Documentation of the game

“Oops, There Goes The Retake”

Welcome to the official documentation of the game "Oops, There Goes The Retake"! This resource is designed to help you get comfortable with the game world, understand its mechanics, and start enjoying the game to its fullest. "Oops, There Goes The Retake" - is a small maze and puzzle game. In it you have to find a way out of the maze before time runs out and before you get caught. The analogue of our game was taken from the game “Pac–Man”. Pac-Man is a famous arcade video game developed by Namco and released in 1980. The main character of the game is a yellow round creature known as Pac-Man, who travels through a maze, eating dots and dodging ghosts. But we made our own version of this game.

In this documentation you will find information about controls, gameplay, characters, game worlds, game modes and other useful aspects. We hope that this information will help you immerse yourself in the fascinating world of "Oops, There Goes The Retake" and enjoy the gameplay.

In our game there is a main character who studies at the university, and he is stuck alone in his own university. And the main character must leave the building at a certain time. But he will have an opponent, his teacher. To play our game you just need to press these keys: 

And when you are in the maze, your opponent will chase you, if he catches you, you lose and do not go to the next level. And you too will lose when time runs out. There are 3 levels in this game and each one will have different difficulties. The goal of this game is that the student must exit the maze without being caught and must not waste time.

The analogue of our game was taken from the game “Pac–Man”. Pac-Man is a famous arcade video game developed by Namco and released in 1980. The main character of the game is a yellow round creature known as Pac-Man, who travels through a maze, eating dots and dodging ghosts. But we made our own version of this game. We also took the location of our university, because our university is like a labyrinth. We have created a map of our university. The building will have three exits which our main character can exit.



Now our characters:



1. Student 2. Teacher

In our game our characters will be in a vertical position because this game is based on mazes:

 <our characters in vertical position>

Our game will only have one main character, so it is based on a single player game. But opponents may increase in levels.

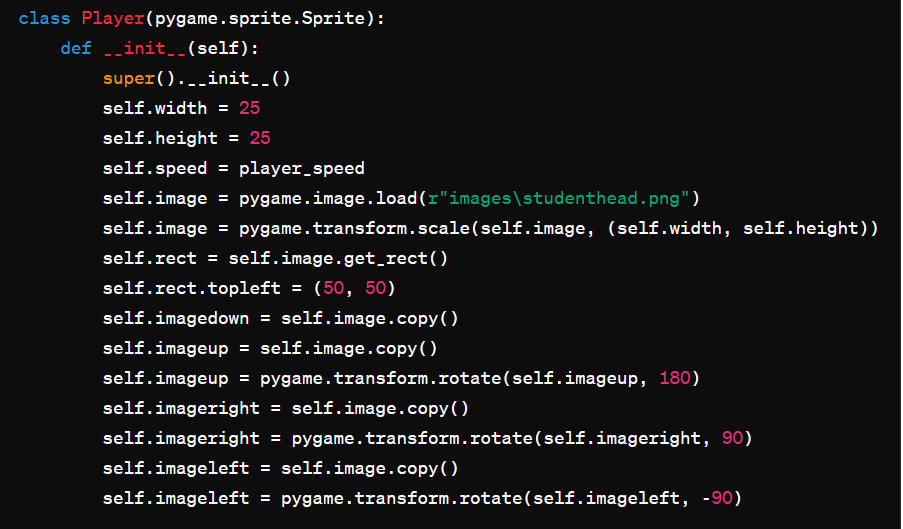
And also in the next levels it will be more difficult. In the second round there will be a “ONAY” card that the student must catch before his time runs out and he is caught.

