Spaceship War

Rules:

1. Enter Your Name and Hit Start Button
2. You will have a Jet moving with your cursor and you can move to any direction
3. The Jet will shoot bullets automatically and will have 5 life at the beginning
4. You will encounter multiple enemies on the screen, and each enemy will take 2 bullets to die
5. Enemy will also shoot bullet, lose one life if got hit, lose one life if collide with enemies
6. Bonuses:
    a. Heart: increase one life
    b. BulletBonus: increase to 2 bullet -> but back to 1 if you get hurt
7. Meteor: Colliding with meteor will lose one life
8. EnemyBoss will shoot bullet and bigBullet, takes 8 bullets to die
9. Scores BreakDown
    a. Hit Heart + 10
    b. Hit BulletBonus + 10
    c. Hit Enemy, each hit + 10, enemy Dead + another 20
    d. Hit EnemyBoss, each hit + 20, boss Dead + another 40

Parent Class:

1. **Entity**
    a. Attributes:
        i. GImage* mImage
        ii. int mXOrigin
        iii. int mYOrigin
        iv. int mXMove
        v. int mYMove
        vi. int mHeight
        vii. int mWidth
    b. Functions
        i. Entity() // Default Constructor
        ii. Entity(int mXOrigin, int mYOrigin, GImage* image) //Parameter Constructor
        iii. virtual void draw(GWindow& gw)
        iv. virtual void move()
            1. Move element (Mainly the background image) by 0.1 pixel in x and y direction
        v. Bool contains(Entity* other)
            1. Check collisions

Classes/Entities:

1. **MyJet**
    a. Description:
        i. MyJet is the fighter that represents user interaction, and it will
        ii. Move with Cursor by setOrigin(x, y) from the cursor position
        iii. Blood:
            1. By default, it has 5 blood (i.e 5 lives)
            2. If Myjet hits a Heart, it will increase one life

            3. If MyJet get hurt, it will loose one life
                a. Hit by Bullet, Meteor, Enemy, EnemyBoss, BossBullet
        iv. Bullet:
            1. By default, it shoots one bullet at a time
            2. If MyJet hits a BulletBonus, it will shoot two bullets at a time
            3. If MyJet get hurt, it will go back to one bullet at a time
    b. Attributes:
        i. Inherit from Parent Entity
        ii. Int blood, int bullet
    c. Functions:
        i. Draw
        ii. Getters of XOrigin, YOrigin, width, height, blood, bullet
        iii. Setter of Origin
        iv. Void => getHurt(), increaseBlood(), increaseBullet()
        v. Bool => Contains EnemyBullet, Enemy, Heart, BulletBonus, Meteor, BossBullet

2. **MyBullet**
    a. Description:
        i. MyBullet is the bullet from MyJet and will be showing consistently
        ii. Move down by 15
    b. Attributes:
        i. Inherit from Parent Entity
    c. Functions
        i. Draw, Move, Getters

3. **Enemy**
    a. Description:
        i. Enemy will show up on the screen in a random pattern, and shooting bullets down. MyJet will lose blood if hit by bullet from enemy or crash by an enemy
        ii. Move:
            1. XDirection: Velocity randomly get -1 0 1
            2. YDirection: Moves down by 1
        iii. Heart: 2
    b. Attributes:
        i. Inherit from Parent Entity
        ii. int XMove, mYMove, heart
    c. Functions
        i. Draw, Move, Getters
        ii. Check collisions of player's bullets
        iii. GetHurt() -> heart -1

4. **EnemyBullet**
    a. Description:
        i. EnemyBullet is the bullet from a small Enemy, and show up consistently
        ii. Move down by 8
    b. Attributes:
        i. Inherit from Entity
    c. Functions:
        i. Draw, Move, Getters

5. **Heart**
    a. Description:

      i. Heart will show up on the screen randomly, and if MyJet hits a heart, it will increase the level of blood of MyJet
      ii. Move down by 5
   b. Attributes:
      i. Inherit from Parent Entity
   c. Functions
      i. Draw, Move, Getters

6. **BulletBonus**
   a. Description:
      i. BulletBonus will show up on the screen randomly, and if MyJet hits a BulletBonus, MyJet's bullets shooting one time will increase to 2
      ii. Move down by 5
   b. Attributes:
      i. Inherit from Parent Entity
   c. Functions:
      i. Getters, draw, move

7. **EnemyBoss**
   a. Description:
      i. EnemyBoss will show up on the screen at a lower frequency
      ii. Heart Default: 8
      iii. Move:
         1. XDirection: Velocity: -1 0 1
         2. YDirection: Down by 1
   b. Attributes:
      i. Heart, Xmove, Ymove
   c. Functions:
      i. Draw, Move, Getters
      ii. Check Collision with Player's bullet
      iii. Get Hurt → Heart -1

8. **BossBullet**
   a. Description:
      i. BossBullet is a larger bullet and only shows up once when the EnemyBoss appears
      ii. Move down by 10
   b. Attribues
      i. Inherit from Entity
   c. Functions:
      i. Draw, Move, Getters

