Software Requirements Specification

for

SushiTale

**Version 1.1**

**SushiWorld**

**26/02/2023**

**Table of Contents**

[**1. Introduction (Section: 30%)**](#_heading=h.1fob9te) **1**

[**1.1 Product Scope**](#_heading=h.fhwlwvipi58r) **1**

[**1.2 References**](#_heading=h.55y8i76q4jy0) **1**

[**2. Product Perspective (Section: 25%)**](#_heading=h.tyjcwt) **1**

[**2.1 Product Functions**](#_heading=h.bwtpfnzh8ik6) **1**

[**2.2 Assumptions and Dependencies (Risk and Mitigations)**](#_heading=h.167hsxaj6guo) **2**

[**3. External Interface Requirements (Section: 5%)**](#_heading=h.4d34og8) **2**

[**3.1 User Interfaces (UI Research)**](#_heading=h.jiu6bwdg7xsb) **2**

[**4. System Features (Section: 40%)**](#_heading=h.9roicp6392e9) **3**

[**4.1 Shop Room**](#_heading=h.6h2yo0a18c9t) **3**

[**4.2 Rest Room**](#_heading=h.8e25nd9pu926) **3**

[**4.3 Arrow Battle Room**](#_heading=h.najhr23h1g9g) **3**

[**4.4 Platform Battle Room**](#_heading=h.9wpbeaj7onhh) **4**

[**4.5 Final Boss Room**](#_heading=h.p62tbmovo5hz) **4**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  | Resubmission | 1.1 |
|  |  |  |  |

# Introduction (Section: 30%)

## Product Scope

Sushi Tale, a role-playing game that takes you on a culinary adventure through a fantasy world filled with delicious sushi dishes. The game will be a single-player, turn-based RPG that takes place on a randomly generated map. The player will control a character who must explore the map, defeat enemies, and collect ingredients to prepare for the final boss battle.

## References

Room Mechanics Referenced from “Slay the Spire”

Slay the Spire is a roguelike deck-building video game that was developed by the American studio MegaCrit. The game revolves around the player collecting cards and building a powerful deck to defeat enemies so as to survive. The player starts on a map with a basic deck of cards and climbs “The Spire”, going through all the different rooms while trying to stay alive. The player then will encounter many bosses, enemies and events along the way depending on the route of rooms they have chosen.

The content of each room and the route is procedurally generated therefore giving a different experience every time. Rooms that have monsters which vary in stats that rewards the player after defeating them, the choice to heal or upgrade cards, shops to purchase cards and more upgrades, treasure with random items and events that would affect the player in various ways. This would require careful planning, strategic decision-making and some luck to make it all the way.

This game gave us the idea to make a roguelike video game where the player will go through a map that has many different rooms. We would make different rooms like Battle, Shop, Rest and Events similar to the types of rooms available from the game.



Battle Mechanics Referenced from “Audition Online”

Audition Online is a casual online multiplayer rhythm game. T3 Entertainment created it, and it was published in 2006. Players can create avatars, dance to music, and engage in virtual dance contests with one another. The game includes both solo and couple dance modes, as well as a number of music genres, including pop, hip-hop, and ballads. The game places a big emphasis on social interaction by letting users chat, establish dance squads, and engage in dance fights.

The basic game is played by pressing the arrows that are shown on the screen (apart from the red Chance arrows, which the player must press the opposite of), and pressing spacebar or the control key on every fourth beat of the music. The more precisely the spacebar or control key is hit, the higher the score, with "PERFECT" being the best, "GREAT" and "COOL" still being considered good enough, "BAD" receiving a lesser score, and "MISS" receiving no score. You won't be able to play for the next move if you receive a "MISS." The player with the most points at the time of the game will have their character in front of the camera and in the center of the stage.

We were inspired by the mechanics of “Audition Online” gameplay where the player has to use arrow keys to dance and would implement that mechanic into our battle rooms where we press the arrow keys in quick succession to damage and defeat the mobs.



https://thevideogamedatabase.fandom.com/wiki/Audition\_Online

https://videogamesftvms2015.wordpress.com/2015/01/18/audition-dance-battle-game-dance-in-the-real-world/

Platform Battleroom/Boss Mechanics Referenced from “Cuphead”

"Cuphead" is a well-known indie video game created by StudioMDHR Entertainment. It was launched in 2017 for Microsoft Windows and Xbox One. The game is famous for its 1930s-style cartoon graphics, tough difficulty, and unusual run-and-gun action.

In "Cuphead," you play as the titular Cuphead, who, along with his brother Mugman, has made a bargain with the devil and must collect the soul contracts of other lost souls to settle their obligation. The game is separated into stages, and each level's boss encounter represents its conclusion.

