Zense HackKnight 2K20

PROJECT : SHATTER SHOOTER

TEAM NAME:

FRUMENTUM

TEAM MEMBERS:

KAUSHIK MISHRA - IMT2020137

SHREYANSH RAI - IMT2020501

Project Description:

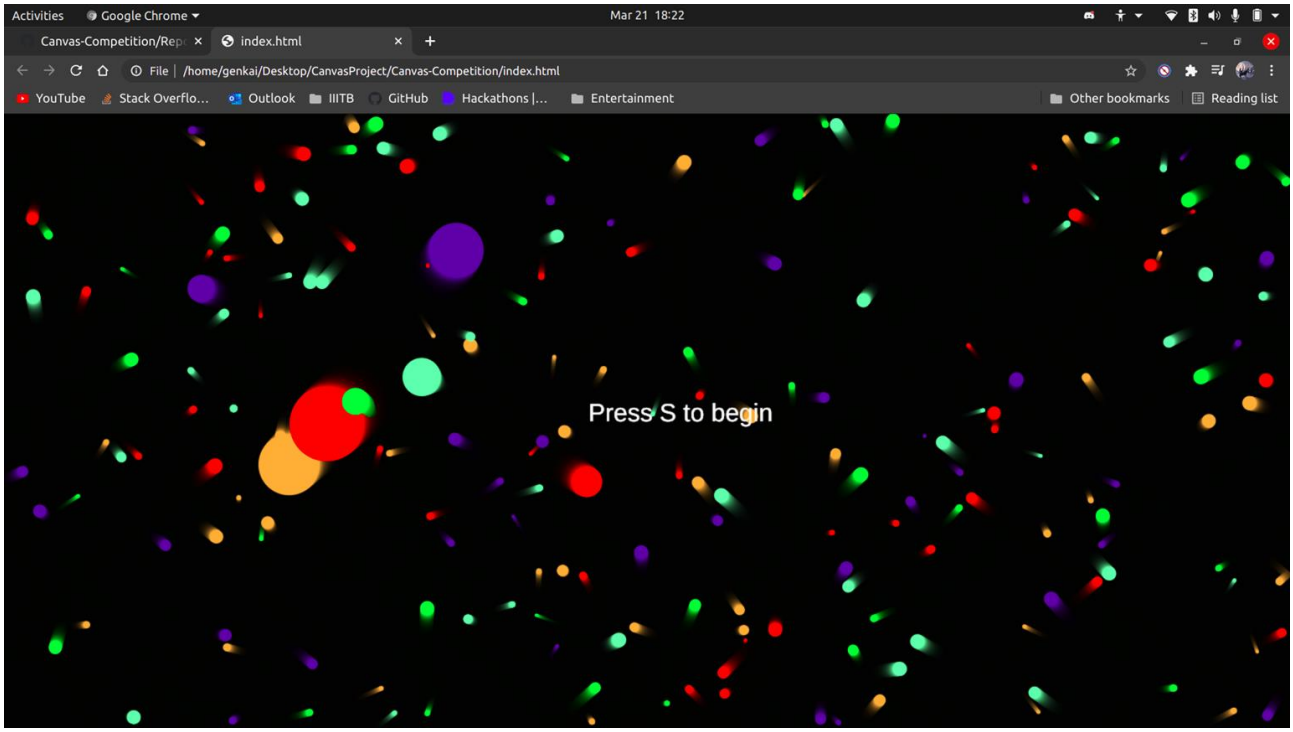
A fully realised bubble shooting game with a menu screen, a game screen and an end screen! All having beautiful animations and interactivity to engage the player while also implementing randomness and physics at all places possible.

The Game lasts infinitely and after defeating every boss the difficulty increases steadily.

