

LASER GAME (sujet de bac)

1. Instruction 3 : `joueur1 = Joueur("Sniper", 319, "A")`
2.
 - a.

```
def redevenir_actif(self):  
    if not self.est_actif :  
        self.est_actif = True
```

b.

```
def nb_de_tirs_recus(self):  
    return len(self.liste_id_tirs_recus)
```

3.
 - a. test 1
 - b. si le tir est fratricide l'équipe perd 20 points
- 4.

```
def collecte_information(self,participant):  
    if participant.equipe == self.equipe :  
        for id in participant.liste_id_tirs_recus:  
            if self.est_un_id_allie(id):  
                self.incremente_score(-20)  
            else:  
                self.incremente_score(-10)  
    if participant.est_determine():  
        self.incremente_score(40)
```

CARTES (sujet de bac)

```
class Carte:  
    def __init__(self, val, coul):  
        self.valeur = val  
        self.couleur = coul
```

b.
c7 = Carte(7, "coeur")

2.

```
def initialiser() :  
    jeu = []  
    for c in ["coeur", "carreau", "trefle", "pique"] :  
        for v in range(2,15) :  
            carte_cree = Carte(v,c)  
            jeu.append(carte_cree)  
    return jeu
```

3.

La structure des données la plus adaptée est la file, puisque l'on a affaire à une structure de type FIFO (First IN First OUT). Le classement des cartes doit suivre la "règle FIFO", car la carte remportée (la dernière arrivée) doit être placée en dessous du tas.

4.

```
def comparer(carte1, carte2):  
    if carte1.valeur > carte2.valeur :  
        return 1  
    elif carte1.valeur < carte2.valeur :  
        return -1  
    else :  
        return 0
```