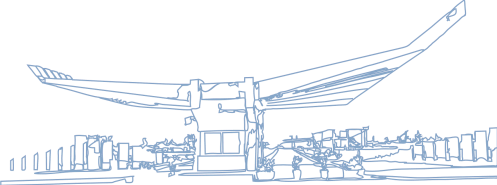
**INTRODUCTION TO SOFTWARE ENGINEERING**

**Snake Game**

****

**Team Members:**

**Engjëll Abazaj**

**Indrit Ferati**

**Kristi Samara**

**Snake Game Requirements Specification**

Table of Contents

1. EXECUTIVE SUMMARY

1.1 PROJECT OVERVIEW

1.2 PURPOSE AND SCOPE OF THIS SPECIFICATION

2. PRODUCT/SERVICE DESCRIPTION

2.1 PRODUCT CONTEXT

2.2 USER CHARACTERISTICS

2.3 ASSUMPTIONS

2.4 CONSTRAINTS

2.5 DEPENDENCIES

3. REQUIREMENTS

3.1 FUNCTIONAL REQUIREMENTS

3.2 NON-FUNCTIONAL REQUIREMENTS

3.2.1 User Interface Requirements

3.2.2 Usability

3.2.3 Performance

3.2.4 Manageability/Maintainability

3.2.5 Security

3.2.6 Standards Compliance

3.2.7 Other Non-Functional Requirements

3.3 DOMAIN REQUIREMENTS

4. DESIGN THINKING METHODOLOGIES

4.1 Negotiation

4.2 Empathy

4.3 Noticing

4.4 GUI

5. SOFTWARE DESIGN

5.1 Use Case

5.2 State Diagram

5.3 Class Diagram

APPENDIX

APPENDIX A.DEFINITIONS, ACRONYMS, AND ABBREVIATIONS

APPENDIX B.REFERENCES

APPENDIX C. ORGANIZING THE REQUIREMENTS

1. Executive Summary

***1.1 Project Overview***

A well-known and traditional game is the snake game. It involves a snake that consumes a specific item that happens to emerge randomly. As it consumes more food, the snake grows bigger and moves faster. The snake's own body or walls should not be touched. Due to the lack of specific skill requirements, the game is appropriate for almost any age group. Everyone may play it on a computer or a mobile device, and it is highly entertaining and interesting.

***1.2 Purpose and Scope of this Specification***

Describe the purpose of this specification and its intended audience. Include a description of what is within the scope and what is outside of the scope of these specifications.

**In scope**

The following specifications describe the purpose of this product and how it intends to be used:

* The purpose of the game is to provide an engaging and fun gameplay experience for players.
* Objective: Moving the snake and consuming as much food as you can while avoiding collisions with the walls or the snake's own body is the goal of the game.
* Gameplay: Grid-based gameplay will be used, and the snake will travel in a predetermined route across the grid. If the snake strikes a wall or its own body, the game is over.
* Controls: The snake can be moved in the game by using the arrow keys or other designated keys.
* Scoring system: The snake will receive points for each food item it consumes, with extra points if it consumes several food items quickly.
* Power-ups or bonuses: The snake may be able to collect special goods that grant perks like faster speed or more points.
* Technical requirements: The game must function on desktops and mobile platforms and might need particular programming languages or libraries.
* User interface: The interface of the game will be user-friendly, and the player will receive feedback that is concise and simple to understand.

**Out of Scope**

* Multiplayer functionality: It might be beyond the scope of a basic game to include a multiplayer option where players can compete against one another.
* Advanced graphics: For a simple game, creating advanced graphics or animations might not be necessary because it would take more time and resources.
* Customizable game settings: Customizable game options like altering the snake's size or pace might not be appropriate for a simple version of the game.
* Localization: If the game is designed for a particular audience or region, adapting it to multiple languages can be outside the scope of the project.
* In-game purchases or ads: Implementing in-game purchases or ads may be out of scope if the game is intended to be a simple and free experience.

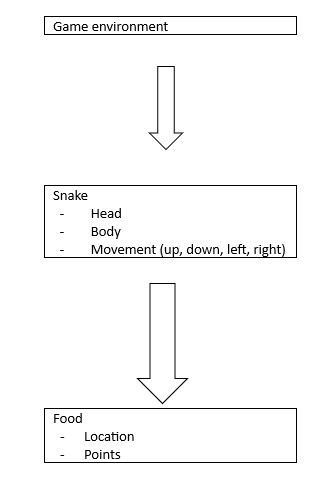
**2. Product/Service Description**

***2.1 Product Context***

The snake game is a standalone, self-contained game that ordinarily has no interaction with other games or associated platforms. The game is intended to be played independently on a variety of platforms and gadgets without the aid of external interfaces or connections.