The Controls

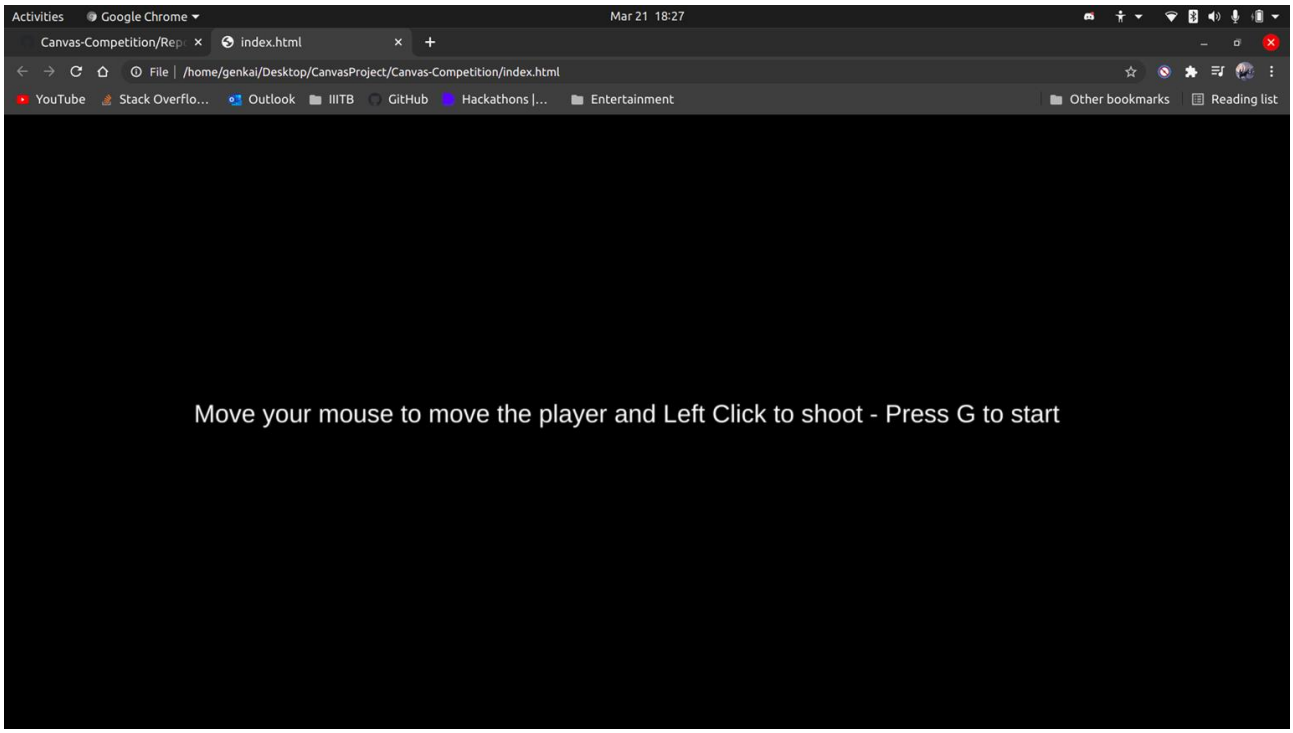
THE MENU SCREEN:

As can be seen the Menu Screen is beautifully animated with randomly spawned bubbles with random properties that bounce of the edges of the screen with a trail behind them and expand in size as the bubbles pass over the mouse pointer and contract slowly thereafter.



THE INSTRUCTION SCREEN:

Instructions are provided to the user about how the game is controlled which is inherently a simple design that uses the current location of the mouse pointer to set the player's location in the space they are allowed to move in and the player can left click to shoot.

