# Requirement Analysis

## Introduction

The Project deals with the running snake game using c programming on Linux and Windows operating systems. Players can play snake games just running make run command on terminal or CMD. As players score more and more points the speed of the snake will increases. This project aimed at the users to play fewer games outside the home. This will prevent big injuries to the players.

## System Analysis (Research):

This system is basically concerned with the injuries of children. to prevent injury kids can play the snake game in their operating system. it doesn't matter. which operating system they are using.

## System Features:

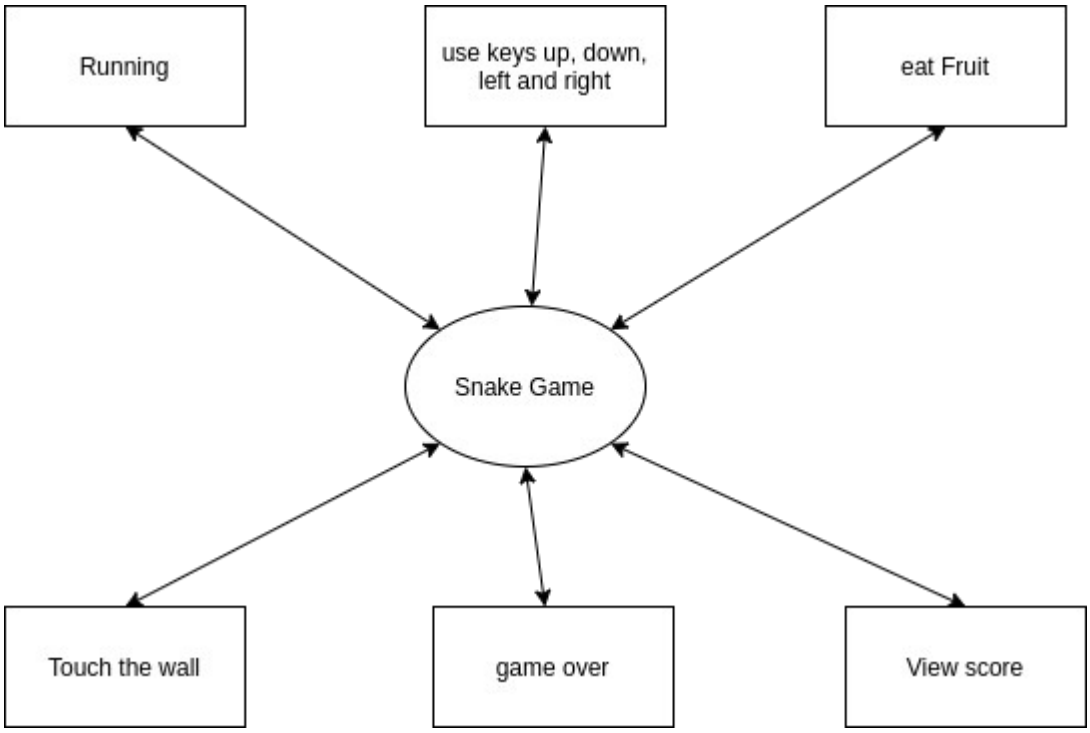
- System will allow the user to play snake game
- User can play for unlimited time.
- User will be able to play on linux as well as windows operating systems.
- User can see their scores, speed and version of game.
- Display the details of the creator.
- User can stop game whenever he/ she want.

