4.1 Graphical User Interface

The game will be played within a window with an inner game-playing area of 640x480 pixels, with a menu bar above it. This window size will fit most modern monitors which usually have a resolution of 800x600 or higher.
4.4.2 Game Utilities

Through the main menu, user can perform common game-playing related operations, such as start a game, pause/resume game, save/load game, screenshot, video recording, turn on/off audio, turn on/off chat, look for help about the game.

4.4.3 Input Devices

The game uses computer keyboard as its default input device. The control keys (e.g., arrow keys for direction) can be remapped with user-defined keys. The game may support joystick or game pad in the future.

4.5 Performance

4.5.1 Speed

The game should interact with player in a reasonably fast speed, at least the same speed of its original arcade version.

4.5.2 Multimedia Elements

The video refresh rate should be 30 frames per second or higher so the animation looks natural and smooth. If audio or sound effects are used in the game, they should be sampled at moderate to high sampling rates and dynamic ranges (bits/sample) to maintain good quality.
4.5.3 Networking

Playing the game online requires reliable and reasonably fast Internet connection. If unexpected Internet disconnection happens, the player who loses the Internet connection will begin to play in offline 1-player mode. The player who does not lose Internet connection will be able to either play offline, or play online and accept connection request, but not send join request.

4.5.4 Security

High security level is not the interest of this game. Only unsophisticated security measures will be implemented: The game does not broadcast so only people who know the player’s IP address can connect to it. The player that waits for join request will see the requester’s IP address, and must confirm each connection request manually.

4.6 Maintainability

The game will be designed using standard Java 2 Platform API of the current JRE version, so it can be easily maintained and supported.

4.7 Environmental Requirements

Hardware requirements include: a computer keyboard as input device; to play the game online, the computer needs to have Internet connectivity; a monitor capable of handling 800×600 resolution and 16-bit colors is recommended.

Software requirements is that the computer must run a current implementation of the Java Virtual Machine, namely, JRE version 1.4.2 as of Feb 2005.
5. Use Case Model

5.1 Description
The use case model describes the functional requirements of the solution. The use case model consists of the following parts:

- Use Case Diagram
- Use Case Textual Description
- Activity Diagram

The model uses graphical symbols and text to specify how users in specific roles use the solution (i.e., use cases). The textual descriptions describe the use cases from a user’s point of view: they do not describe how the solution works internally, nor do they describe its internal structure or mechanisms.

5.2 Purpose
The main purpose of the use case model is to establish the boundary of the solution and fully state its functional capabilities with respect to the users. The use case model also:

- Provides a basis for communication between end-users and solution developers
- Is the primary driver for estimating the application development effort
- Provides a basis for planning the development of the releases
- Provides a basis for identifying objects, object functionality, interaction, and interfaces
- Provides the primary basis for defining the user interface requirements
- Provides a basis for defining test cases
- Serves as the basis for acceptance testing
- Provides a basis for producing user support materials and documentation

5.3 Notation
The graphical notation used in this document is adopted from Unified Modeling Language (UML), a non-proprietary, third generation modeling and specification language.
5.4 Example Use Cases

5.4.1 Player Movement

A player may move left or right while on platforms only. The platform must also extend in the direction of movement. The player can not fall over the edge. To move vertically, the player must be on a ladder or have a ladder in the direction they are attempting to move. If the player moves into contact with an enemy, the player will die (see 5.4.5). If the player walks over a bonus item, the respective points and/or peppers will be rewarded. Walking over any burger piece will result in a burger segment dropping (see 5.4.2).
5.4.2 Walking on a Burger

When a player walks over a burger, segments of the burger will drop. Once all segments of the burger have dropped, the entire burger piece will fall (see 5.4.3). The burger piece falls one level normally and three levels when an enemy is on top of the falling piece.
5.4.3 Falling Burger Pieces

A burger piece will fall from one platform to the platform below. Any enemies under the piece will die, giving points to the player. If the burger piece lands on another burger piece, the lower piece will now drop a level with the same rules applying to it.
5.4.4 Pepper Throwing

The player is only allowed a set number of peppers (with the ability to earn more). A pepper can only be thrown while on a platform. Any attempt to throw a pepper while on a ladder will be discarded. If the thrown pepper hits an enemy, the enemy will go into a sneezing fit allowing the player to walk by without being harmed.
### 5.4.5 Player Dies

**Single Player:**

When a player loses a life and has more available, the level is restarted. During a restart, the burger pieces remain in their positions, the player returns to the start position, and the enemies are removed to their start positions. If a player loses their last life, and they have not achieved a high score, the game resets. If they do achieve a high score, they are asked to enter it into the list, and then the game restarts.

**Multiplayer:**

When a player loses a life and has more available, the died player is reset to its starting position, and everything else (including the other player) continues moving. If only one player has lost their life, the game continues for the second player as normal. Once both players have lost all lives, their scores will be compared to the high score list, and updated as needed.
5.4.6 Attempt to Join a Multi-Player Game

To join a game the guest must know their prospective host’s IP address. The game at the specified IP address must be accepting connections for a game. Once the game request is sent out, if the host accepts the game, a multiplayer game begins (see 5.4.7). If the host rejects the game, the connection is broken.

Figure 7
5.4.7 Multi-Player Game Play

Once a multiplayer game has started, the two players’ high score lists are merged. The main game loop consists of transferring the game state (player location, points, and enemies) between users and updating the chat window. Player deaths follow the multiplayer rules (see 5.4.5). If at any point the connection between users is broken, the games will return to single player mode on both user machines. Once a game is completed, high scores are compared and adjusted as needed, and the high score lists between users is once again merged.

Figure 8
6. Expected Change

6.1 Description

We follow the requirement from the date that it was created, through all its changes. We minimize future confusion by recording the rationale for making major changes. When a requirement is deleted we record when and the rationale behind the deletion. The date that the requirement passes its quality checks, and who passed it, is also recorded.