**Software Requirements Specification**

**for**

**Tactical Chicken**

**Version 1.0 approved**

**Prepared by Bill Quain and Timothy Worthem**

**WIU CS 491**

**October 9th, 2022**

# Table of Contents

[**Table of Contents**](#_bg80uefdm0in) **2**

[**Revision History**](#_hcw17x0vs0e) **3**

[**1. Introduction**](#_rl68fmgxvvou) **4**

[1.1 Purpose](#_plpgh0b6qnxx) 4

[1.2 Document Conventions](#_1qpnjlyquh3y) 5

[1.3 Intended Audience and Reading suggestions](#_1tbbfihbgtjt) 5

[**2. Overall Description**](#_cxj2waqcmul5) **6**

[2.1 Product Perspective](#_lf9fk0xcyyu6) 6

[2.2 Product Features](#_9i2qwsl71ho1) 6

[2.3 User Classes and Characteristics](#_hi01c7mx9zpb) 6

[2.4 Operating Environment](#_oozojfrv5kzj) 7

[2.5 Design and Implementation Constraints](#_g51latb2cuze) 7

[2.6 User Documentation](#_16arrge8wkmx) 8

[2.7 Assumptions and Dependencies](#_hynjwz9lc7up) 8

[**3. Systems Features**](#_vy3ps0byt6o) **8**

[3.1 System Feature 1 (Popping up the window)](#_ph8uvb5bzhop) 8

[3.1.1 Description and priority](#_pp5961dvraqy) 8

[3.1.2 Stimulus/ response sequences](#_4hnoeytmwonr) 8

[3.1.3 Functional Requirements](#_o0owwcgomlw3) 8

[3.2 System Feature 2 (Menu which includes buttons)](#_sjh5181o4r48) 9

[3.2.1 Description and Priority](#_43eqllqi6j7m) 9

[3.2.2 Stimulus/Response Sequences](#_jsxn4t57nfiy) 9

[3.2.3 Functional Requirements](#_ddzf9d7y6i9o) 9

[3.3 System Feature 3(Shooting and Moving)](#_47mof5grmbc5) 9

[3.3.1 Description and Priority](#_g8bpe83gvxfg) 9

[3.3.2 Stimulus/Response Sequences](#_rqzhmj5e6nqn) 9

[3.3.3 Functional Requirements](#_3guir83fld2v) 9

[3.4 System Feature 4(attack player)](#_cbid6xw3vq62) 10

[3.4.1 Description and Priority](#_1jhfjuf3x6d0) 10

[3.4.2 Stimulus/Response Sequences](#_i0hewsgga6ze) 10

[3.4.3 Functional Requirements](#_jkbu9kzas5ra) 10

[3.5 System Feature 5(settings)](#_bu1uqvk32hkg) 10

[3.5.1 Description and Priority](#_rmhk4cq6v1se) 10

[3.5.2 Stimulus/Response Sequences](#_w4v5cufmduvd) 10

[3.5.3 Functional Requirements](#_idqiz5zgdg0x) 10

[**4. External Interface Requirements**](#_9pld3jl5rjm9) **11**

[4.1 User Interfaces](#_llylpz4r0lkv) 11

[4.2 Hardware Interfaces](#_k5kzrxj4tvd1) 11

[4.3 Software Interfaces](#_1v4s53iq7bn8) 11

[4.4 Communications Interfaces](#_fer9pvv792rw) 11

[**5. Other Nonfunctional Requirements**](#_5ct7iot2fx3t) **12**

[5.1 Performance Requirements](#_l5mwccds1k8i) 12

[5.2 Safety Requirements](#_mou5xa6bre1z) 12

[5.3 Security Requirements](#_jgio1wzh09t5) 12

[5.4 Software Quality Attributes](#_51f7r7zb9r27) 12

[**Appendix A: Glossary**](#_i5sfogp7lxzo) **13**

[**Appendix B: Analysis Models**](#_q3wi4g4za4jo) **13**

[**Appendix C: Issues List**](#_ft6m1zoqgx2s) **13**

# **Revision History**

| **Name** | **Date** | **Reason For Change** | **Version** |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |

# **1. Introduction**

## **1.1 Purpose**

The goal of Tactical Chicken is to provide a top down action game that features pixel art. The player should be able to navigate around a two dimensional space. The user will have the ability to shoot enemies while collecting keys to proceed to the next level.Throughout the levels, there will be different types of enemies. There will be the basic enemy as well as minibosses. This game is primarily targeted towards people seeking digital entertainment.

