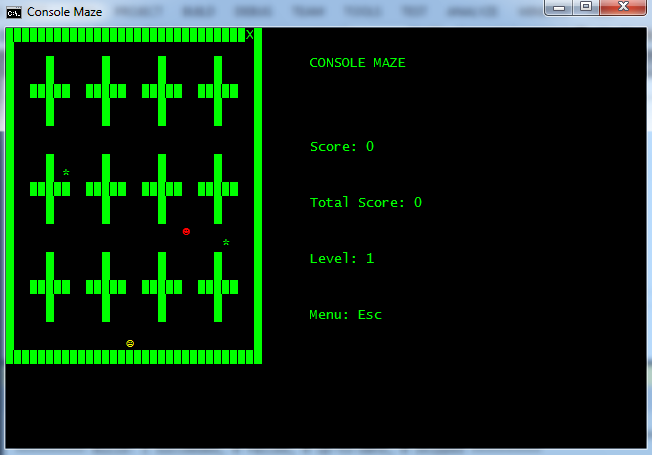
**Team “Stingray”**

1. Lubomir Ganchev (luboganchev)
2. Roman Peshterski (roman.peshterski)
3. Konstantin Severy (ksevery)
4. Diana Pesheva (dpesheva)
5. Georgi Yaramov (georgi.yaramov)
6. Hinka Hineva (hinkah)
7. Borislava Hinova (arthemida)

# Project Description

This is a simple maze game written in C# programming language. The idea is to navigate the player through the playground and reach the Exit. Use the arrow keys to make moves. In order to unlock the Exit the player should collect coins from the playground. The required amount is 60% or more of all coins in the current level. The challenge is that there is a bad guy looking to kill the hero.

Using Esc key the player could open the menu by any time.



The project satisfies completely the general requirements of the assignment.

It has the following elements:

1. **Classes**:

* ***ComsoleMaze:*** the entry point of the game, set game field.
* ***Player***: defines the name, character, collected coins, color, direction, row and column
* ***Bot*** : defines the name, character, collected coins, color, direction, row and column
* ***Movement***: mange the moves, check if the move is available, change the direction, do the move
* ***IEntity***:
* ***Item:*** defines color, character, row and cow of the item
* ***Level:*** holds all coins collected in the current level, the required percent to unlock the Exit, open the Exit if there are enough coins
* ***MainMenu***: draw a menu on the console and allow the user to choose one of the options in it.
* ***Map***: Initialize the playground, the player, the bot, the items and the Exit.
* ***Status***: manage the status bar shown during the level
* ***ConsoleIteraction:*** draw map, print/delete characters on the console

Required Classes: 0, Classes in app: 11

1. **Methods**:

* ***CheckForMenuShortcut***
* **SetGameField**
* ***PrintOnTheConsole***
* ***RemoveFromConsole***
* ***DrawMap***
* ***DrawPlayers***
* ***IsTheExitOpen***
* ***OpenTheExit***
* ***ManageMenu***
* ***PrintMenu***
* ***LaunchLevel***
* ***ReadDisplayHighscore***
* ***ReadDisplayHelp***
* ***ReturnToTheMenu***
* ***InitMap***
* ***InitCoordsForPlayerAndBot***
* ***SetCoins***
* ***ChangeDirection***
* ***Move***
* ***IsMoveAvailable***
* ***CollisionDetect***
* ***PlayMusicCoin***
* ***PlayMusicExit***
* ***ChangePlayerDirection***
* ***PrintStatusBar***
* ***ConsoleBuffer***

Required Methods: 10, Methods in app: 26

1. **Interfaces**:

* ***IEntity***: Inherit by the player and the bot

Required Interfaces: 0, Interfaces in app: **1**

1. **Exceptions**:

* ***FileNotFoundException- 3 exceptions***
* ***DirectoryNotFoundExeption – 1 exception***
* ***Exception- one global exceptions***

Required Exceptions: 2, Exception in app: 5

1. **Multi-dimensional arrays:**

* ***map – two dimensional array***

Required Multi-dimensional arrays: 1, Multi-dimensional arrays in app: 1

1. **One-dimensional arrays:**

* ***items – List<Item> holds all items (coins)***
* ***menuItems – holds all option in the menu***
* ***levels – holds the maximum possible points for all levels***

Required One-dimensional arrays: 3, One-dimensional arrays in app: 3

1. **Existing .NET Classes**

* ***System.ConsoleIteracion***
* ***System.ConsoleColor***
* ***System.Random***
* ***System.Threading.Thread***
* ***System.IO.StreamReader***
* ***System.Collections.Generic.List<T>***

Required Existing .NET classes: 3, Existing .NET classes in app: 6

1. **External file use:**

* 3 files with different maps. One per level.
* 1 file containing the highscore
* 1 file containing the help

Required External file: 1, External files in app: 5

1. **Sound effects**

* Sound when the player get a coin
* Sound when the player reach the Exit

<https://stingray.codeplex.com/>