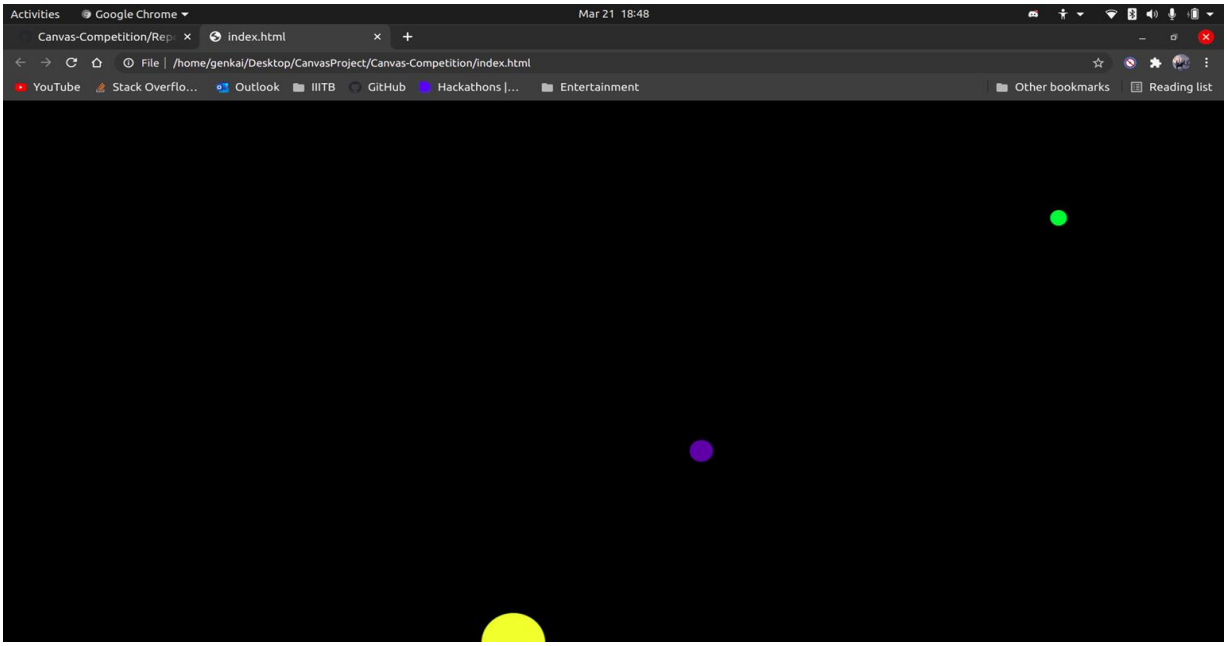


THE MAIN GAME:

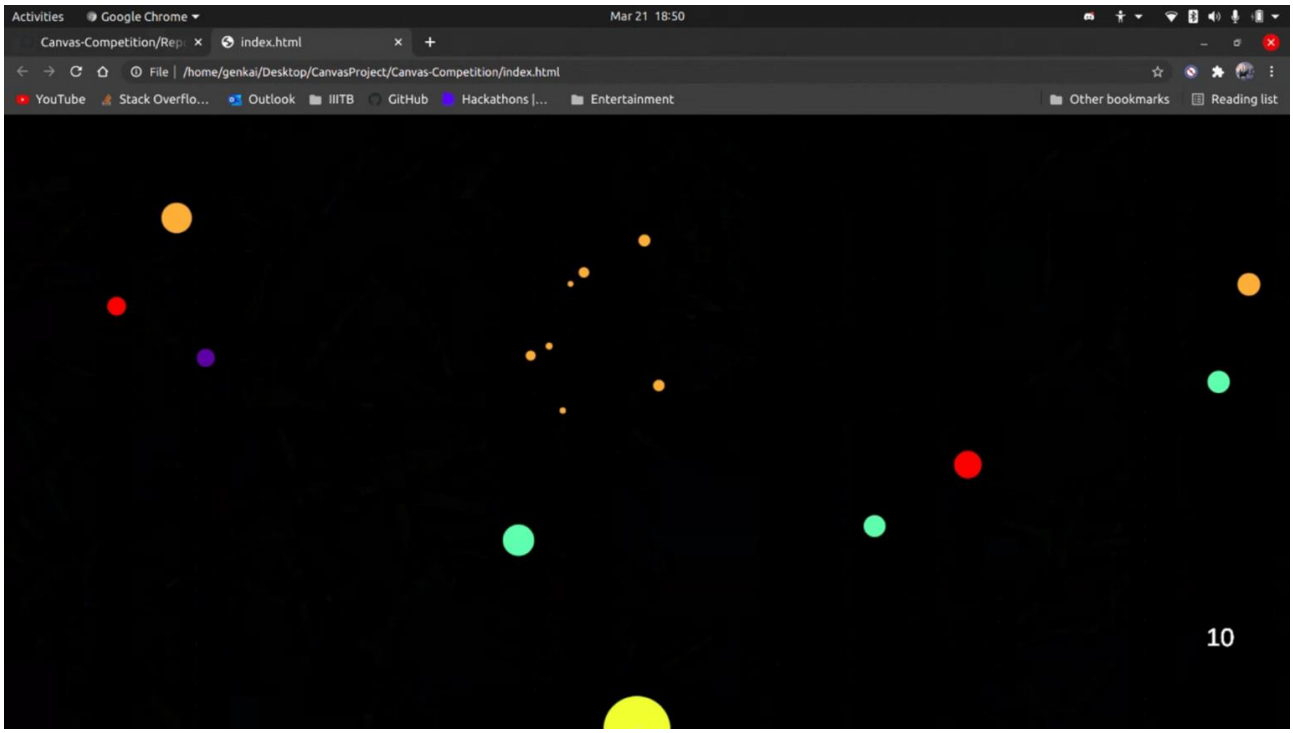
The game has a simple objective, to survive as long as possible! The player is awarded 10 points for every bubble that is destroyed and no points for dodging the bubble. If the bubble hits the player the game ends.