While the snake game may resemble other games, including retro arcade favorites like Pac-Man or cutting-edge mobile games like Candy Crush, it differs in terms of gameplay mechanics and goals.

Here is a diagram that shows the major components of the snake game:



The area of the game where the snake and food items are located is called the playing field. The snake item may move in many directions and is made up of head and body segments. The food item is generated at random on the playing surface and awards points to the snake when it consumes it.

Since the snake game is meant to be a straightforward and self-contained gaming experience, it typically lacks significant external interfaces or connections.

***2.2 User Characteristics***

The snake game will have three types of users that will be differentiated by their roles:

● Student

● Faculty/Staff

● Other

**Student**

Age: 10-18 years old

Experience: Casual gamer, may have played similar games before

Technical Expertise: Basic familiarity with gaming and gaming equipment

Characteristics: May be looking for a simple and enjoyable game to play in their free time or to take a break from their academic work. Possibly enjoys playing against friends.

**Faculty/Staff**

Age: 30-60 years old

Experience: Casual gamer, may not have played games before

Technical Expertise: Basic familiarity with gaming and gaming equipment

Characteristics: May be looking for a fun and engaging game to play during breaks or downtime. May appreciate games that don't take a lot of practice and don't demand a lot of time.

**Other**

Age: Any age

Experience: Casual gamer, may have played similar games before

Technical Expertise: Basic familiarity with gaming and gaming equipment

Characteristics: May be looking for a simple and fun game to play during your free time as a way to relax. May enjoy games that don't have a steep learning curve or time investment.

***2.3 Assumptions***

* Availability of compatible devices: It is anticipated that the game will be played on equipment that meets all of the system requirements, including those screen size, processor speed, and graphics.
* User expertise: Users with a basic understanding of video games and gaming systems are regarded as being able to play the game. Users with more sophisticated technological skills could have different needs or expectations from others for the game.
* Availability of internet connectivity: It is anticipated that the game can be enjoyed without a reliable internet connection. However, some features, like multiplayer modes or online leaderboards, might need an internet connection.
* Operating system compatibility: It is expected that the game works with particular operating systems, such as Windows, macOS, iOS, or Android. The specifications might need to be adjusted if the game is incompatible with a certain operating system.
* Availability of necessary software: It is believed that the game will be played on hardware that has the required software, such as game engines or web browsers. The specifications might need to be adjusted if the required software is not available.

***2.4 Constraints***

* Parallel operation with an old system: The design possibilities may be limited by compatibility difficulties or the requirement to integrate with outdated technology if the snake game needs to run in parallel with an older system.
* Audit functions: The design possibilities may be limited if the snake game has audit features, such as an audit trail or log files, due to the requirement to capture and retain pertinent data regarding user activities, system events, or data changes.
* Access, management, and security: The design possibilities may be limited by the requirement to integrate safe authentication and authorization systems, data encryption, or other security measures if the snake game requires access control, user management, or security features.
* Criticality of the application: The design options may be limited if the snake game is essential for a specific business or user function due to the requirement for high availability, fault tolerance, or disaster recovery capabilities.
* System resource constraints: The design choices may be limited by the requirement to maximize performance and reduce resource utilization if the snake game has disk space or other resource restrictions.
* Other design constraints: The snake game can have to adhere to design guidelines or other restrictions, such as architecture, programming language, or framework selections. The design possibilities accessible to developers may be impacted by these limitations.

***2.5 Dependencies***

* Platform dependencies: Depending on the platform or operating system—such as Windows, macOS, iOS, or Android—that the snake game will be running on, the system requirements may change.
* Framework dependencies: The specifications for the snake game may vary depending on the framework or coding language chosen for development, such as Unity, Unreal Engine, or HTML5.
* Resource dependencies: Depending on the availability of resources like disk space, memory, or network bandwidth, the snake game's system requirements may change.
* Third-party dependencies: The specifications for the snake game might be reliant on third-party applications or services, including a game engine, a cloud storage service, or a payment processing system.
* Time dependencies: Depending on the amount of time available to create and deploy the game, such as a deadline or launch date, the requirements for the snake game may change.
* Design dependencies: The specifications for the snake game may be influenced by how other parts or systems, including user interfaces, game mechanics, or graphics, are created.