1+0+0+8+0+1+9+1+6 = 26 ⇒ Even

Part 2:

- Mechanic: Projectile Controller
    - Move upward
    - Destroy Galaga on contact
    - Destroy self on contact with Galaga
- This was done as a singleton as projectiles will be shot one at a time and move quickly, so there shouldn't be more than one on screen at a time.

```
void Update()
{
    // move upward at constant speed

    // if collide with Galaga {
    //      destroy Galaga
    //      destroy self
    // }
}
```

Singletons are entirely self-reliant, so other than checking for collision with the Galaga, all can be managed with no external reference.

Singleton

Part 3:

- Mechanic: Score Manager
    - Listen to each Galaga for destruction

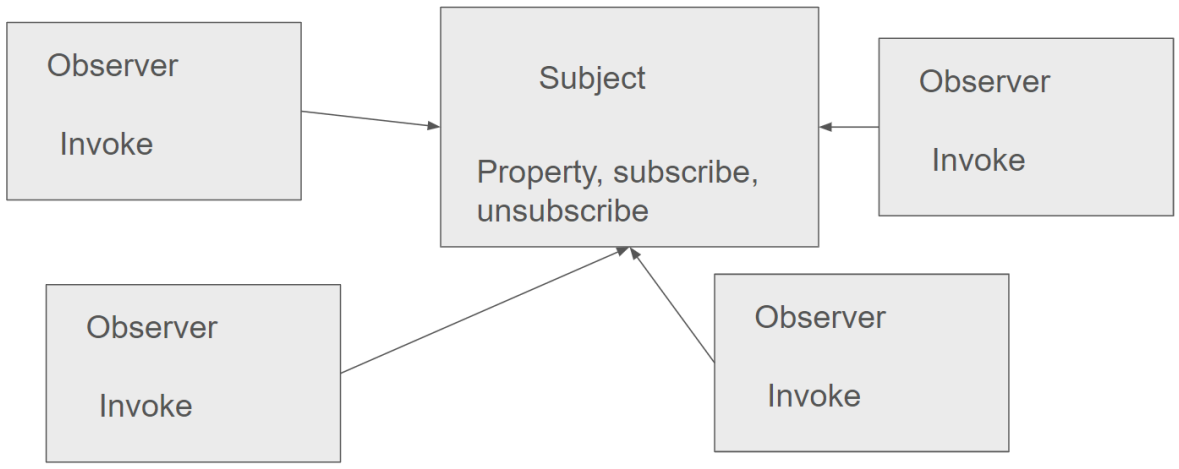
- Add score based on starting health of Galaga
- Update score + display

```
Unity Message | 0 references
void Update()
{
    // At the start of each level, subscribe to all the Galagas

    // Check if a Galaga got destroyed {
    //      unsubscribe from the destroyed Galaga
    //      Add points to score relative to health of Galaga
    //      Update score display
    // }
}
```

Observers need to subscribe to the subjects they are getting data from. In this case, it is each Galaga present at the start of the level.

Observer



Part 4:

- Mechanic would have been enemy spawner via factory
  - Create base enemy format with customizable health
  - Spawn in various locations

- This would have been done this way to create many different varieties of Galaga enemies.

Factory