Pseudo Asteroids

For this programming assignment you are asked to implement a pseudo Asteroids game (a shout out to the 1980's). Your implementation for this iconic game will include the following

- Asteroids
    - At least **5 Asteroids (** Sf::CircleShapes), with the following constraints
        - Each asteroid has a different size
        - Each asteroid has a different color
        - Each asteroid moves in a random direction. Once the random direction is set, it will continue moving in that direction
        - Each asteroid has a random distance (delta x, delta y) it travels for each loop iteration. Once the delta x, delta y for an asteroid is set. The asteroid will always travel that distance on each loop iteration.
        - If the asteroid travels off screen, it will reemerge from opposite side.
- Missile
    - At least 1 missile (sf::RectangleShape)
        - Missile x-position coincides with the player's x-position when launched
        - Missile travels in a vertical direction until
            - It collides with an asteroid.
                - Remove asteroid and missile (avoid update or drawing)
            - It travels off screen
                - Remove (avoid updating or drawing missile)
- Player
    - 1 player (sf::RectangleShape)
        - Located near bottom of screen. (give yourself at least a 100 pixels from bottom) else you want see asteroids that travel past top of screen and reemerge at bottom of screen 😊.
        - Player is prevented from traveling off screen
        - Player movement is controlled with arrow keys
        - Player launches missile using space-bar
- Win or Lose
    - Player wins if they successfully shot all asteroids
        - Display a message "You win "in the graphic window (delay at least 5 seconds to revel in the success.)
    - Player loses if player collides with an asteroid
        - Display a "GAME OVER, YOU LOSE" message in the graphic window. Again, delay program exits by 5 seconds to let the agony of defeat soak in 😊

Shark Bait:

- First time you shoot an asteroid, it breaks into two smaller asteroids (0.25 pts)
- Allow player to move in four directions (up, down, left right) (0.25 pts)
- Include asteroid collision. If two asteroids collide, there direction is changed. (0.25 pts)

You are expected to use

- Acceptable programming practices
    - Comments
    - Reasonable variable names
    - No magic numbers
    - Indentation
    - Concise statements
    - Avoid spurious and unnecessary code
    - Well-formed logic expressions and loops
- You are encouraged to use std::vectors or std::array
- Must likely will need parallel arrays
- Looping statements, selection statements, logical, relational expression, flags
- Consider using random, or {random_device, default_random_engine and uniform_in_distribution in random header file
- Use textbook and https://en.cppreference.com/w/
- Movement of game pieces should be smooth and not depend on moving mouse or other items.

Screen Shot of Assignment