 <ONAY card>

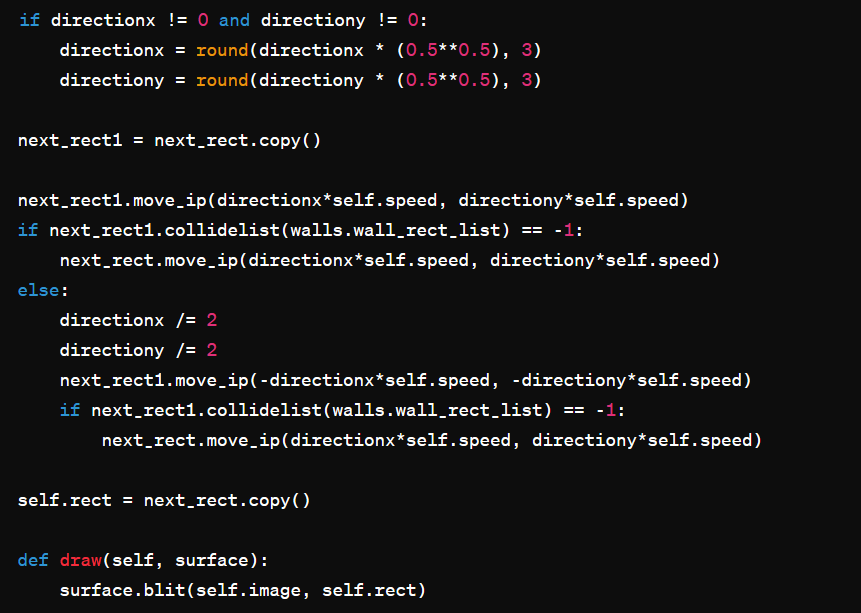
Also, in the third round there will be the same thing, but there is one more teacher, this means that there will be not one, but two opponents.

<second opponent teacher(in the 3rd round)>

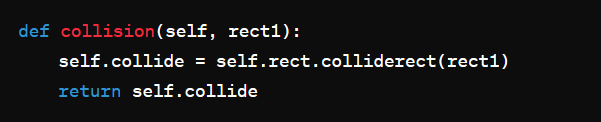
Now let’s explain a little how the code of this game works.



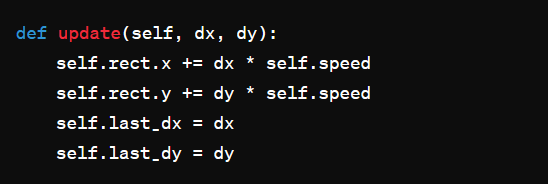
This piece of code defines the Player class in the game, using the pygame library to create a player sprite. This class is designed to display and control the player character.



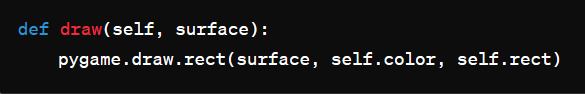
This code is part of the update method of a class representing the movement of an object (probably the player) and handling collisions with obstacles (walls.wall\_rect\_list). The draw method implements displaying an object on the screen.



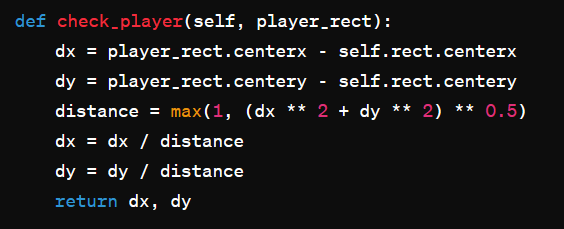
This code snippet defines a collision method within the class, presumably to check for collision between two rectangles: the rectangle associated with the current object (self) and another rectangle, ‘rect1’, which is passed as an argument to the method.



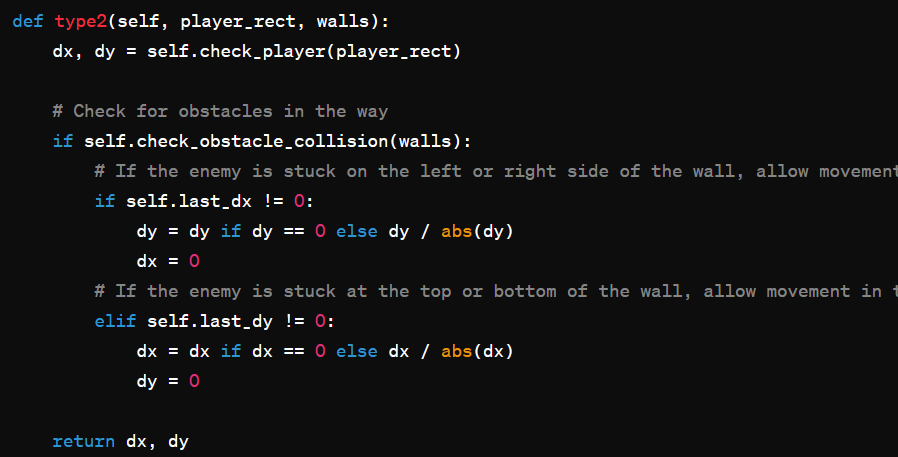
This method is used to update the position of an object based on the specified change values of the ‘dx’ and ‘dy’ coordinates.



