# Design Overview for Hungry Snake

Name: Gia Huy Nguyen

Student ID: 103441107 - SWH00039

# Summary of Program

The software is a recreation of the popular snake game called Hungry Snake. Users would initially be welcomed by a start screen that allows them to modify the music, key-bindings, and various game parameters to make the game more or less tough.

It will include a three-dimensional interface to encourage the user to engage with the game. This is a fairly simple rendition of classic snake... nothing very unique here. However, my objective was to add a multiplayer component to it as well as experiment with a Ruby framework.

Sketch of sample output: Diagram

Description automatically generated

# Required Data Types

Describe each of the records and enumerations you will create using the following table (one per record).

Table : <<record name>> details

|  |  |  |
| --- | --- | --- |
| Field Name | Type | Notes |
| Player | String | Name of the player |
| Difficulty | Integer | Difficulty of the game |
| Music Volume | Integer | Music volume level |
| Score | Integer | Score player chieved |
| High Score | Integer | Highest score player chieved |

Table : <<enumeration name>> details

|  |  |
| --- | --- |
| Value | Notes |
| Up | Movement direction |
| Down | Movement direction |
| Left | Movement direction |
| Right | Movement direction |

……

# Overview of Program Structuredd

# Functions (these functions will execute in response to the user's inputs)

* def initialize

The start of this function is to start the Gosu GUI

* def add\_segment

This method is used to initialize the snake's movement. This also stops the snake from backing up. This indicates that the player is unable to maneuver the snake within itself.

* def update

This method takes a user's input commands and converts them into a movement. When a player eats an apple, the function is also utilized to update their score. The function is also used for snake bounds, which means that if the snake reaches the game's boundaries, it will die and display a game over screen.

* def draw

During the game, this function displays the snake's location, the apple's position, and the player's current score. When a player loses a game, the feature displays the Game Over screen.. The function also draws the Game over screen when a player has lost the game.

* def self\_harm

This function is used to kill the snake when a player hits the snake's body with its own head.