Combat takes up the majority of Cuphead. You may anticipate doing a lot of shooting in Cuphead because it is a side-scrolling shooter. Although it can take many different forms, combat always has a similar structure. Use your dodge/defense as much as you can and fire enemies before they shoot you. You lose one HP point for every hit you take. The level resets to the beginning when all of the player's HP is lost.

For the platformer battle rooms and boss battles, we were inspired by the run-and-gun action of Cuphead. We would not be using the side-scrolling part of the map but rather have one main platform and battle the boss by shooting and dodging its attacks.



**Links:**

<https://holdtoreset.com/cuphead-gameplay-basics-guide/>

<https://www.blogdot.tv/cuphead-clip-joint-calamity-boss-fight-gameplay/>

<https://slay-the-spire.fandom.com/wiki/Slay_the_Spire_Wiki>

# Product Perspective (Section: 25%)

## Product Functions

* Main Menu
* For the player to start the game, choose levels, change settings and exit the game.
* Map/Level
* The player has to navigate the map from the start to get to the end through exploring different rooms and progress through the level.
* Room Mechanics
* There should be 4 different room types
* Battle room
  + Players battle enemies to gain currency/upgrades.
* Rest room
  + Players can choose to heal up or upgrade an ingredient acquired
* Event room
  + A random event would happen that would benefit or harm the player
* Shop room
  + Allow Players to obtain items using currencies gained.
  + Players can have a choice to purchase one of the ingredients with their currency
  + Once they leave the room, the shop will close forever
* Instructions
  + The player can read the instructions page to understand the objective and the controls of the game.
* Save and exit
  + The current game is being saved, so that players can resume progress.
* Gameplay Mechanics
  + When a new game starts, players will be spawned onto a 8 x 6 tiled map, each tile representing a room that players can enter once.
  + Player will be spawned on the bottom left tile.
  + The Final boss room will be on the top right tile.
  + Player has to choose which direction to go, and the aim is to reach the final boss room
  + The player will have a set of stats e.g health, strength, which will affect the performance during the combat.
  + The player will be able to upgrade their stats by going to specific rooms.
  + When the player runs out of health points, all progress is lost and needs to start a new game.
* Gear progression
* There should be 4 upgradable ingredients that could be upgraded from level 1.
  + Salmon - Increase max HP
  + Tuna - Increase time in battle (Buff applied in Arrow battle room)
    - Allow players to have more time to enter the arrow inputs
  + Swordfish - Increase damage dealt to boss.
  + Squid - Increase Evade
    - In Arrow Battle room
      * Chance to not receive damage that turn
    - In Platform Battle room
      * Chance to not get damaged by boss’s attacks

* Battle Mechanics
  + Arrow Key room
    - When the battle begins, players will be given a set of arrow keys to input in order before the time runs out.
    - Based on the number of correct inputs, it would then determine the damage taken to the player/enemy depending on who’s turn to attack/defend.
  + Platform Battle Room
    - Player Shoots bullets, jumping up and down to dodge boss attacks
    - Bullets deal damage to bosses.
    - Varying boss attack patterns.
* Level Design
  + The difficulty level will decide the final boss stats
  + The difficulty of the combat will increase based on the number of rooms explored.
    - Inverted keys
    - Higher enemy stats
    - Unrevealed keys that reveal themselves when approaching it

## Assumptions and Dependencies (Risk and Mitigations)

* Random room generation
* Risk: random rooms can’t be implemented,
* Mitigation: we can change them to fixed rooms
* Time taken to clear the game
* Risk: it might take too long for the player to clear the game
* Mitigation: reduce the number of possible rooms
* Event room
* Risk: Unable to integrate into our game
* Mitigation : Remove this feature
* Boss room
* Risk: Unable to implement enhanced difficulty.
* Mitigation: Increase boss difficulty stat wise instead of mechanics.

