**20875 - SOFTWARE ENGINEERING PROJECT – FLAVIO UBALDO CAROLI - 3159436**

**Render Engine and 2D RPG Game ‘Dream Frame’** (the name does not refers to something actually doable in the game yet’

**Project Description:** I developed a render engine and a 2D RPG game from scratch, all implemented using C++. The render engine is designed for efficient 2D graphics operations, supporting an RPG with some essential interactions.

Everything is built with an extensive use of SDL2 library, that is typically essential for engines/game development in C++, because of utilities functions for input handling, texture creation and rendering.

For running details refer to README.md in the github repository of the project: https://github.com/flaviocaroli/RenderEngine2D/tree/newAnimations

**Project Components:**

1. **Render Engine:** that gathers A resource meanger (to load textures or other assets) map, collision rendering system and other structural details.
2. **2D RPG Game:** This is minimal, it comprehends a player character that can move through the map (also the camera follows him to keep it always at the center, and he stops moving if reaches the border of the map), run and ‘jump’ and it is animated for all directions through the pressing of ‘WASD’ on keyboard for normal walking and ‘SPACE’, ‘j’ for running and jumping respectively while moving. I used 2 websites for copyright-free sprites to use them as assets: <https://seliel-the-shaper.itch.io/character-base> and, <https://craftpix.net/freebies/free-top-down-orc-game-character-pixel-art/>. I added to the map some Trees and NPCs that can interact with th player with some simple dialogues if pressing ‘e’ on the keyboard when they are near you.

**Challenges Encountered:**

* **Collision Detection Enhancements**: Fine-tuning the CollisionManager was very annoying, since I had to consider two different cases: when I had to define the coordinates and dimensions of rectangles of trees that are bigger than the player (256 vs 128 pixels) to give impression that the two entities touched before the character has stopped moving. Similar problem was to decide the numbers of rectangles for NPC since I wanted to make the collision as accurate as possible with them, in order to include dialogues when the chararcters touch.
* **Animations:** The difficulty with this was to firstly calculate the exact row and columns for each sprite for animations in source image (the one inside assets folder) and also to learn how the frameticks work to load one image after the other in cycle-fashioned way. Also the keyboard input was a little pain when I introduced running and jumping, since they are only doable when the character is moving and I created a lot of new variables to take care of it, and it is still not perfect (in fact if the player keeps space key pressed while changing direction the speed is not kept as the ‘running speed’, for now it can only change direction of run if the space key is released and then pressed again.
* **Dynamic Dialogue Management**: This was not difficult per se (there is a sub library of SDL\_TTF that does the majority of the job) but I struggled when I tried to display the dialogue boxes exactly above the nPC that are talking with the player, since all the coordinates computed dynamically.