The Game implements physics in all places:

The Bubbles are not just bouncing with a fixed velocity. There is a slight acceleration downwards that also helps the bubbles to rebound and then come back down again



The bubble upon being shot erupt into several shattered Spheres. The number of spheres are decided based on the size of the original sphere that was burst. The colour is also assigned based on the color of the larger sphere that was destroyed.



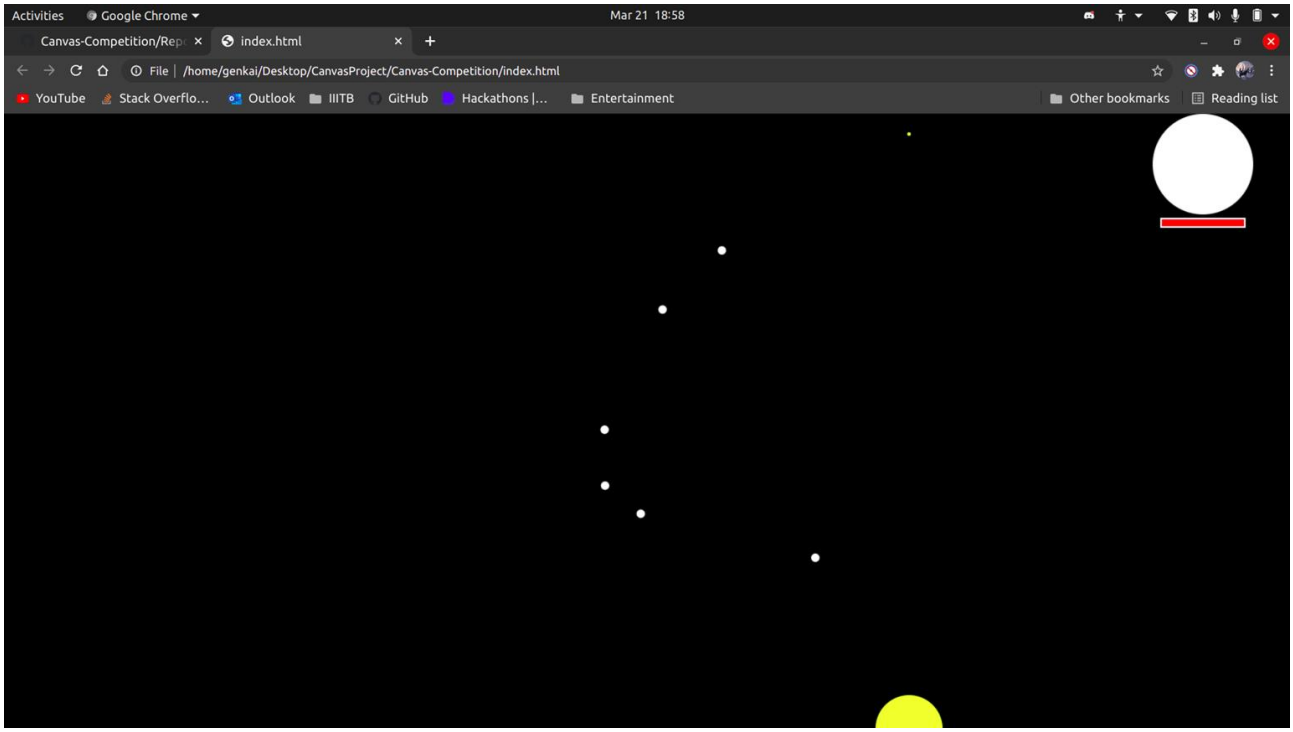
The shattered spheres have a full implementation of physics complete with gravity and a coefficient of restitution. The Shattered spheres after generation will bounce following a projectile motion due to the physics assigned to them and then bounce with an e<1. To keep the screen clutter free the