Assumptions

1. Development delays: The game development might take longer.
2. Wrong damage computations, fixed damage based on stats

Dependencies

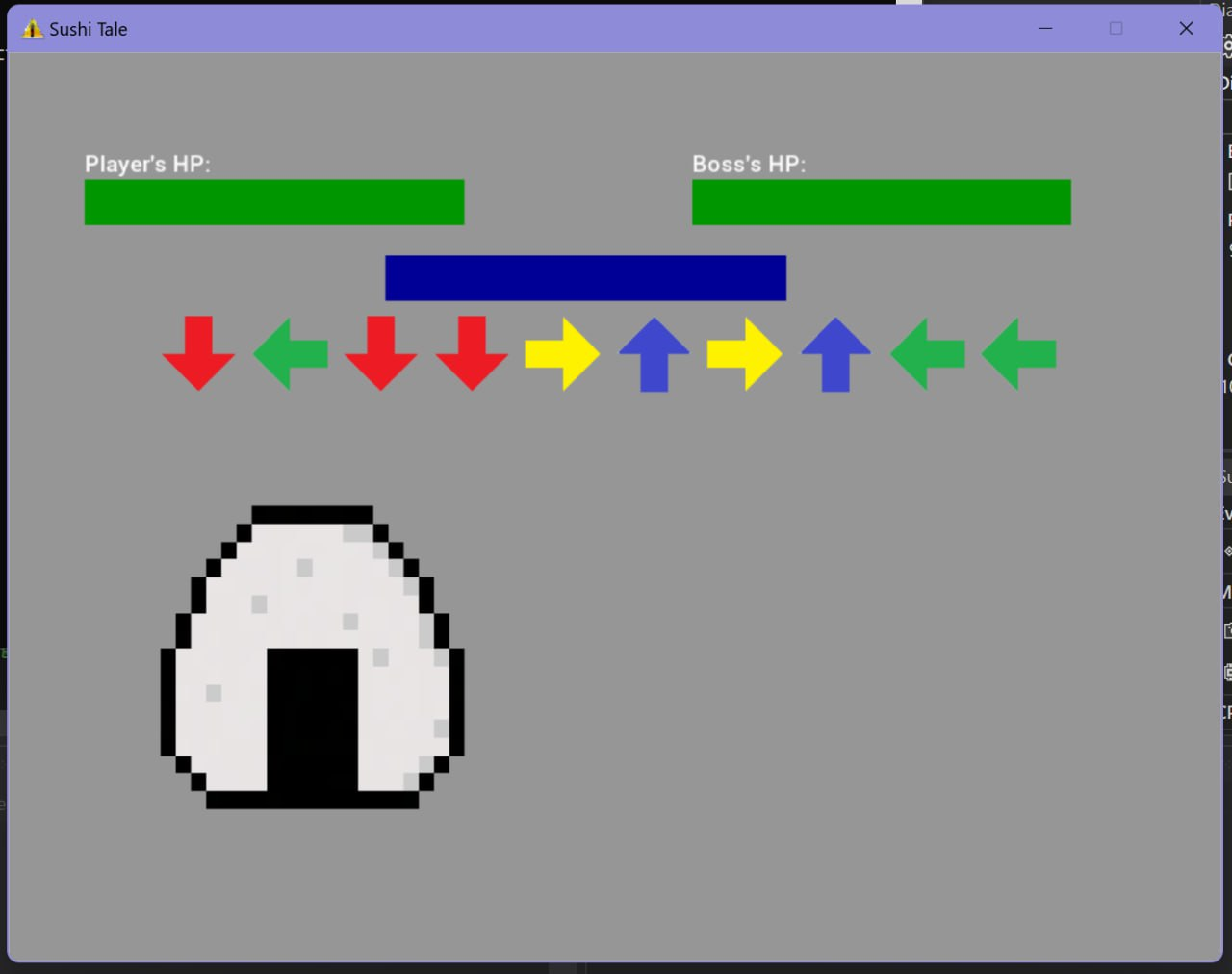
1. None.

# External Interface Requirements (Section: 5%)

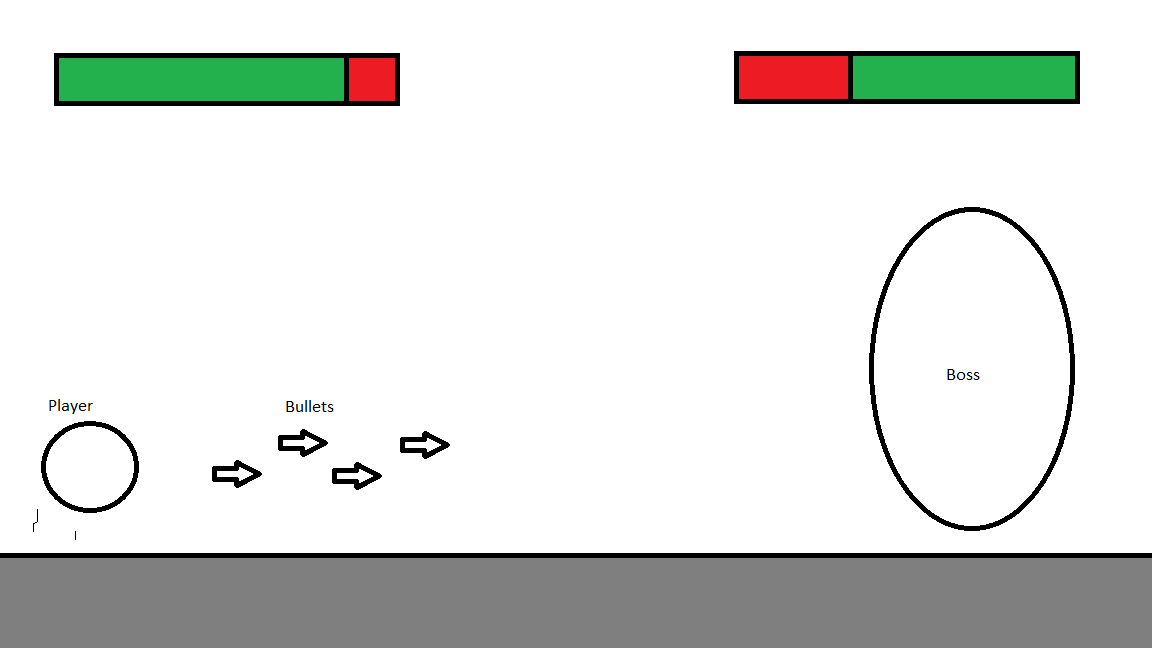
## User Interfaces (UI Research)

