# Attack On the Ghost

**Contributors:**

- IMT202072 – BTV Sumanth
- IMT2020078 – Shubhanshu Agrawal
- IMT2020079 – Akella V S S Surya Narayana Sastry
- IMT2020121 – Manan Patel
- IMT2020512 – Aakash Khot

**Contributions**:

- Akella V S S Surya Narayana Sastry: Design the start screen, Help in maintaing the number of ghost on screen .
- Aakash Khot:Design player, added music, implemented the energy of the hero,help in designing bullet class.
- Shubhanshu Agarwal: Renders the string above the ghost and took the input from the player.
- BTV Sumanth: Design the Ghost path in a sinusoid manner, implemented the hero and score, design end screen and help to implement basic game logic like when and how things will happen in game.
- Manan Patel: Design the start screen and the end screen of the game, integrated the code from other teammates, design the bullet path, implemented the game logic.

**Project Github Link:**

- Github

**Classes formed in the code:**

- Game: Contains methods like update, render, event handler and ghostHandler.
- Ghost, SinGhost(derived class of Ghost): Handles the ghost movement.
- Player:Handles the player movement when firing firebolt.
- Bullet: Handles the firebolt movement when fire.
- StartScreen: handles the buttons to start and quit the game. Also present the highscore of the game and also implements the endscreen.
- Score: Update and display score and maintain the highscore.
- Energy: Maintain the energy level of the ghost. If Energy became zero then shift to end screen.
- String_check: Generates random string above the ghost.

**Abstract:**

Our team chose to create a game with all the basic elements such as a character, scoreboard, movement of objects, collisions, and music. In this mini-project, we focused on two main aspects other than just using SDL to complete it. We are implementing all the concepts that we learnt throughout this course. We also tried to implement the programming paradigms of Cpp to the maximum extent we could while

using all the functionalities provided in the SDL library. This game can be played with the help of control access of the keys on the keyboard.

The game is designed keeping the following hurdles in mind. The player (User) needs to be quick on his fingers. The game starts with the hero in the middle of a battleground with himself surrounded by the ghosts. The user should type the string present on the ghost in order to eliminate it. If the string is typed correctly, an incoming firebolt kills the approaching ghost. Difficulty of the game will gradually increases as the score increases. The images used on screen are taken from web and they have been rendered. Music has been added to the game to give the player an experienced edge.

**Introduction:**

Our major aim is to explore the features of SDL2 and implement the things we learnt in the current course. In the fast-growing field of software engineering and development and even more rapidly developing gaming sector, the future is hard to predict. A game is just more than any software as it must provide content that is more enjoyable and fun. The game we designed is a single-player and multi-level game with our protagonist (character) to kill all the as per instructions that are mentioned.

**System Requirements:**

- Operating System: Windows 10/ Any Linux distro
- Cpp Intellisense
- SDL2

- Any Text Editor or IDE (Visual Studio and Visual Studio Code have been used)

Instructions for installing SDL2:

For Windows: Go to the SDL2 website and head to the download page. Under Development Libraries choose SDL2-devel- [VERSION NUMBER] - VC.zip. Unzip and you will see several folders with the names *docs*, *include*, and *lib*. *include* contains the header files which we will be needing soon. Create a simple C++ VS project and copy/move the *include* directory in your project folder. Same thing with *lib*. Now in *lib* there exists two folders: *x64* and *x86*.

In *x86* contains the 32-bit version of the library whereas in *x64* contains the 64-bit version

For Linux (Ubuntu): Enter the following command in terminal:

sudo apt-get install libsdl2-dev

**Demonstration:**

- This game starts with an opening screen which consists of the instruction to start the game, high scores of our game. We can also choose to start or quit the game like any generic game.
- After following the instruction to start the game, it displays the second screen where our game starts. It consists of the following: Our main character at the center of the screen, and ghosts coming

to attack the protagonist. The ghosts come in a proper pattern and all the ghosts come with a text on top of them. The text the user inputs is shown at the bottom left corner of the screen. The main task of the player is to kill the ghosts i.e type the string present on them.

- The game completes if all the ghosts are killed and the player loses if the ghosts manage to kill the player. The player gets up to a maximum of 5 lives. The ghosts even change their appearance as they approach the protagonist.
- Music was also added according to the theme of the game.
- We have used the concepts of the OOP and implemented the constructors and destructors.

**Challenges faced:**

- There were many places where a memory leak has been found and it was slowing down our game.
- Removing ghosts after they have been killed
- animation of the hero on killing the ghost was not working
- Making the update method to positions of all the objects.

**Future Aspects:**

- Increasing number of levels and difficulty (speed of the ghosts)
- Increase the length of strings on the top of ghosts
- Implement something which takes more time for user to crack, such as mathematical equations.

- Include more immersive objects such as power ups.
- Running the game infinitely with some powerups and added obstacles.

**References:**


1. https://dev.to/noah11012/using-sdl2-in-c-and-with-c-too-1l72
2. https://www.parallelrealities.co.uk/tutorials/#shooter
3. https://www.flaticon.com/4.