## **1.2 Document Conventions**

Times New Roman font used throughout the document. There are multiple chapters with sections, a table of contents, and multiple appendices.

## **1.3 Intended Audience and Reading suggestions**

This document is meant to be read by people such as developers, project managers, marketing staff, users, testers, and documentation writers. This document contains information regarding an introduction and description to the product, system features, external interface requirements, and other nonfunctional Requirements. Developers will want to read section two on overall description and section three on system features. Project managers and documentation writers will want to read this whole document. The marketing staff, users and tester should read sections one and two which is about an introduction and overall description of the product.

**1.4 Project Scope**

The purpose and benefits of Tactical Chicken is to provide an entertaining experience for the user. The objective of Tactical chicken is to implement a two dimensional level in which the user has to defeat the enemies to obtain keys to be able to move on to the next level. The Goal of this project is to provide a quality piece of software that entertains the user. This will be accomplished by using sound software engineering and game design practices.

**1.5 References**

Guide to UML Use Case Diagrams:

https://drawio-app.com/uml-use-case-diagrams-with-draw-io/

Guide to UML Activity Diagrams:

<https://about.draw.io/create-uml-activity-diagrams-in-draw-io/>

# **2. Overall Description**

## **2.1 Product Perspective**

The context and origin of Tactical Chicken is a new, self contained product. The inspiration of this game is based off the games Call of Duty Zombies Dead Ops Arcade and Robotron.

## **2.2 Product Features**

In Tactical Chicken the software allows the user to move the character around a level using the WASD keys and shoot using the up,down,left, and right arrow keys. The game also features multiple types of enemies. The player can change the chicken's appearance in the settings menu as well as difficulty settings. The player can pick up items in the level and add it to the characters inventory.

## **2.3 User Classes and Characteristics**

* Character (Parent class)
  + Player
  + Enemy
    - Miniboss
* World object(parent)
  + Walls
  + Obstacles
* Items(parent)
  + Weapons
  + keycards

## **2.4 Operating Environment**

The operating environment for Tactical Chicken will be Windows 10. Hardware requirements are a mouse and keyboard.

**2.5 Design and Implementation Constraints**

The game will only be made using the english language. The game will not support three dimensional graphics due to consequences of tools selected for the project. Performance due to limitations of the Python programming language.

## **2.6 User Documentation**

Instruction on how to play the game will be given through a short tutorial that will be integrated into the background art of the level. Throughout level 1 the player will encounter areas of the world in which they will be given hints on how to proceed.

## **2.7 Assumptions and Dependencies**

It is assumed that the player will have a system capable of running the software. A legal copy of Windows 10. Also, it is assumed that the user will have a mouse and keyboard. This software is being developed with third party software which includes Pygame, LDtk 2d level editor, Piskel pixel art program, and PyCharm IDE community edition.

# **3. Systems Features**

## **3.1 System Feature 1 (Popping up the window)**

### 3.1.1 Description and priority

The game must be able to open a window that will display the graphics. This is a very high priority.

### 3.1.2 Stimulus/ response sequences

The user will start the software which will open up a window to start the game.

### 3.1.3 Functional Requirements

The game will only run in one window at a time in a proposed resolution.

## **3.2 System Feature 2 (Menu which includes buttons)**

### 3.2.1 Description and Priority

Within the window, the user will be able to press different graphical buttons to navigate the game start menu. This is a high priority feature.

### 3.2.2 Stimulus/Response Sequences

On the start menu the user will have the option of start game, quit game, or settings. There will be game art in the background in the menu. If the user starts the game, the window will load level 1. If the user selects the quit button the program will close. If the user selects settings they will have the option to change the difficulty level and character appearance.

### 3.2.3 Functional Requirements

The menu must be capable of receiving mouse input to navigate the menu. Display graphics for the buttons and background art. Also, be able to terminate the software.

