CS 319

Object-Oriented Software Engineering

Final Report

Defender

GROUP 2-D

Taha Khurram - 21701824

Mian Usman Naeem Kakakhel - 21701015

Muhammad Saboor - 21701034

Sayed Abdullah Qutb - 21701024

Balaj Saleem - 21701041

1. Implementation

Following the project analysis report (1st iteration), our group started working on the implementation of our project. All of our group members wrote the code in C++ language on the text editor for Linux, we all chose to code in Linux due to our dependency on g++ command to run our game. In order to keep a track of all the code, all group members uploaded their work onto GitHub where one person would merge the codes together. The process was simple, first each individual would attempt to complete their task as much as possible after which they would go to the unstable branch on GitHub and upload the files in the src or resources folder, if there was a file with the same name already present then they would have to first download that file from GitHub and then merge their code onto it after which they could upload the file onto GitHub, this precaution was necessary to prevent loss of data by overwriting our code. Each of us would upload the code to a separate branch and not directly to the unstable branch after which that person would merge into the unstable branch and finally he would send a Pull request to stable branch and the one managing the GitHub repository would merge the files onto the stable branch.

As we had our class diagram ready, we decided to split the work into even parts so that we may be able to implement as much of the code as possible in the time available to us, which is explained in detail below:

The work assigned to Taha was the weapons class, its subclasses and the menu frame. Taha was able to write the code for MassDestructionWeapon, the clear bomb screen class, the weapon of mass destruction class, the asteroid class, the bullets class, the blue bolt class, red bolt class and the orangeGlow class. He was unable to complete the code for the MenuFrame and the MassDestructionWeapon was not implemented due to time constraints.

Saboor was assigned to code the ShopFrame, its clickListener and the DataManager class. Saboor was able to complete the code for the DataManager class and also performed its unit testing, he was also able to complete the ShopFrame class but was not able to complete the ClickListener for ShopFrame due to which we could not implement it into the main game yet. Saboor also helped in the integration of the classes into the game frame, including the enemy and the asteroid class, so that we may be able to have as many of our features as possible.

Usman was assigned to work on the camera, display manager, event listener for the keyboard, and the Game Frame. He was able to complete the classes assigned to him and worked on the integration of the human, weapons and enemy classes. In the game frame he completed the spaceship class and was able to complete the camera class which included the HUD which shows the high score, life and fuel of the spaceship. He further made the point class for x,y coordinate which were used in implementing and updating the UI with other classes such as weapons, enemy and humans. He worked with Saboor in integrating the asteroid and enemy classes. As of now he was able to make the spaceship move, and shoot bullets along with an asteroid being generated on the screen and enemies being able to spawn on the screen. However, he was unable to complete the collision class which is needed to deal damage to the spaceship and the enemies.

Balaj was assigned to write the code for the Enemy class and its subclasses and the code for the Pickup class. He was able to write the code for the Enemy class but was not able to complete the code for all of its children classes and we were able to implement one of the enemy children classes into our game.

Abdullah was assigned to write the code for the WarpZone class, the Human class and the Item class. He was able to complete the code for the Human class and the WarpZone class, and we were able to integrate the Human class with the game. He wrote the Item class and its subclasses, DamageItem, ShieldItem, SkinItem, but these have not been implemented yet because we do not have the Shop feature of our game.