9. **Meteor**
   a. Description:
      i. Meteor is an enemy with no bullets and show up on the screen randomly, MyJet will lose one life if hit by a Meteor
      ii. Move down by 10
   b. Attributes:
      i. Inheirt from Entity
   c. Functions:
      i. Draw, Move, Getters

10. **Life**
   a. Description:
      i. Life will show up on the screen how many lives player still have left

      ii. Does not move, but change when lives number change
   b. Attributes:
      i. Inherit from Entity
   c. Functions:
      i. Draw, Getters

Main.cpp
Functions:

1. void selectSort
   a. Get a vector of int from large to small
2. Vector<int> getHighScores(ifstream& file)
   a. Read through the file and get score int
   b. Call selectSort and order from large to small
   c. Get first 10 largest numbers
3. UpdateLives
   a. Display how many lives are left of player
   b. Input: Player, lives Vector, Gwindow
4. void drawScreen
   a. Purpose: Loop through all vectors and draw each element on gw
   b. Input: Everything
5. Void moveEverything
   a. Purpose: Loop through all vectors and move each element by their own pattern
   b. Input: All Vectors
6. removeAtBorder
   a. Purpose: Remove from vectors if they are out of GWindow
   b. Input: All Vectors
7. Create Enemy; EnemyBoss; Heart; BulletBonus; Meteor [5 functions]
   a. Input: Their own vector and GImage
   b. Purpose: Add to the vector each belongs to
8. Void loadBullet
   a. Input: player, Mybullet Vector, GImage
   b. Purpose: Get Player Position and add corresponding position of bullet to the vector
9. Void load2Bullets
   a. Input: player, Mybullet Vector, GImage
   b. Purpose: Get Player Position and add 2 corresponding position of bullets to the vector
10. enemyLoadBullets
    a. Input: enemies Vector, GImage, ememyBullets Vector
    b. Purpose: Loop through each enemy and get their position and add bullet to each of them
11. bigELoadBullet
    a. Input: EnemyBoss Vector, GImage1, Gimage2, BossBullet Vector, EnemeyBullet Vector
    b. Purpose:
       i. Loop through each EnemyBoss and get their position
       ii. Add both unique BossBullet and normal EnemyBullet object to each vector
12. getHeart; hitBonus [2 functions]
    a. Input: player and each Vector
    b. Purpose:

```
                i.   Loop through all elements in each vector and check for collision
                ii.  Add life/ Bullet number
                iii. Increase score
13. Bool playerShot()
        a. Input: MyJet, Vectors(EnemyBullet, BossBullet, Enemy, Meteor, EnemyBoss)
        b. Purpose:
                i.   Check Collisions with ALL that will make player Lose Life
        c. Return the bool of whether the player is dead
14. hitEnemies
        a. Input: enemies Vector, playerBullets vector
        b. Purpose:
                i.   Check if a bullet hits an enemy, 2 hit -> enemy dead
                ii.   Each hit is 20 score, if Enemy dead add another 20
15. hitBoss
        a. Input: (enemyBoss Vector, playerBullets vector
        b. Purpose:
                i.   Check if a bullet hits an enemy boss, 8 hit -> enemyBoss dead
                ii.  Each hit is 20 score, if Boss Dead, another 40


16. Main()
        a. Create a 600x600 window
        b. Create Glabels and GTextFiles and GButtons to start and stop games  and display
           scores (SetActionCommand)
        c. Create Background Image and draw
        d. Declare all vectors and create GImage for all entities
        e. Declare a bool to track whether player is dead
        f. Create a player
        g. Initialize a Map < string, int> scores to store all scores
        h. Create a ofstream fileOut with name “results.txt”
        i. Set a Timer of 25.0 msec and initialize counter = 0
        j. While true
                i.   WaitForEvent
                        1. TIMER EVENT
                                a. Remove everything outside of border
                                b. Draw screen
                                c. If counter % 10 == 0
                                        i.   If player has one bullet -> load1bullet
                                        ii.  If has two bullets -> load2bullet
                                d. Counter % 30 == 0
                                        i.   Load enemy bullets
                                        ii.  If there is EnemyBoss -> load its bullets too
                                e. Counter %55 == 0
                                        i.   Create enemies
                                f. Counter % 200 == 0
                                        i.   Create Meteor
                                g. Counter % 300 == 0
                                        i.   Create Bullet Bonus
                                h. Counter % 400 == 0
                                        i.   Create Heart
        i.  Counter % 600 == 0
                i.  Create Big Enemy
        j.  Move Everything
        k.  Check if Player hit Heart, BulletBonus
        l.  Check if player’s bullet hit Enemies, EnemyBoss
        m.  Check if Player is Dead
                i.  If Dead
                        1.  Stop timer
                        2.  Print Game Over
                        3.  Save name and score to scores MAP and write a
                            file named “results.txt”
                        4.  Clear All Vectors and counter = 0
                        5.  Delete player and background
                        6.  Create new player and background
                        7.  Draw Screen
        n.  Counter ++
        o.  Draw Screen
2.  ACTION EVENT
        a.  Hit Play Button -> timer start
        b.  Hit Stop Button -> timer stop, write to “results.txt”
        c.  Hit Display Button -> display high scores at bottom

3.  MOUSE EVENT
        a.  Get the movement of cursor and set to MyJet
```