shattered spheres will sink into the screen after one bounce. The transition is kept in such a way that they do not abruptly disappear but sink into the screen smoothly.

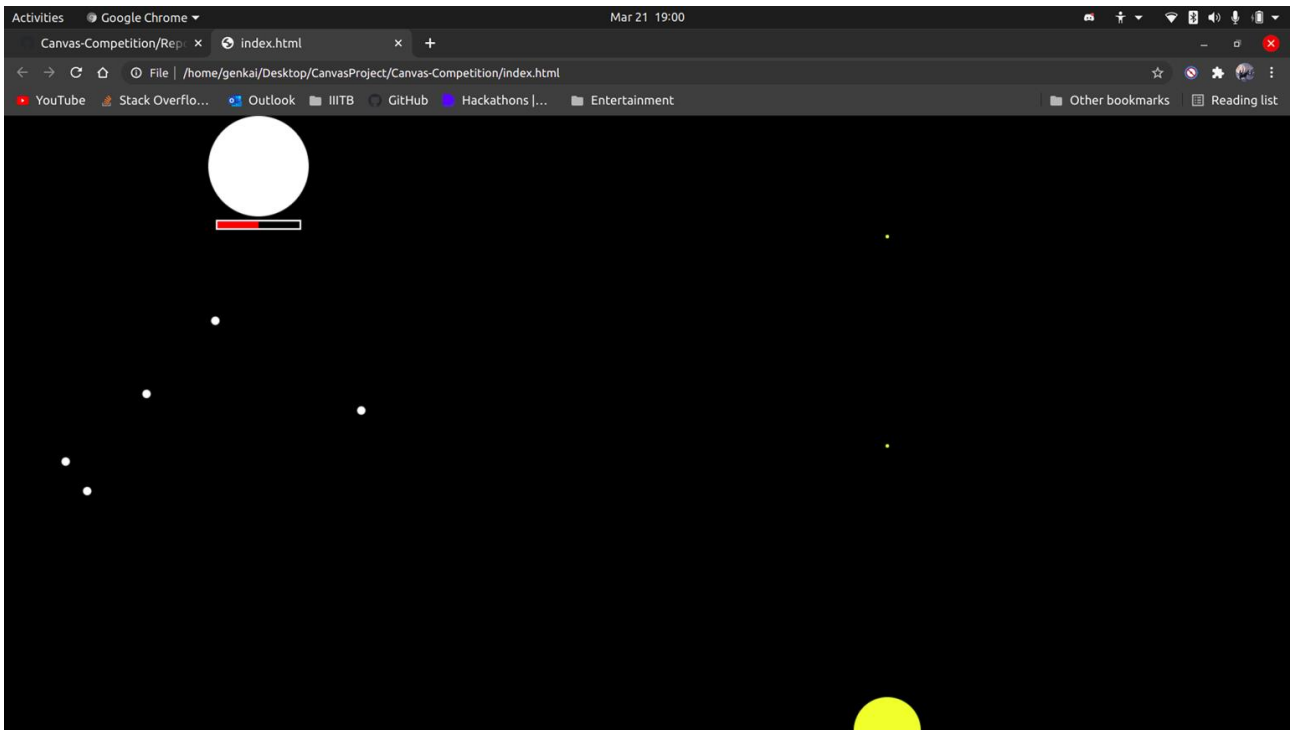
Each sphere that is shattered fetches the player 10 points and in order to keep the experience immersive the score counter has been implemented such that when the player scores he is updated on his score and then the score fades away until the next time the player scores.

