FINAL REPORT – Pac-Man

**1. Objectives**

The implementation of the Pac-Man game in our project aims to raise awareness about digitalization and green sustainability by embedding these themes into the gameplay. This makes learning about environmental issues in Europe more interactive and enjoyable, encouraging players to engage deeply with the content. By navigating the maze and solving related questions, players develop critical thinking and problem-solving skills applicable to real-world sustainability and digitalization scenarios. The engaging format enhances retention, making the educational content more memorable. Ultimately, the game seeks to inspire players to take action, form well-informed opinions, and become advocates for sustainability, while motivating them to further explore these important topics.

**What is Pac-Man?**

Pac-Man is a well-known maze action game created by Namco and first released in arcades in 1980. Initially named Puck Man in Japan, it was released in North America by Midway Manufacturing as Pac-Man. Designed by Toru Iwatani, the game quickly became one of the most iconic and influential video games in history.

In Pac-Man, the player controls a yellow, circular character named Pac-Man. The main objective is to navigate through a maze, consuming all the small dots (pellets) while avoiding four ghosts: Blinky, Pinky, Inky, and Clyde. The maze also features larger flashing dots called Power Pellets, which temporarily turn the ghosts blue and make them vulnerable, allowing Pac-Man to eat them for bonus points. The game progresses through multiple levels, with each level becoming more challenging as the ghosts' speed increases.

Pac-Man's straightforward yet addictive gameplay, along with its iconic characters and sounds, has ensured its lasting popularity. It has inspired countless sequels, spin-offs, and merchandise, remaining a significant part of video game culture.

**Pac-Man on the market**

Pac-Man's influence in the gaming market has been profound since its 1980 debut. Its initial success led to numerous sequels and adaptations, including beloved titles like Ms. Pac-Man and Pac-Land. Modern adaptations, such as Pac-Man Championship Edition and Pac-Man 256, have kept the franchise relevant for new audiences. Beyond video games, Pac-Man has inspired merchandise, animated series, and remains a symbol of arcade gaming's golden age. Its enduring popularity and adaptability showcase its lasting impact on the gaming industry.

**What is the connection between Pacman and education?**

Pac-Man's gameplay mechanics promote various cognitive skills that are valuable for education. By navigating through mazes, players develop problem-solving abilities as they strategize to avoid ghosts and collect pellets. The game also enhances spatial awareness and planning skills, as players must navigate the maze efficiently while considering the movements of the ghosts. Additionally, Pac-Man exercises memory and pattern recognition, as players memorize maze layouts and ghost behaviors to optimize their gameplay. The engaging nature of Pac-Man encourages players to persist through challenges and learn from their mistakes, fostering resilience and a growth mindset. Incorporating Pac-Man into educational settings provides a dynamic and enjoyable platform for enhancing these cognitive skills in students.

**About our game**

Introducing our latest Pac-Man game, themed around the critical issue of water and earth pollution. The game is structured on two levels, in the first level the small dots have been transformed into shimmering water droplets, while the power-ups are represented by water bottles, and on the second the power-ups have been transformed into leaves.

  The protagonist, Blue Pac-Man, is on a mission to combat water pollution, facing off against ghosts(Blinky, Pinky, Inky, and Clyde) symbolizing various pollutants.

In terms of graphics, we built our Pac-Man character as a blue, circular character that can be controlled using directional keys(w,s,a,d). Pac-Man can move up, down, left, or right within the maze.

The maze also contains four larger, flashing water bottles/leaves, named Power Ups in our game, which, when eaten,a pop-up question will appear on the screen and if answered correctly, it will temporarily turn the ghosts vulnerable, allowing Pac-Man to eat them for extra points. After being eaten, ghosts return to their home base (a box in the center of the maze) to regenerate. Each ghost has unique movement patterns and strategies, requiring players to develop tactics to avoid or confront them. All of the ghosts have the same pattern of chasing the player: directly chasing Pac-Man using the shortest path. They all share the same purpose, and that is to keep the Pac-Man from winning(eating all the dots). The player only has three lives.

If the player wins the first level, he can advance to the second, which is based on the same principle and has the same goal, only that this time we have the pollution of the earth in the foreground.

The scoring system includes points for eating small dots, Power-Ups, and ghosts. Eating small dots gives 10 points each,  eating Power-Ups gives 50 points each, and eating ghosts after consuming a Power-Up provides increasing points: 200 for the first ghost, 400 for the second, 800 for the third, and 1600 for the fourth.

**2. Documentation**

Our approach includes the creation of the graph in C and the implementation of the link with the Python interface.

The first step was defining the structures and creating the graph that would contain the essential elements of the game: dots and power-ups. On the Python side, we start with the import boards and then we continue with the game interface, the questions, and the making of the game.

  The input is used to control the Pacman and works at the same time as the rest of the code, the global declaration being used to be able to also send the command to the Python code. To create this connection between the C and the Python we used socket initialization. A socket is a software object that acts as an endpoint for communication. In network programming, a socket represents one end of a communication link between two programs running on the network. The’ socket’ function creates a new socket. The parameters define the address family, socket type, and protocol. Our code initializes the Winsock library, creates a socket, and prepares it for network communication using the TCP protocol.

The player must eat all the points without being caught by the ghosts, and with the help of the power-ups he can give himself an upgrade for a few moments and he can eat the ghosts increasing the number of points. Each ghost possesses its own thinking. Clyde is going to turn whenever advantageous for pursuit, Blinky is going to turn whenever colliding with walls, otherwise continue straight, Inky turns up or down at any point to pursue, but left and right only on collision, Pinky is going to turn left or right whenever advantageous, but only up or down on collision.

We use the ‘check\_collisions’ function to see what happens to the ghosts: If the Pacman is touched by the ghosts without having an active power-up or after deactivating it, the player loses a life. If he has it the number of points increases.

‘ghost\_collisions’ is the function that returns which part the ghost can take and it helps us to troubleshoot issues with the ghosts. The ghost can go down through the gate only if it is dead and in the other case it can go up through the gate.

We also implemented a series of functions for drawing the game: ‘draw\_board’, ‘draw\_player’, ‘draw\_ghosts’, and ‘draw\_misc’ which displays the score and the number of lives.

**3. Justifying our solution**

Implementing our Pac-Man game using C for the game logic and Python for the interface offers several advantages. C provides high performance and efficiency, essential for real-time game logic, along with low-level access to memory and system resources, and the ability to compile code across various operating systems. Additionally, C has mature libraries like SDL for handling graphics and input.

  On the other hand, Python's simpler syntax and readability facilitate faster development and maintenance of the user interface, supported by its rapid prototyping capabilities and extensive libraries such as Tkinter, PyQt, and Pygame. Python also integrates seamlessly with C, enabling us to write performance-critical components in C while managing the interface in Python. This hybrid approach combines optimized performance with quick development cycles, leveraging the strengths of both languages for an efficient and flexible development process.

**4. Project implementation**

Our project was developed using a combination of C and Python. C is a high-performance, low-level programming language, while Python is a high-level language known for its simplicity and readability. C provides multiple benefits: it is highly efficient, giving developers precise control over system resources and memory management, which is crucial for performance-critical applications. Python, on the other hand, offers a simpler syntax and automatic memory management (allocating and deallocating memory is done automatically). It also comes with a vast standard library and many open-source resources, which ensure efficiency for developers.

**5. Conclusion**

            In conclusion, our game aims to raise awareness of water and earth pollution through immersive gameplay and pollution-related questions, thus achieving our goal set for this project.