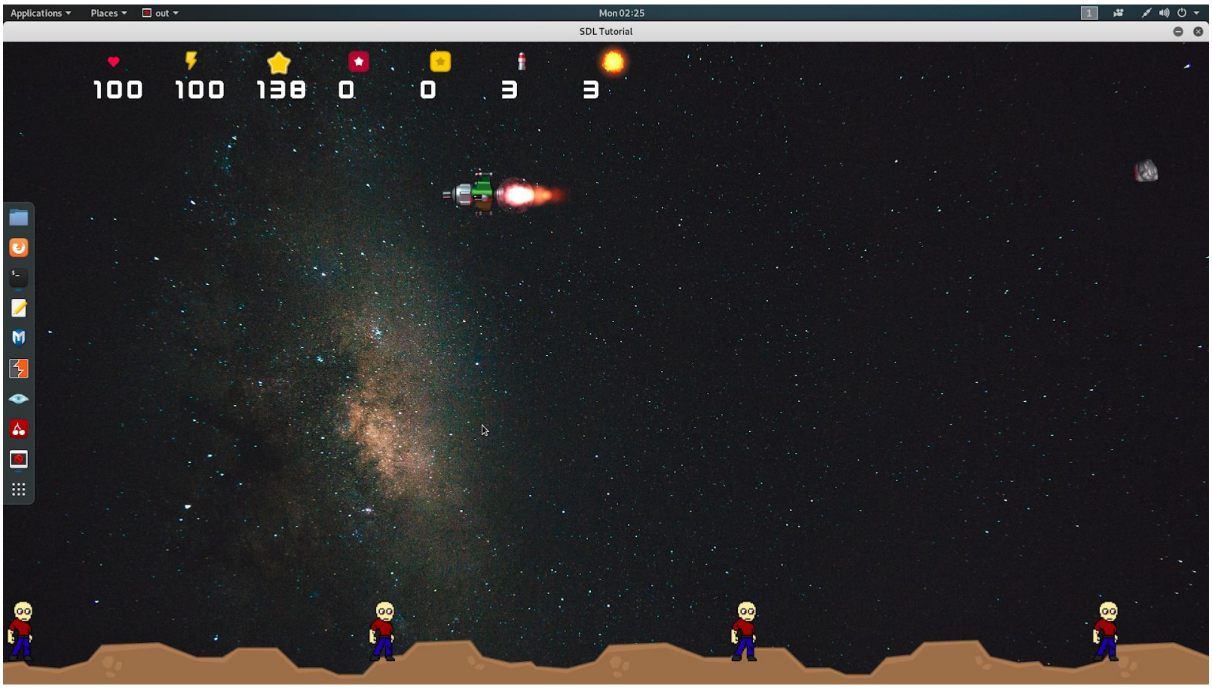
After assigning the code we would usually discuss the classes assigned to us on the WhatsApp group and hold meetings to discuss the changes to the code we have made in order to make the implementation step easier. We divided our project while keeping in mind our class diagram which made the implementation step a lot easier, however due to ongoing midterms for each member in our team we were unable to complete the code in time for our game. We would have also liked to meet more often but due to conflicting schedules we were unable to meet much and most of our interactions and discussions were held on the WhatsApp group. It was due to these reasons that we were unable to complete our game till the deadline.

2.Work done

As of now, we have a Spaceship in the game which supports all of its movements and can shoot bullets, we also have one of our enemies implemented which is called the saucer which chases the spaceship, we have an asteroid event implemented which will spawn an asteroid on the top of the screen and it will fall down, the humans have also been added to the game. In regards to the HUD we are able to see the highscore, health, the amount of fuel left and the remaining number of missiles and the clear screen bombs. The high score is also updating with time as the player survives through the game.