After destroying the first five spheres a boss is spawned-

The boss has a set number of Hit Point or HP and this is initially equivalent to 10 shots of the player's pellets. The boss unlike the other enemy bubbles also has functionality to shoot a spray of bulletes that performs Circular motion as it sprays outwards towards the player. The Boss attack is also affected by gravity and it rebounds and changes trajectory upon coming in contact with the edges of the screen. The HP indicator is made to move with the boss and is updated with the HP remaining each time the boss is shot.



The shots fired are random and get even harder to dodge because of the movement of the shots and coupled with the boss' movements. The boss Attacks upon being shot at.



After The Boss:

The game continues after the boss is destroyed and the spawn rate for the enemy bubbles increases and more of the bubbles have to be destroyed in order to fight the boss once more.

The boss itself gets tougher to beat progressively and HP of the boss and the number of pellets per boss attack increases each time the boss is encountered.

End Game:

The player can Lose in 2 situations-

1) The Player gets hit by an enemy and dies.

2) The Boss kills the player by attacking him.

After the player dies the game moves on to the end screen and over here the EXACT bubbles that were shot down by the player appear on screen and revolve around the cursor.

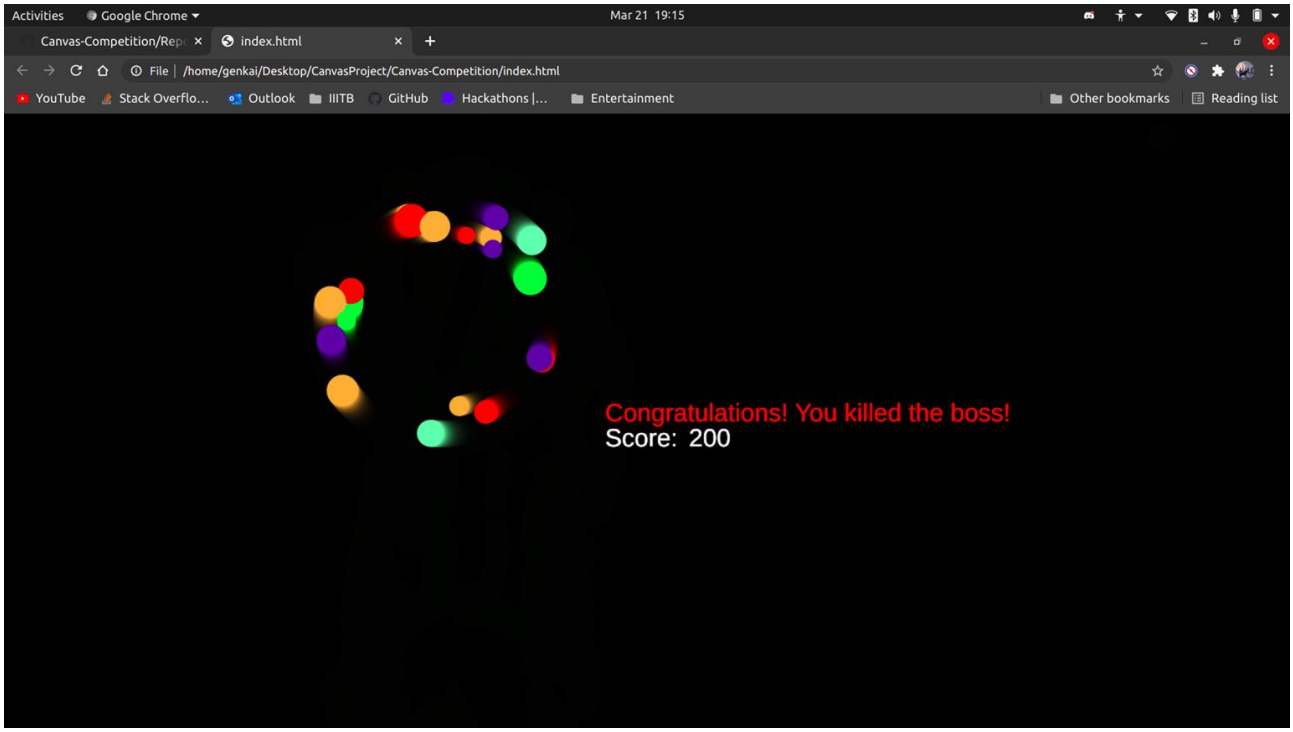
The centre of rotation is fixed at the cursor and upon clicking the mouse buttons, the ring of defeated enemies expand outwards and start rotating faster.

