*C#2 Team Work – Console Game*

*Project documentation*

Team: “Ed, Edd ‘n’ Eddy”

# Team members list

1. Никола Тодоров (nmtodorov)
2. Владимир Василев (Vlado\_XXX)
3. Васил Богданов (Eagle)
4. Цонко Генов (tsonko\_genov)

Project

Console game: Block Labyrinth

Block Labyrinth is a C# script game created to run on console application. It shows how with C# language and object-oriented programming can be created a game with colors, text, movement patterns, enemy logic (additional units for game difficulty) and sound.

# *Introducing the game*

Block Labyrinth simulates a labyrinth game, where the player moves through game space to reach the end of the labyrinth and exit the level. The game space can be just simple for moving inside the labyrinth, can contain walls (obstacles) and keys of different colors - keys have to be reached to break the walls, or including enemies which can drop the player live points to zero and end the game.

The main character used for playing the game – **Hero**, is presented by the char ‘\u25a0’, and the same char is used for the block tile representation (boundary wall), **Keys** for breaking path limiting walls inside the game space, path limiting **Walls**, the finish level symbol – **End**, and additional level decoration, such as Level 5 green elements of Telerik logo. This block tile representation is used for visual integrity and design interface, which represents to the player the main topic of the game – “Labyrinth of Blocks”. The **Hero** element is in black color, **Keys** are in yellow and green color, **Walls** following in dark yellow and dark green colors corresponding to matching **Keys** color, **Zombie** (enemy) is in red color, and **End** symbol is in dark magenta color.

# *Levels*

## *New*

Intro level, which introduces to the player the basic gameplay – moving the **Hero** to the **End** symbol to finish the labyrinth.

## *GoGoGo*

The level presents the standard gameplay, to move forward player need to hit colored **Keys** to break **Walls** of the same color, so he can clear the path in front of him and reach the level **End.**

## *Labir*

The level adds game difficulty, where player finds himself in a labyrinth map, where he need to reach colored **Keys** to free the path from **Walls**, and by choosing not the right pattern of hitted **Keys** he will be stuck in the labyrinth and can’t reach the level **End.**

## *BehindEnemyLines*

The level encounters the player with the first enemies in the gameplay. Player starts with 4 Lives and 2 enemies, and if an enemy hits the player he loses 1 life point, and level restarts. No obstacles in the level included so the player can familiarize with the game logic and clearing the level.

## *BehindEnemies2*

The last level before end of game, to the previous game logic here is a labyrinth added, and enemies switch moving path directions. The places which contains colored **Keys** for breaking the **Walls** are constructed so, that the player can hide the **Hero** when chased by the enemies. For additional gameplay a Telerik logo type is added in the map.

# *Code structure*

## BlockLabyrinth.cs

Main class, calls the GameMenu, loading levels and game finish. Contains the window size of console and sound for the game.

## Engine.cs

Class contains code logic for initializing and reloading Map, Hero - his lives counter (score field) and movement, and Zombies (enemies).

## Map.cs

Class for creation the playfield and inner game field, with char, map high and width, and color values.

## GameMenu.cs

Class load on the console display a game menu text, where with cursor (keyboard key arrows)

player can choose a set of options:

* Start game (start game levels)
* Choose Level (loads another menu where player can choose levels 1-5 by pressing keyboard key numbers 1-5 or go back to the Main menu with typing the command “back”)
* Create Level (player can create own level)
* Exit Game

## MapMaker.cs

A class that lets the user/player create his own game level (playfield, enemies, colors). The following command accessible from keyboard are given:

*By pressing:*   
**E** – create empty block

**W** – wall

**Y** – yellow key

**L** – yellow wall

**G** – green key

**N** – green wall

**H** – hero start point

**X** – exit symbol

**ESC** – refresh

**F10** – save created level

## Hero.cs

Class for implementation of the **Hero**. Initializes player’s **Hero** properties for movement coordinates (X and Y) and symbol representation char ‘\u….’(mentioned above).

## Zombie.cs

Class for implementation of the enemy, initializes properties for movement coordinates (X and Y) and symbol representation char ‘\u….’ (mentioned above).

# *TFS Repository*

*URL*: <https://samoletche.visualstudio.com/>

*Username*: vladimir.vasilev@yahoo.com

*Password*:

***Additional information:***  
*For more information about the script and game visualization please check source code folder and PowerPoint presentation demo in the added archive.*

**Note:**  *This documentation reflects the current state of the Block Labyrinth game sample at the time of writing.*