=Fancy Project v0.0.0.0.0.4.2.1 11 May 2018
Written May 1 - 18, 2018, by Carl, Osman, and Claire

8th period team 1: The One and Only Spooky Fancy 100% Balanced,100% Legit, Not a Scam, Free Trade, Quality Certified Guarantee Meme Team led by Three Smart Bois

Introduction:

This program is basically a shooter game which would hopefully solve two problems:
boredom and lack of knowledge or creativity for game design.
To elaborate on that, if some random person with free time happens to be bored,
that person could run this program and the boredom would cease to exist.
Other people could look at this program for both inspiration and to help understand
certain concepts crucial for game development that this program relies on.
Primarily, this program is a game. The goal of the game is to destroy as many
enemies as possible without getting hit by obstacles or enemy bullets.
The player moves up/down/left/right by using the arrow keys. Ammo is infinite.
Shooting an obstacle causes it to explode.
The player could also die if it is hit by an enemy's bullet.
The player destroys enemies by hitting them .
Enemies regenerate over time, with the sole purpose of infuriating the player.

Instructions:

Run GameWindow.java to open the game. Press S to start immediately, or press i to see the instructions first.
If you press i, the instructions show up. Pressing S while on that screen will start the game.
After starting, select a starting difficulty of 1, 2, or 3.
At any time before or during the game, the L key can be pressed to enable/disable low quality graphics.
The game starts here. Use the arrow keys to move the player. The user has limited Health Points (HP), which are
displayed in the bar at the top left corner. Orange represents how much you have left; black is HP lost. You die
if all your HP is lost. You can lose HP by getting hit by an enemy's bullet, or by colliding with an obstacle.
Your goal is to get as high a score as possible (and as high a difficulty as possible) before you die. You get points
by shooting enemies, obstacles, and for staying alive. The game's difficulty increases throughout the game, meaning
that there will be more obstacles and enemies the longer you stay alive for. Hitting Boosts and PowerUps can restore HP
to the player. If you die, press F to restart the game (and pay respects).

Classes:

ShooterGame.java contains the logic and mechanics for the game.
ShooterGraphics.java draws the objects in the game, and handles everything graphics-related.

KeyboardInput.java has the codes for commonly used keys; it makes getting key inputs easier.
GameWindow.java contains the main method; run it to open the game.
Shooter.java represents a shooter object; that is, something that can move around and fire projectiles.

- Player.java represents the character the user plays as. It is a shooter.
- Enemy.java represents an enemy, which shoots down at the player. It is also a shooter.

Projectile.java represents something that's a projectile. It is used to represent the bullets which the user and enemies shoot.
Obstacle.java represents obstacles, which destroy the player if they come into contact. There are 3 different kinds of obstacles, which are all represented by this class.
Boost.java represents things which give the player extra health points if they are hit.
PowerUp.java is similar to Boost.java, but introduces other kinds of power-ups as well.
SidePanel.java represents a panel in the frame, which displays the player's score and the current game difficulty.
Explosion.java is for drawing an explosion gif; its job here is to make things easier when we handle things getting blown up.
ProgressBar.java represents a progress bar (OH REALLY?! HOW DID YOU KNOW?!!) which can be used to display remaining health points.

Resources:

Image sources: codepen.io, Google.
Art made using Krita and MS Paint.
Ships made using blender and fusion.
Inspiration from Bullethead, Space Invaders, Asteroids.
Thanks Mr. Taylor for random suggestions, tolerating our spam, and providing gifs.
Visual VM used for profiling.

Responsibilities:

CARL: Worked on events, actions, explosions, renders (the player and enemy images) and keyboard inputs, as well as working on game mechanics/logic. A lot of debugging as well. Profiling in order to find methods to optimize in order to remove the sources of lag.
CLAIRE: Graphics and menu/UI. Made the different screens (start, instructions, difficulty). Made a lot for UI and put things together into this monstrosity of a final project here. Finding bugs and combining elements.
OSMAN: Created the original "base" structure for the game-mechanics and rules. Made Power-Ups. Made the fancy rainbow background. Combined different versions.