## **3.3 System Feature 3(Shooting and Moving)**

### 3.3.1 Description and Priority

The user should be able to fire the character's weapon in four different directions; up, down, left, and right. Projectiles will be able to damage enemy health and will not be able to pass through solid objects. Enemies are also capable of firing at the player causing damage to player health. The player should also be able to move in four different directions by using the WASD keys. Enemies will move about the level with limited mobility. Priority is high.

### 3.3.2 Stimulus/Response Sequences

Throughout the game the user will be able to shoot and move in one of four directions. This also applies to enemies. The user will be able to see the projectiles moving across the screen.

### 3.3.3 Functional Requirements

The game must be able to register keyboard input from the arrow keys: up, down, left, and right, and the move keys: w, a, s, and d. The window will have to display each projectile and character movement. Also each projectile must be able to be registered by both the player and enemy as damage.

## **3.4 System Feature 4(attack player)**

### 3.4.1 Description and Priority

The enemies in the game will have to have some kind of intelligence to attack the player. This will be done by having the enemies detect if the player is nearby, moving closer to the player and shooting them. This is a high priority

### 3.4.2 Stimulus/Response Sequences

Throughout the game the player will have to face enemies and those enemies will

have to have enough intelligence to be an entertaining challenge.

### 3.4.3 Functional Requirements

The enemies will detect the player based on an abstract idea of what they can “see” , meaning that the enemy will check a rectangle surrounding the enemy to see if the player is within that range before engaging. Enemies will move towards the player while firing at them. The enemies will fire on a set interval to make the game more fair to the player.

## **3.5 System Feature 5(settings)**

### 3.5.1 Description and Priority

Within the menu screen, the user has the option to edit the settings. The user can edit the character appearance and change the difficulty. This is a low priority.

### 3.5.2 Stimulus/Response Sequences

If the user selects the setting button, it will display the settings screen allowing the user to edit the appearance or difficulty. On that screen, the user can pick from pre-made character appearances. Also, the user will be able to pick the level of difficulty they want to play.

### 3.5.3 Functional Requirements

The game will have to be able to update the settings so when the game is played, the new settings are adjusted correctly. If the user changes the character appearance, the program will update the game with that appearance. Also, if the user changes the difficulty, the program will update the game to that difficulty.

# **4. External Interface Requirements**

## **4.1 User Interfaces**

The user will be interacting with the game window throughout the program. Within the game window the user will use the mouse to click buttons on the screen. While the user is playing the game they will be able to move and shoot in different directions. They will also be able to terminate the software. The user interface will consist mostly of buttons and character information.

## **4.2 Hardware Interfaces**

A mouse and keyboard are required to play the game. The input on the keyboard will

allow the player to move and shoot while the mouse will allow the user to interact with the game menu. A monitor with appropriate resolution is also required to see the graphics. A windows 10 computer will be needed to play the game.

## **4.3 Software Interfaces**

This product will require Windows 10 to run. Python 3.10 and pygame will also be required to display graphics and run the game logic.

## **4.4 Communications Interfaces**

This software will not use any type of communication interface or service.

# **5. Other Nonfunctional Requirements**

## **5.1 Performance Requirements**

This program will require at minimum windows 10, intel UHD integrated graphics, a system with 8gb of RAM, and enough free disk space to store the game.

## **5.2 Safety Requirements**

Game contains cartoon level violence which includes shooting enemies with projectiles. Players must be over the age of 10 years old.

## **5.3 Security Requirements**

There is an extremely low chance of security or privacy issues as no personal information is required to play the game.

## **5.4 Software Quality Attributes**

This software will be portable, reliable, maintainable, and usable. Python programs can run on just about any Windows 10 computer making it portable. Reliable is a value of the developers of the software. The game will be very usable given the simple design.

