**Escape The Cube**

1. **Team members**

|  |  |
| --- | --- |
| **Name** | **User Name** |
| Nencho Nenchov | nencho83 |
| Hristo Nenov | hristo.nenov |
| Nikolay Pavlov | PaperNick |
| Pavel Aslanov | Pavel.Aslanov |
| Reni Getskova | geckova |
| Miglena Hristova | MiglenaHH |
| Mariana Naidenova | marianamn |
| Ivaylo Hristov | ivaylo.hristov |

1. **Brief description of the game**

The game represents labyrinth in 3D Cube. The Player is placed in the middle of the Cube and his task is to find a way out. In the beginning of the game he chooses category of questions:

* IT quiz
* Movie quiz
* Science quiz

Questions in each category are separated in three severities. The Player has 30 seconds to answer the question. If he answers before the time has elapsed the remaining time is added to the time of his next question.

Player moves through the playfield and gives a direction by pressing the arrow keys. When he enters a room he has to answer a question from the chosen category in order to move in the next room. When he gives wrong answer his credits decreased he stays in the same room and new question is given. If the Player runs out of credits the game is over his score is recorded if it is among the Top 10 scores.

If the Player gives the correct answer he chooses his next direction. The closer he gets to the wall of the room, which is the exit of The Cube, the harder the question gets. When he successfully answers the last question he sees his current score and the rank list.

1. **Technical implementation of the game**
2. Main Menu

The main menu takes as an argument \*\*1 one-dimensional\*\* array that lists all the options. The boolean variable isShown controls the menu visibility. The player can navigate and choose options from the Menu. The starting point of the game.

1. Game

\* Game() - takes the size of the cube as a parameter and generates a labyrinth

\* Start() - creates a new game and new player positioned at the center of the cube.

\* CreatePlayer() - takes the center of the cube as perimeter

\* GameOn() - controls the main functionality of the game

\* Play() - gets the player input

\* ProcessDirection() - handles the directions of the player

\* DisplayCurrentPlayerInfo() - displays information about the player state in the labyrinth

\* IsDirectionPassable() - checks direction

Control the game state:

\* Reset()

\* Resume()

\* Exit()

1. Cube generation

The Cube is built using **1 three-dimensional** array with size 5x5x5. A random generator creates a path with exit, and when the game starts the player is positioned in the middle of the cube.

1. Player

The Player class handles information about: **Player, Coordinates, Category, Score**

This class has **9** **methods:**

* Player() – initializes a player
* Nickname()
* Credits() – the remaining moves of the player to exit The Cube
* BonusScore() – formed from the remaining time which is added to the next question time for answering
* Category() – in the beginning of the game the player chooses a category
* Position() – keeps the current position of the player
* ChooseDirection() – after giving an answer the player chooses his next direction
* IsVisited() – checks if the chosen room has been visited
* MarkedIsVisited () – marked the room asvisited

1. Challenge – this class has **2** **methods**

* TakeParts() – read **text file** from CSV
* DisplayChallenge() – display questions and answers

1. Scores - for this class is used **2** **methods:**

* SortScores() – sorts the Top 10 scores in descending order
* CalculateScore() – calculates the score. Result is formed by the credits multiplied by the remaining time.

1. Utility Class – for this class is used **3** **methods:**

* GenerateLabyrinth()
* WriteToCSV()
* ReadFromCSV()

1. **The URL of Git repository**

<https://github.com/nencho83/CyclopsKing>