# Team Stryker Application: Ice Jumper

[**https://stryker.codeplex.com/**](https://stryker.codeplex.com/)

Team Members

|  |  |
| --- | --- |
| Pavel\_Dobranov  Павел Добранов | NikolayVasilev  Nikolay Vasilev |
| stambeto09  Станимир Якимов | jintrax111  Константин Илиев |
| Svetoslav1987  Svetoslav Hlebarov | lilia.kostova.7  Lilia Kostova |
| dre4mless  Александър Сеферинки | Kiril.b.iliev  Кирил Илиев |
| shevitza  Евгения Христова | mental  Иван Конев |

Story and rules of the game

The player must cross a basin covered with ice by jumping from block to block. For simplicity, assume that the player is light enough and can’t reverse the block of ice at jumping. If the player jumps between ice blocks, right on the water - the game ends. When standing on a block of ice, the player is "bear" in the water with ice. If it come out of the playing field - the game is over. The ice blocks moving several adjacent flow in opposite directions. The player has to move to be able to stay in the playing field and to pass on the other side without falling into the water.

The game has ten levels of difficulty, which differ in the width of the water pool. The player has three lives.

Idea for realization

The player moves on the game field and enters in his different areas - sides, ice and water.The water and the sides are static and there are arrays of ice blocks moving above. When the player is on the upper side - he hasn't begun to move - when he's on an ice block the game continues, when he fall into the water , he's drown, when he is on the lower side he's completed the level.



Static variables, enumerations and structures

**Enumerations enum:**

-       Elements of the river – water, ice, grass

-       Elements of the directions – up, down, left, right

**Structure**

-      Player's position horizontally and vertically. Player's name,scores and bool value if is loaded.

|  |  |
| --- | --- |
| **PROPERTY /STATIC VARIABLE/** | **DESCRIPTION** |
| static string GameName = "Ice Jumper"; | Name of the game |
| static int gameWindowWidth = 90; | Width of the window |
| static int gameWindowHeight = 40; | Height of the window |
| static int riverWidth = gameWindowWidth - 20; | Width of the game field (the river) |
| static int riverHeight = 7; | Height of the game field |
| static int defaultReverPositionX = (gameWindowWidth - riverWidth) / 2; | Game field’s position in the window horizontally |
| static int defaultReverPositionY = 5; | Game field’s position in the window vertically. |
| static int[][,] iceCubes; | IceCubes array |
| static Player player; | Heroe’s variable of type structure , which have his position in the game field. |
| static int currentLevel = 1; | Current level |
| static int lastLevel = 10; | The last  level is ten. |
| bool isGameComplete | End game |
| bool isLevelComplete | End level |
| bool hasDrown | If the player was drown. |
| static string ScoreFilePath = @"HighScores.txt"; | The path file for results. |
| static Random generateRandomNumber = newRandom(); | Random number generator |
| static bool isLevelComplete = true; | Bool variable for complied level |

Static Methods

| **METHOD** | **DESCRIPTION** |
| --- | --- |
| static void InitializeIceCubes(int rows) | Ice block length and distance between the next ice block are generated with random numbers |
| static void PrintGameInfo(int levelTime) | Game Info |
| static void PrintRiver(int rows) | With Console.SetCursorPosition() we draw the area of water and grass. |
| static void PrintPlayer(Player player, bool isMoving) | Draws the player on grass , ice or water background if he is moving , if he has moved ,it deletes him from his old position. It uses SetCursorPosition. |
| static void StartLevel(int currentLevel) | Deletes the screen and shows the next level number, counts until three, to  catch the atention of the user and makes a sound signal to begin. |
| static bool PlayLevel() | Makes a new structure - player and positions him in the middle of one of the sides.     - Calls Game Info     - Calls the Ice Cubes initialization     - Calls Print River     - Calls Ice Cubes /moving ice/     - Calls the logo drawing     - Counts the time     - In infinite bucle tracks if enter or cursors are pushed.     - If enter is pushed , returns isLevelComplete = true for new level     - If the arrows are pushed - sets coordinates of the player in the coresponding position calling the MovePlayer method.   - Moves the ice blocks of the river each 100 ms. |
| static void UpdateIceBlockPosition() | Recalculate the ice blocks positions -third row- in the right, fifth row - in the left... |
| static void MovePlayer(Directions direction) | Recalculate and calls player's drawing, depending on the direction and if he is in the playing field. |
| static void MoveIceCubesOnConsole(int row, Directions direction) | Moving ice blocks. |
| static void CheckForDrown(out bool isOnCubeStartIndex, out bool isOnCubeEndIndex) | Checks whether the player is in the play field and returns true or false. |
| static bool CheckForCompleteLevel() | Checks whether the player has reached the other side. |
| static void GameOver() | Prints game over message and makes a sound signal. |
| static void HighScorePrint(string[] highScores, int index) | Print results. |
| staticvoid LevelFailed() | Prints level failed message for the current level, when the player have failed and expects the user to enter any key to continue. |
| static void LevelComplete() | Print level completed message for the current level, when the player has managed to pass the level and waits the user to push any button continue. |
| static void PrintPrompt(string[] prompts) | Print results. |
| static string GetPlayerName() | The Name will be written in file. |
| static List<string> LoadSaveFile() | Load and save the file for results. |
| static int IndexOfUserInSaveFile(string userName) | Player's names list. |
| static void SaveGame() | SAVE GAME (format: name|lives|score|level)  Try-catch block. |
| static void LoadGame() | Player's lives scores and level |
| static void DeleteSaveGame() | Removes data from file for scores and names. |