# Vectors, moviment autònom i OOP:

## Enunciat

Resol els següents punts usant Processing:

1- Llegeix i explica el funcionament de cada una de les classes que es donen en aquest document. **(0.5 punt)**

2- Com adaptaríeu el codi de l'exercici del cercle que rebota usant la classe PVector. Es recomana usar variables de tipus PVector per a la posició i per a la velocitat. Llegiu el capítol 1 sobre vectors del llibre online gratuït The Nature of Code de Daniel Shifman. **(1 punt)**

3- Usant la Classe PVector dibuixa una línia que vagi del centre de la finestra fins a la posició del punter. Com podem calcular la **magnitud** (*length*) de la línia? Mostreu un text per pantalla amb el valor de magnitud de la línia creada. Llegiu el capítol 1.5 de The Nature of Code. **(1 punt)**

4- Normalitza la línia de l'exercici anterior per a que apunti sempre al punter amb una magnitud constant. Quin valor té ara la magnitud? Què vol dir que la línia utilitza un vector unitari? Llegiu el capítol 1.6 de The Nature of Code. **(1 punt)**

5- Usant OOP, creeu una classe *Character* que utilitzi el moviment de la classe *Walker* (herència). Usa la nova classe *Character* per crear un objecte de nom *Warrior* i un altre objecte de nom *Enemy*. Aquests dos nous objectes han de tenir les següents característiques:

El *Warrior* ha de tenir 5 vides, ha de ser un cercle de color verd de 30 unitats de radi, ha de portar un text al damunt on s'indiqui el seu nom i el nombre de vides. Recorda que cal que es mogui segons la classe Walker.

L'*Enemy* ha de tenir 3 vides, ha de ser un cercle de color vermell de 30 unitats de radi, ha de portar un text al damunt on s'indiqui el nom, el nombre de vides i la distància respecte al *Warrior*. Recorda que cal que es mogui segons la classe *Walker*.

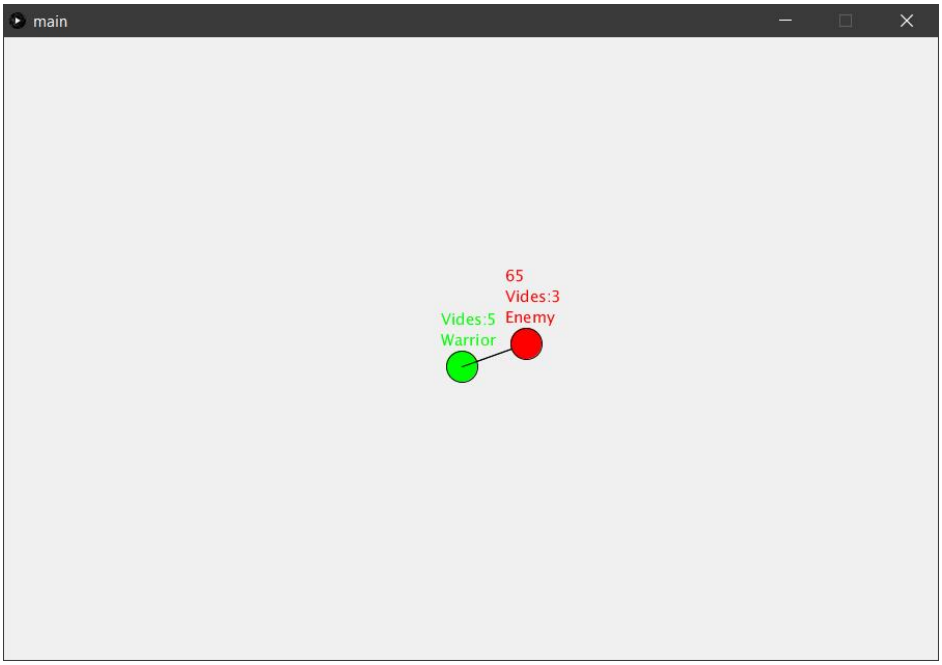
Dibuixeu una línia entre *Warrior* i *Enemy*.

**Afegiu les següents normes a l'escena:**

1. Els *Characters* han de començar al centre de la finestra però a una distància de 60 unitats horitzontals entre ells.
2. Si l'*Enemy* col·lisiona amb el *Warrior*, el *Warrior* perd una vida. Llavors els dos *Characters* han de tornar a la posició inicial.
3. Si el *Warrior* s'allunya més de 100 unitats, l'Enemy perd una vida. Llavors els dos *Characters* han de tornar a la posició inicial.
4. Si el Warrior arriba a 0 vides, perd. Si l'enemy arriba a 0 vides, perd. Elimineu l'el·lipse del *Character* que ha perdut i modifiqueu-ne el seu nom amb el text "Winner" o "loser" en cada cas.

**(5.5 punts)**

**Exemple de l'escena que es demana:**




6- Explica breument si el motor 2D que heu escollit us permet treballar d'alguna manera usant OOP. **(1 punt)**

### *Walker Class:*

```
class Walker
{
        int x;
        int y;
        Walker()
        {
            x = width/2;
            y = height/2;
        }
        void walk()
        {
            int choice = int(random(4));
            if (choice == 0) {
                x++;
            } else if (choice == 1) {
                x--;
            } else if (choice == 2) {
                y++;
            } else {
                y--;
            }
        }
        void display()
        {
          stroke(0);
          point(x,y);
        }
}
```

### *PVector Class:*

```
class PVector {
        float x;
        float y;
        PVector(float x_, float y_)
        {
            x = x_;
            y = y_;
        }
        void add(PVector v)
        {
            y = y + v.y;
            x = x + v.x;
        }
}
```

## Lliurament

Entregar el link del repo de Github amb el codi necessari. Podeu usar el Readme.md per explicar com heu resolt els apartats, adjuntar fotografies, links, gifs si ho necesiteu...