**“Crazy Rusher Game” Brief Documentation**

**Team name: "Spiderman"**

**Team members:**

* Viktoriya Stoyanova (viktoriya\_st)
* Desislava Ivanova (desislavaiv)
* Zdravka Sandova (Zdravka\_Sandova)
* Nikolay Nikolov (nikolay.n)
* Teodora Tikova (teodoratikova)

**TFS Repository:**

* [***https://arreatface.visualstudio.com/***](https://arreatface.visualstudio.com/)

**Project explanation:**

**Game plot**

**“Crazy Rusher Game”** consists of a matrix, dungeon maze, known as “The maze”, which is being unhabited by swarm of evil creatures, in short known as “The evil creatures”.

It is generally accepted, that peace and happiness always reign over the forces of evil, so our main character “Smiley” is being teleported to the dungeon, to destroy the swarm, before they turn the maze into an infectious hive, and populate the world with bloodthirsty dummies.

It stands to reason that the goal of the game is to destroy the evil creatures, faster and make higher score than other players.

**Game code**

Game code is composed using several classes and main ***CrazyRusher.cs*** file, which runs the game engine.

***GameField.cs:***

* Initializes the main matrix, which represents the maze;
* *PlayerCoordinates()* sets the player’s coordinates onto the matrix;
* *SetMatrixContent()* fills the matrix with its inner and outer walls;
* *PrintMatrix()* prints the matrix and its content on the console;
* *SetConsoleDimensions()* defines the main matrix’s dimensions;
* *DrawBorderLines()* is used also in the welcome screen of the game, and in the gameplay. It separates the game field with the status field;

***GameID.cs:***

* *PrintInitialPicture()* reads external files and prints their content on the console. The external files contain the game’s welcome logo;
* *GetPlayerName()* gets the player’s nickname;
* *PrintGameName()* prints the game logo while gameplay process;

***Player.cs:***

* *SetPlayerCoordinates() sets the player char’s coordinates in the matrix and places it init;*
* *MovePlayer() handles the pressed buttons, which can be up arrow, down arrow, left and right arrow for movement, and space bar for shooting;*

***GameScores.cs***

* *CollisionScores()* method calls all other methods considering different game situations:
* if the player shoots an enemy, then he takes 10 scores;
* if the player is caught by an enemy, then he loses a life;
* if the lives are 0:
* GameOver() method is called:
  + clears the console;
  + sets the background color red;
  + writes “Game Over”;
  + CalculateTimeElapsed method is called:
* takes the start and the finish time in seconds and calculates their difference;
* WriteHiScoresHistory() method is called:
* Calculates scores by the following formula: *scores \* (1000 - elapsed time in seconds);*
* Writes the scores in text file;
* Displays all time scores from the file;