# **Appendix A: Glossary**

**Intel UHD integrated graphics** - a graphics processor integrated into select Intel Core processors for laptops

**LDtk -** 2D level editor

**Piskel** - pixel art program

**PyCharm IDE** - Integrated development environment dedicated to the Python programming language

**Pygame** - Game engine library for python

**Python** - general purpose programming language

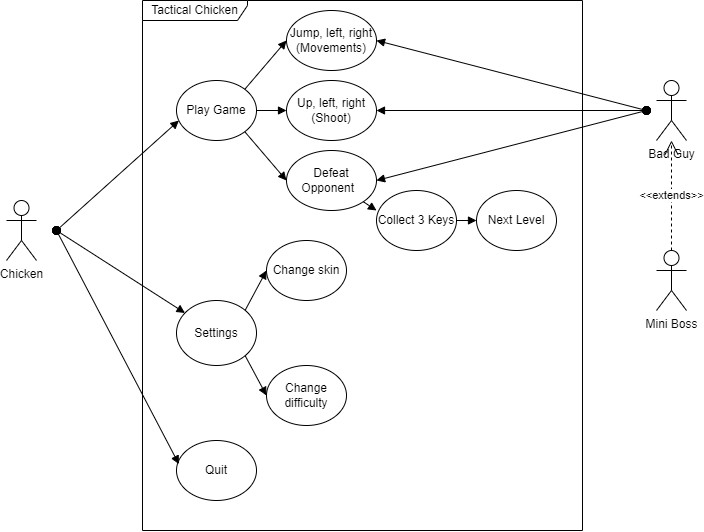
**RAM** - Random Access Memory

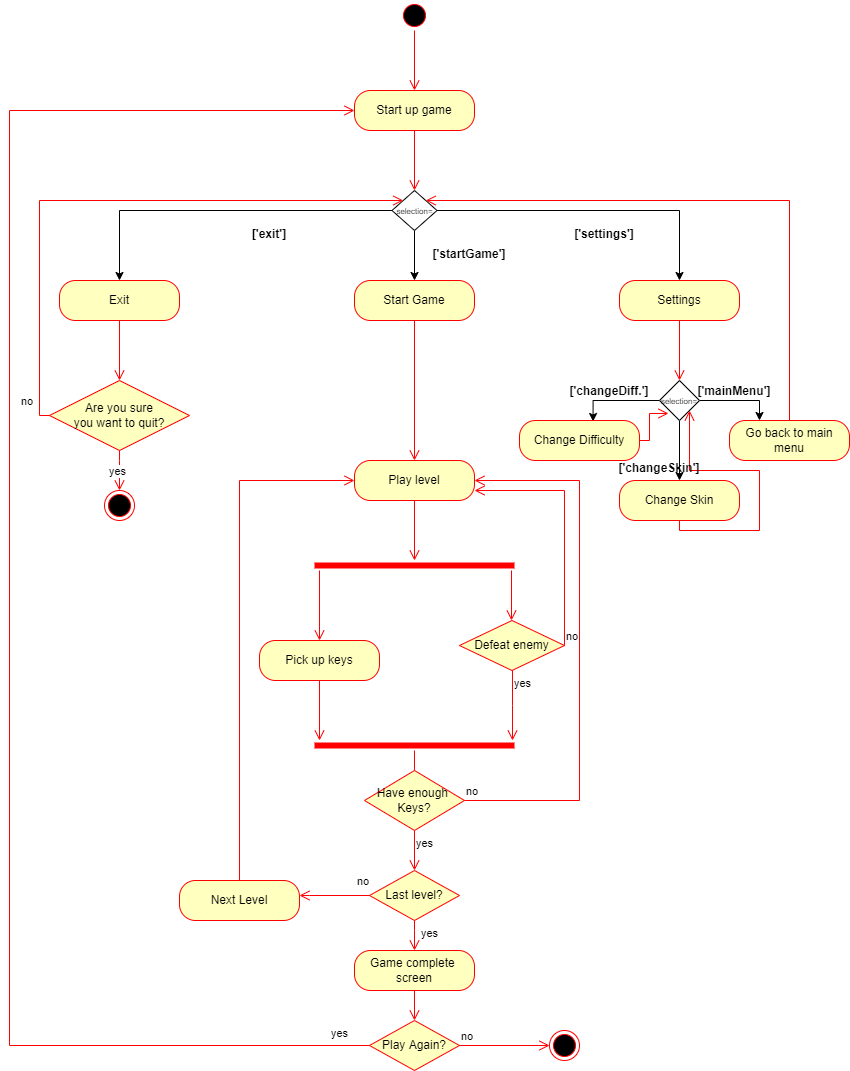
**Windows 10** - popular operating system

# 

# **Appendix B: Analysis Models**

# 





# **Appendix C: Issues List**

There are no issues discovered yet with the program.