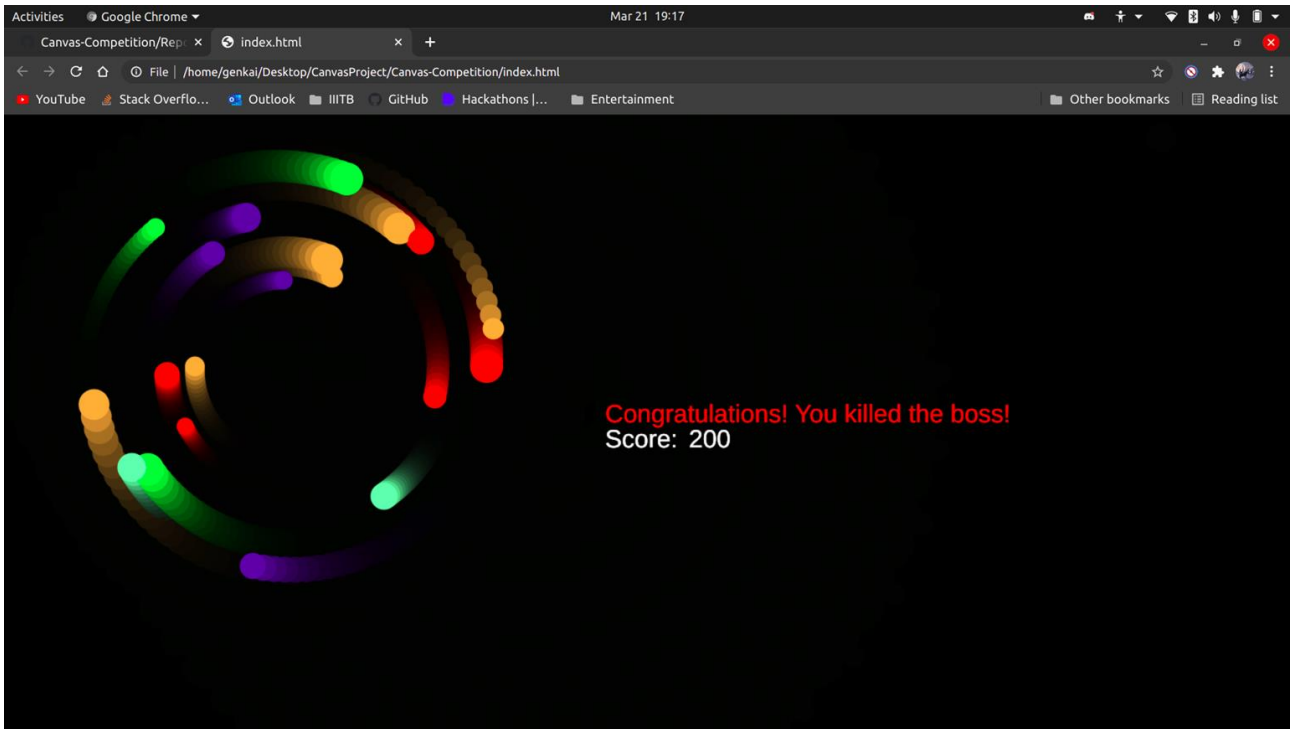
Angular acceleration has been implemented in the rotating ring of enemies and the enemies leave behind a trailing effect generating beautiful visuals.

