# Design Overview for “Rolling Ball” project

Name: Gia Hy Nguyen

Student ID: 101922778

# Summary of Program

I plan to create a clone of a very famous game in the 80s, which is known as “Rolling Ball”.

The rule is that player has to keep the ball stays within the screen area, prevent it from touching the upper thorn and falling beyond the bottom of the screen. When playing, player can collect items such as extra live and shield to increase the chance of getting higher score. The game’s difficulty will be increased over time.

A sketch of sample output:

Graphical user interface

Description automatically generated

Figure 1. Sample output

# Required Data Types

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

Record 1: Player

|  |  |  |
| --- | --- | --- |
| Field Name | Type | Notes |
| :image | Gosu::Image | Image of player |
| :x | Integer | x-position of player |
| :y | Integer | y-position of player |
| :moving\_velocity | Integer | Velocity of moving horizontally |
| :falling\_velocity | Integer | Velocity of falling horizontally |
| :lives | Array | Remaining lives of player |
| :inventories | Array | Player’s collected items |
| :score | Integer | Player’s score |

Record 2: Block

|  |  |  |
| --- | --- | --- |
| Field Name | Type | Notes |
| :image | Gosu::Image | Image of the block |
| :x | Integer | x-position of the block |
| :y | Integer | y-position of the block |
| :moving\_velocity | Integer | Velocity of the block moving upwards |
| :is\_obstacle | Boolean | Check whether this block is an obstacle |

Record 3: Item

|  |  |  |
| --- | --- | --- |
| Field Name | Type | Notes |
| :image | Gosu::Image | Image of the item |
| :x | Integer | x-position of the item |
| :y | Integer | y-position of the item |
| :velocity | Integer | Velocity of the item moving upwards |
| :item\_type | Enumeration | Type of the item |

Enumeration 1: ZOrder

|  |  |
| --- | --- |
| Value | Notes |
| LOWEST | z = 0 |
| LOW | z = 1 |
| HIGH | z = 2 |
| HIGHEST | z = 3 |

Enumeration 2: ItemType

|  |  |
| --- | --- |
| Value | Notes |
| LIVE | type = 0 |
| SHIELD | type = 1 |

# Overview of Program Structure

Moving functions:

1. move\_left (player)

* Move player to the left and keep player within the screen area.

1. move\_right (player)

* Move player to the right and keep player within the screen area.

1. fall (player)

* Let the player falling downwards.

Game play functions:

1. spawn (player)

* Spawn the player randomly on a non-obstacle block.

1. reduce\_live (player)

* Reduce player’s number of live and spawn the player.

1. check\_boundary (player)

* Check if player has touched the top and bottom boundary of the playable screen.

1. stand\_on\_block (player)

* Keep the ball stands on a block when the ball lands on it.

1. generate\_blocks/items

* Generate blocks, obstacles and different types of items randomly.

1. collect\_items

* Let player collect the items when touching them.

1. increase\_difficulty

* Increase the difficulty over time.

Structure chart:

Diagram

Description automatically generated