## Benefits:

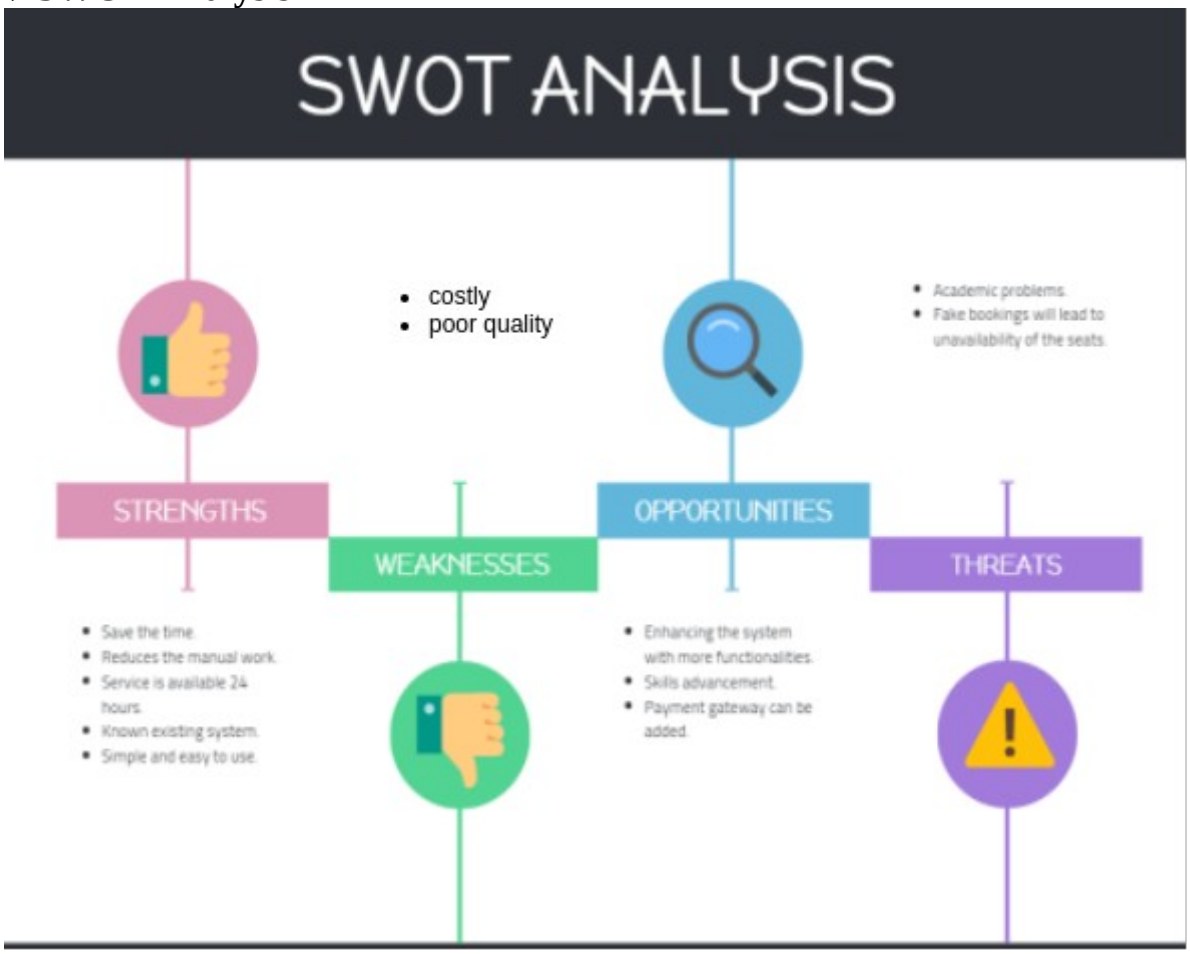
It will prevent the injuries and save the time of users. User can play this game on Linux and Windows operating systems. Size of game is so less. User can make change as they want in source code by cloning from the Github.

## Defining the system

- **System Functions:**
  - **System details:**
    - Play the snake game on Linux and windows operating systems.
  - **Running:**
    - Run the make run code on terminal Or CMD.
  - **Score Board:**
    - The details of creator, score, version of the game and speed of snake.
  - **Stop the game:**
    - Press Ctrl + C for stop the game.
  - **Extra features:**
    - Play game for unlimited times.
- **Block Diagram:**



## SWOT Analysis




## 5 W's:

- **Why:**

  To reduce the manual work, along with that to save the time and efforts of Users to personally visit the ground and play the game.

- **Who:**

  Any one can play it.

- **What:**

  A simple and easy to use system for the Users to do all the activities related to play the game in any OS.

- **When:**

As the India has the largest railway network in the whole of the world and to handle it manually is quite tough job thus it is required now to design such systems which will help to do this job efficiently.

- **Where:**

  This problem is expected all over the world.

## Detail Requirements:

- **High level requirements:**

| ID | Description | Status |
|---|---|---|
| HL01 | User Shall be able to Run. | Done |
| HL02 | User Shall be able to play the game. | Done |
| HL03 | User Shall be able to view Score and speed. | Done |
| HL04 | User Shall be able to change the source code. | Done |
| HL05 | User Shall be able to stop when they required. | Done |
| HL06 | Data should be saved in case of failure. | Future |
| HL07 | Proper GUI can be implemented for further. | Future |

- **Low level Requirements**

| ID | Description | Status |
|---|---|---|
| LL01 | The user will be able to run the make file successfully in the all os. | Done |
| LL02 | User Shall be able to play the game. | Done |
| LL03 | User Shall be able to view Score and speed. | Done |
| LL04 | User Shall be able to change the source code. | Done |
| LL05 | User Shall be able to stop when they required. | Done |
| LL06 | Data should be saved in case of failure. | Done |
| LL07 | Proper GUI can be implemented for further. | Done |
| LL08 | User can stop the game whenever they want | Done |
| LL09 | User can view the score and speed | Done |

| | | |
|---|---|---|
| LL10 | User can make changes in source code by cloning from Github | Done |