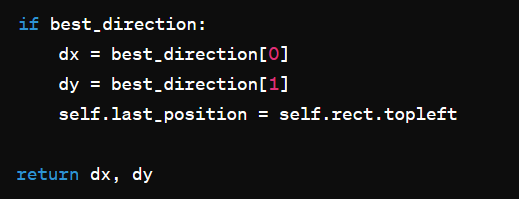
This method is used to draw an object on the specified surface using the ‘pygame.draw.rect’ function.



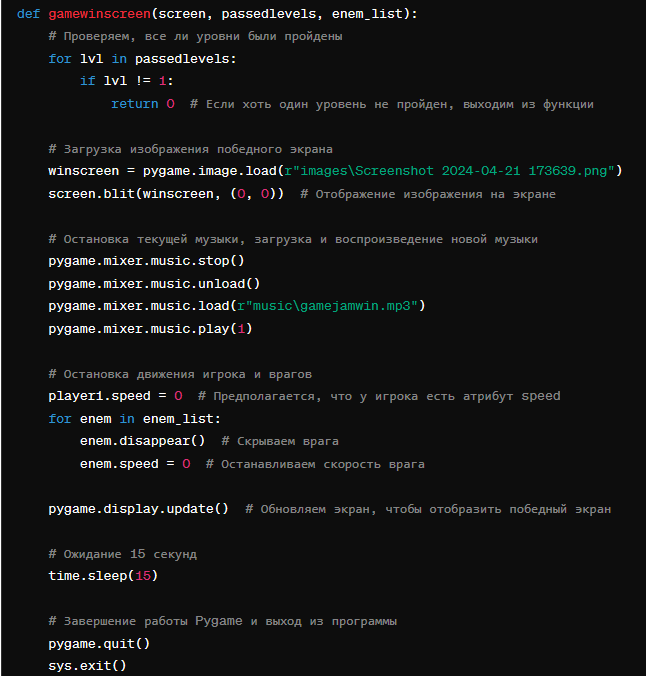
This method is used to calculate the direction (unit vector) from the current object to the ‘player\_rect’, presumably representing the player.



This piece of code is a type2 method inside a class, presumably for an object in the game, that controls the behavior of an enemy (or other object) relative to the player (player\_rect) and obstacles (walls). Let's look at each part of the method.



This piece of code looks like part of a method or function that handles the direction selection (‘dx’ and ‘dy’) based on some conditions or logic. Let's look at what's happening in this code.



An example of using the ‘gamewinscreen’ function is provided at the end of the code, where we initialize ‘Pygame’, create a screen, set the levels completed and the list of enemies, and then call the ‘gamewinscreen’ function to display the victory screen.

The rest of the code explanation is written in the code itself..

Our handcrafted game "Oops, There Goes The Retake" is an entertaining and classic arcade game that can be expanded and improved with further development. This documentation provides a basic set of information to get started with the project and improve it further. By reviewing this documentation, the developers have gained an understanding of how to implement the main components of the game, including characters, maze, levels, points and interface.

Our team and I created a simple game for endterm. But with the help of this game we could work together and understand to support each other, and also it was a little experience for us in the future. we were also able to work as a team and create a game in a short time. Thanks to the teachers of KBTU for this format of education ! )))

Team members : Abdikadyr Amire, Talgat Tastemir, Sultanbek Zhanibek, Zhantuar Merey

Authors : 14’s team (11:00 – 13:00)

Teacher : Arnur Kelgenbayev