1. //The How to Play button will be included in the main menu, explaining how to play the //game.
2. //Player stats and item levels will be accessible while exploring the map.
3. //Pause for players to take a break
4. //Option to allow players to choose out of two randomly generated room types when //entering a new room.

Arrow Battle Room



Platform Battle Room



# System Features (Section: 40%)

# Game Engine

Game State Manager

* The Game State Manager will manage all the states in the game and change the states such as the main menu, game, pause screen, etc.

Audio Manager

* The Audio Manager will handle all the audio in the game such as the background music and sound effects to enhance the gaming experience.

Assets Manager

* Handles loading images, processing sprite sheets into textures or animations.
* Loads and store fonts to be used through program runtime

File IO

* Allows player’s progress to be saved, and to be resumed at a later time

# Shop Room

Description

* Player would be able to choose if they want to buy an ingredient

Stimulus/Response Sequences

* Upon entering the shop, the player is able to see 2 choices of ingredients/items
* Each shop will have a different price
* Based on player’s money, they have a choice to buy or leave
* Once sold, no refunds
* Once the player leaves, the shop will be closed till the player finds another shop.

Functional Requirements

* REQ-1: Shop would sell a random ingredient that the player has not gotten
  + if all ingredients were present, the shop would sell upgrades
* REQ-2: The shop would disappear once the player leaves the room

# Rest Room

Description

* Player would be able to choose to heal or upgrade

Stimulus/Response Sequences

* Upon entering the room, the player will have to choose to heal up or upgrade an ingredient
* Heal up would heal the player a fixed value of health
* Upgrade an ingredient
  + Player would be required to fill a bar by spamming a key, this would decide their upgrade passing rate

Functional Requirements

* REQ-1: Rest room is a one-time use
* REQ-2: Create the upgrade mechanics. -> The bar would increase per key pressed, the bar would also be decreasing at a rate depending on the level of ingredient. The upgrade will then pass or fail based on the percentage of the bar filled
* REQ-3: Heal up the player's health

# Arrow Battle Room

Description

* Battle against enemies
* Dealing damage to the enemy using arrow keys

Stimulus/Response Sequences

* Upon entering an arrow key battle room, the enemy will have a specific set of fixed attack power. During every turn, the player will enter a set of arrow keys of varying orientations within the set time.
* If the Player successfully inputs all arrow keys, full damage will be dealt to the enemy.
* If the Player fails to clear within the time limit, the player will receive damage equivalent to enemy attack power minus the number of correctly pressed keys.
* If the Player fails any arrow input, the player will receive damage.

Functional Requirements

* REQ-1: Arrow Key generator -> function to create a new random set of arrow key inputs.
* REQ-2: Input Timer -> To check the time taken for the player to key in arrow keys
* REQ-3: Damage Calculation -> Compare player’s input, speed and accuracy of input to determine the damage to be dealt to the enemy. Other factors, such as the player's state may affect the damage value dealt.

## Platform Battle Room

Description

* Battle against enemies
* Dealing damage by shooting them while dodging

Stimulus/Response Sequences

* Upon entering the room, enemies would be on the right attacking the player to the left
* Player would then dodge/shoot until either side runs out of hp

Functional Requirements

* REQ-1: Create enemies attack patterns -> create several attack styles and random the pattern
* REQ-2: Platforming the area for the player to use.
* REQ-3: Physics affecting the player and bullets.

## Final Boss Room

Description

* Final room where the player fights against the boss

Stimulus/Response Sequences

* Upon entering the room, the player can choose from the previous type of battle room
* Difficulty would be increased based on player stats
* IF Player chooses Arrow
  + Rotating arrow
  + Arrow require multiple input
  + Reversed Arrow key (arrow key order starts from the back)
* IF Player choose Platform
  + Limit Player’s vision
  + Anti gravity
  + Create mobs

Functional Requirements

* REQ-1: At least one battle room type is implemented
* REQ-2: Physics systems must be implemented.
* REQ-3: Increase of difficulty compared to the whole game

**Appendix A: Glossary**

**Appendix B: Analysis Models**