1. Es un problema en donde balas estallan y dan a cañones o a otras balas,
2. Esta balas tienen un radio de impacto proporcional al cuadrado de la distancia que los separa
3. planeo asumir 3 velocidades de disparo constantes que estén disponibles para los cañones (y que dependa del modelo de cañón) que haría referencia a tipos de pólvora que se puedan usar
4. la variable en la cual se modifica para buscar un disparo efectivo es el ángulo ya que la velocidad es algo que no se puede variar de muchas maneras pero igualmente buscará entre las 3 pólvoras(que son Vi)
5. con estas 3 pólvoras la idea es lograr los 3 intentos que se piden en el texto
6. lo de las velocidades puede ser un limitante ya que al tener una velocidad “constante” el rango de disparos se podría ver limitado

Análisis de clases

clase tipo bala:

- **atributos:**

pos [2]: será un arreglo sencillo que contendrá la posición en (x,y) <float>

vel [2]: será un arreglo sencillo que contendrá la velocidad en (x,y) <float>

ace [2]: será un arreglo sencillo que contendrá la aceleración en (x,y) <float>

tiempo <float>: el instante de tiempo en el que se encuentra la pelota de esta depende su velocidad y posición

inPOS [2]: será un arreglo sencillo que contendrá la posición inicial en (x,y) <float>
inVEL [2]: será un arreglo sencillo que contendrá la velocidad inicial en (x,y) <float>

- **métodos:**

bala(inpos,invel): inicializa la bala

avanza(t): esta función hace que la bala avance en el tiempo

set_new_inpos(inpos): le das una nueva posición inicial

set_new_invel(invel): le das una nueva velocidad inicial

getStarted(): reinicia la simulación y vuelve a el tiempo 0

son privados todos los atributos excepto pos, vel y tiempo (pero no se recomienda modificarlos) el resto son públicos

escenario de pelea:
clase escenario:

atributos:

balaD: bala del cañón defensor
balaO: bala del cañón ofensor

delayO <float>: tiempo que se demora en llegar la información espia
delayD <float>: tiempo que se demora en llegar la información espia

radAO <float>: radio de la explocion que provoca la bala atacante ofensiva
radAD <float>: radio de la explocion que provoca la bala atacante defensiva
radDO <float>: radio de la explocion que provoca la bala defensiva ofensiva

angO <float>: angulo de disparo del cañón ofensor dom($\pi/2$, π)*
angD <float>: angulo de disparo del cañón Defensor dom(0, $\pi/2$)*

velO <float>[3]: velocidades disponibles en el cañón ofensor
velD <float>[3]: velocidades disponibles en el cañón defensor

métodos:
1. Generar disparos (al menos tres) ofensivos que comprometan la integridad del cañón defensivo.
2. Generar disparos (al menos tres) defensivos que comprometan la integridad del cañón ofensivo.
3. Dado un disparo ofensivo, generar (al menos tres) disparos defensivos que impida que el cañón defensivo sea destruido sin importar si el cañón ofensivo pueda ser destruido.
4. Dado un disparo ofensivo, generar (al menos tres) disparo defensivos que impidan que los cañones defensivo y ofensivo pueden ser destruidos.

5.) Dado un disparo ofensivo efectivo y un disparo defensivo que comprometa la efectividad del ataque ofensivo, generar (al menos tres) disparos que neutralicen el ataque defensivo y permitan que el ataque ofensivo sea efectivo.