Upon releasing the mouse button the Bubbles will start decelerating and the radius of the ring contracts until the inital position is achieved.

The user can reload the site in order to restart the game.

Or enjoy the visuals :)





OPTIMISATIONS TO CODE:

After much take aways from the initial versions of the project we realised that the resource usage of the program was really high. The RAM usage and the GPU usage were in the ranges of 2.5 GBs and 60% respectively. In order to make our project not just appealing but also efficient we set out to optimise whatever we could.

All the object arrays were spliced down to Zero length when the use for them is over. The pellets being shot by the player are also not only spliced from the array when the pellet collides with the enemy but also when the pellet exits the bounds of the screen reducing system load.

This splicing technique has been extensively used through out the program wherever possible.

INSPIRATION FOR THE PROJECT:

We initially wanted to make a physics simulator with proper collision and gravity implemented. But that isn't always fun. So we decided upon creating a game that would showcase these in a way that was simple and interesting. So we ended up creating a simple space shooter-esque game. We wanted it to have something new at every screen. So we made the menu screen a showcase from simple elastic collisions. The main game has simple gravity to keep the balls bouncing and added inelastic collision for the shattered balls. The boss shows a little bit of spiral motion through its bullets which were implemented using SHM. The end screen shows balls in circular motion and holding the mouse allows the player to make them accelerate in a circular space, increasing their radius and angular velocity.

Keeping all this mind it also had to be a fun game so we made it colourful and progressively tougher. The player is rewarded with a satisfying end screen that presents them with sort of a trophy with the rotating spheres (These are the same ones that the player destroyed as well!) that leave behind a trailing effect.

PROBLEMS IMPLEMENTING THE PROJECT:

The biggest problem for us that wasn't apparent at the beginning was optimisation. As it has been discussed above already we won't elaborate it further here.

Other problems were with the motion of the balls, the number of objects on screen, etc.

We had to keep in mind that no offscreen objects could interact with each other in any way. So in the main game, the player's pellets had to be removed as soon as they hit the top of the screen so that they don't accidentally destroy one of the enemy balls that was offscreen.

Adjusting the velocity and acceleration of the enemy balls, the bosses' pellets, the end screen balls was also a time taking aspect since the game needed to be playable as well as little challenging.

We also faced a few issues using the custom canvas provided by Zense. So initially we ended up making our project in vanilla canvas + javascript. But after a day we successfully managed to import the game and all it's features into the custom canvas template.

WHAT WE LEARNT FROM THE PROJECT:

Learning to work in a team was one of our greatest achievements in this project. The team of two allowed us to easily check each other's code whenever a bug surfaced and quickly run multiple tests to check for any new bugs that came up.

As mechanics played a huge part in the working of our project, we learned a lot about how to implement realistic or custom physics in objects.
We also learnt a great deal about optimisation of code, how important it is to empty arrays that are no longer in use and increase the usage of classes and objects to make our code a lot more readable and easier to debug since we know where to look whenever a problem arises, which can be especially problematic as our projects get longer and longer.

REFERENCES:

https://www.w3schools.com/graphics/canvas_coordinates.asp

https://www.w3schools.com/js/js_syntax.asp

https://github.com/zense/Canvas-Competition