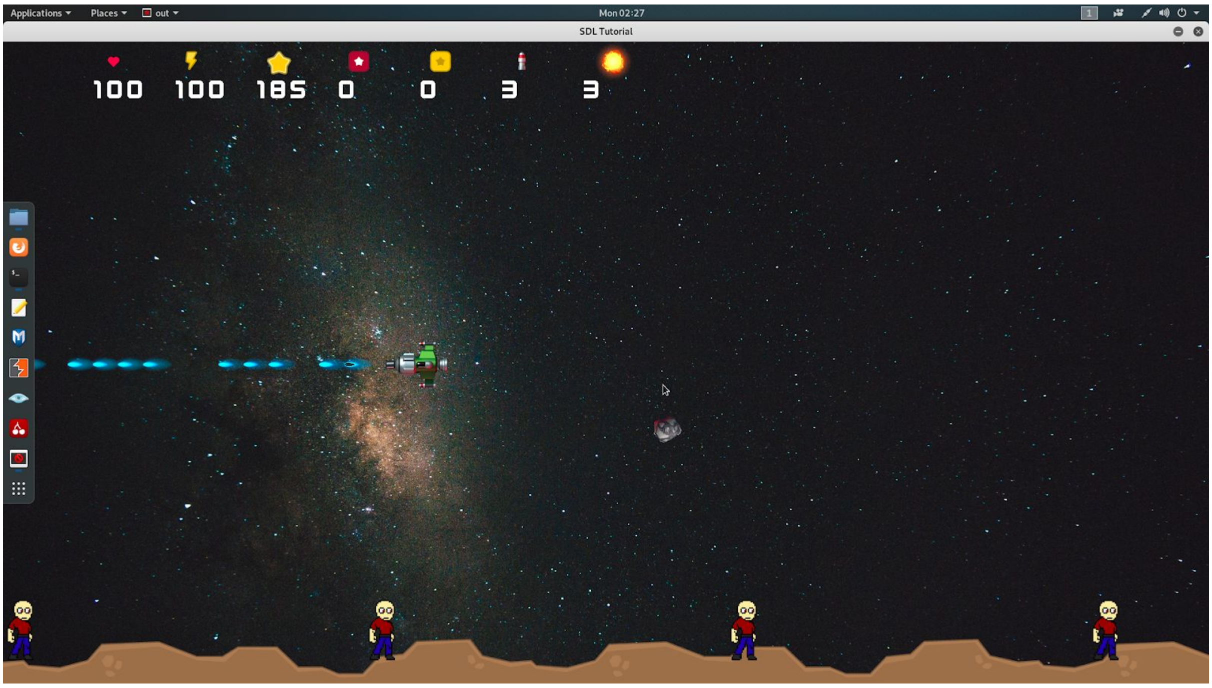
3. Scenes from the game

1) Start of the gameplay



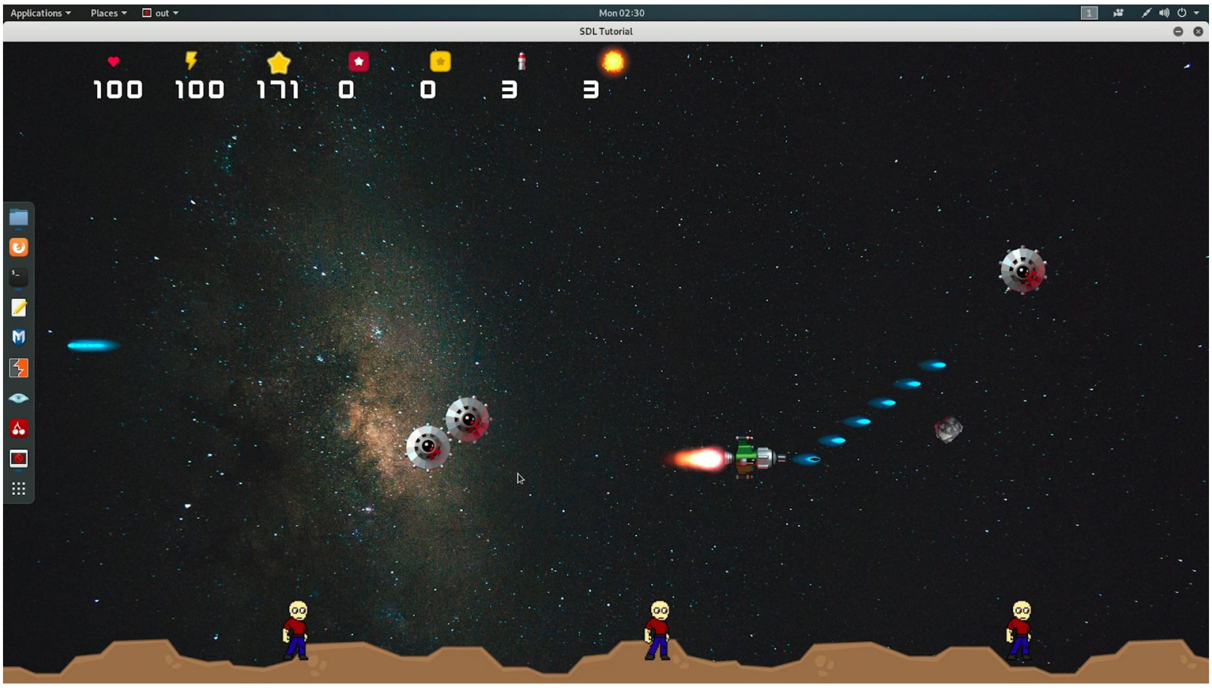
This is the main frame of the game while gameplay. This is the very beginning of the game when the enemies have not appeared yet. The spaceship can be seen looking at the left side and moving towards that direction, hence the thrustors are on. Player can control the spaceship with the arrow keys in this frame and move the spaceship to any direction wanted. The thrustors turn on and off every time the spaceship starts moving and is on until the spaceship becomes stationary. The player can also fire bullets with the spaceship. The spaceship health and fuel level can be seen on the top left corner of the screen. Next to the fuel level is the player's high score. Also the humans are stationary on the ground and are waiting to be picked up by an enemy.

2) Asteroid in the game



This frame of the game shows how an asteroid is falling downwards from the sky. The asteroid, as it can be seen, is the stone-shaped object in the center of the screen. The asteroid has the power to damage the spaceship if it collides with it. Significant amount of damage will be caused to the spaceship if it collides with the asteroid. The spaceship can be seen firing towards the left and is avoiding the asteroid.

3) Spaceship with the enemies present:



In this frame, one of the enemy types is also shown. This enemy is called the Saucer and moves randomly in the screen while shooting at the spaceship. The spaceship can shoot at the saucer and gain points while doing so. While saucer and spaceship is in motion, asteroids can come and make life more difficult for the spaceship and might collide with the spaceship.