**Lost Forest**

Description:

The game consists of the starting point, 3 stages, each with its own level of difficulty:

**Forest:** there is one enemy to defeat. By exploring the board, the player needs to find it and defeat it to earn the key to access the next stage. If not, the game is over.

**House:** there is an enemy to defeat. On the board there can also be found 2 chests with something inside.

**Cave:** there are enemies to defeat and one riddle to be guessed in order to win the whole game. If the player succeeds, he wins the game.

**The Starting Point:** It is the place where the player begins his story and returns after every won stage. It enables the player to access different stages of the game. It can be done only if the player possesses a key to the desired stage. However, once the stage is explored by the player, re-accessing it is not possible.

**The Stages:** There are 3 stages mentioned above. Each stage is arranged as a board (matrix 5x5). On each field is either an enemy, a chest or nothing (just to explore). The player can see only his position on the board, enemies and chests are placed randomly on the board.

**The Exploration:** The player decides between the direction where he wants to move: UP, DOWN, RIGHT, LEFT introducing keys (w,s,d,a) on the keyboard. The place can be either explored, when a chest is found it can be opened, and when an enemy is encountered it needs to be defeated to be able to continue the game.

**The Fight:** At each stage the player has to fight enemies, after winning the fight, he gets a key for the next level. The level of the fight depends on the difficulty of the stage the player is currently in. The player and every enemy has health and damage (randomly generated for the enemy), up to who is stronger, we have a winner.

**The Chest:** When the player encounters a chest on his way, it is opened. Inside there can be more health to defeat the enemies or the chest can be empty.

**End Of The Game:** The game ends if the player loses the fight at a current stage or wins the whole game. At the end of the game, there will be a picture shown with the proper information.

Modules Of The Code:

**game\_board:** the structure of each stage of the game, the rules and the riddle, functions responsible for moving around the board

**board\_elements**: it consists of the functions corresponding to the elements of the game: fight if enemy is encountered, opening chest if chest is found

**main\_game:**  it has imports of the rest of modules and calls its functions to initialize stages of the game, shows messages about current status of the game. It is a skeleton of the whole game

**files:** functions for reading and opening a file, opening and showing an image

**user\_Input:** functions which take an input from a player

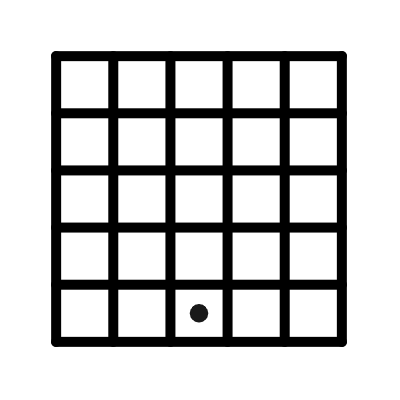
External Modules:

**Random:** to place an enemy, chest on the random position on the board, to select a riddle to solve, to add health to the player, to create damage during the fight

**Time**: to control the time between consecutive messages shown to the player

**PIL - Image:** to show the image to the player with the information about the end status of the game

Code Structure:

Each functionality is coded in a different file, i.e. Game Logic will contain functions for the fight, chest etc. Those files will be imported to the main module where the whole code of the game will be executed.

List – creating list of enemies, list of the fields on the board visited by the player, list of riddles

Matrix – each stage is a board 5x5. Player changes his positions, introducing keys: w, a, s, d.

Link To The Video:

<https://drive.google.com/file/d/13blxngO693wPKfO0GCPzJ9KLmqwGRpUV/view?usp=sharing>

Work Summary:

**Together**

* finding the Idea for the game and setting the rules
* developing the structure for the code
* putting together a pdf for the project

**Maximilian Macheiner:**

* Fight function: Developing the fight function with the use of the random package, implementing it to the main code and add the related guide through the game,
* Image file: Creating the winning- and losing image and implementing it in the code.

**Carlos Sampedro Menéndez:**

* Start: developing the function necessary to read the introduction, as well as writing the player’s name,
* File with introduction: creating the content of the introduction file,
* Drop function: creating a function that randomly chooses what is inside the chest.

**Agnieszka Gwizdek:**

* Programming the board (the whole game\_board module): functions connected with moving around the board (checking whether the move is valid etc.), arranging elements on the board,
* Creating stages of the game, starting point function (game\_board module),
* Input module: creating functions which take an input from the player,
* Putting the pieces of the code together in the main\_game module, arranging other modules,
* Riddle function: creating a riddle for the last stage of the game,
* Making a video.