

*Rapport de  
professionnalisation*

*[nom de l'étudiant]*

# Table des matières

I. Prise de contact, contexte, objectifs du stage et moyens mis en œuvre.....	3
A. Prise de contact.....	3
B. Contexte d'organisation du stage.....	4
C. Objectifs du stage et moyens mis en œuvre.....	4
i. Projet de [nom de l'étudiant].....	4
ii. moyens mis en œuvre.....	4
iii. Les objectifs du stage :.....	5
II. Compétences déjà acquises et/ou à vérifier.....	5
A. Scratch.....	5
i. Texte sur Scratch écrit par [nom de l'étudiant]. .....	5
ii. programme fait par [nom de l'étudiant].....	6
B. Godot Engine.....	7
i. Description du projet.....	7
ii. Compétences.....	8
III. Déroulement du stage.....	8
A. Jour 1.....	8
B. Jour 2.....	8
C. Jour 3.....	9
D. Jour 4.....	10
E. Jour 5.....	12
F. « Jour 6 ».....	13
IV. Annexes.....	14
A. Compétences.....	14
B. Le code des programmes.....	15
i. Jour 2.....	15
ii. Jour 3.....	15
a) 3-2.....	15
b) 3-3.....	15
c) 3-4.....	16
iii. Jour 4.....	17
a) 4-2 : tir d'un seul projectile.....	17
b) 4-3 Tir de plusieurs missiles.....	18
iv. Jour 5 :.....	19

Je me présente : Eric-Roger GARCIA,

Je suis en train de préparer un projet d'autoentreprise dans le domaine informatique

- Formation à l'utilisation de l'informatique à destination principalement des particuliers (dont la programmation)
- Création, suivi de sites internet
- Création de programme (en particulier de jeux vidéo éducatifs en partenariat avec des graphistes, musiciens...)

J'ai été de nombreuses années animateur socio-culturel. Ma spécialité était le multimédia (en particulier l'utilisation de la programmation de jeux vidéo par les enfants). J'ai de l'expérience dans le travail avec des handicapés (notamment des déficients visuels, et psychiques).

Je suis titulaire d'un BEATEPJ option multimédia.

Mon site internet professionnel :

<http://ericrogergarcia.legtux.org/>

## **I. Prise de contact, contexte, objectifs du stage et moyens mis en œuvre**

### **A. Prise de contact**

Depuis quelques mois, je vois [nom de l'étudiant] une fois par semaine, chez lui, pendant deux heures environ pour l'aider dans ses projets de programmation avec le moteur de jeu vidéo professionnel Godot Engine. Cela a permis la mise en place d'une relation équilibrée et détendue.

Nous avons fait, au début, des tentatives dans d'autres environnements (à la médiathèque de Lorient par exemple) mais, pour l'instant, il a besoin d'être dans un lieu familier pour être tranquille et rester concentré.

Mon objectif premier a d'abord été la qualité de cette relation en prenant le temps de comprendre son fonctionnement particulier. J'ai aussi consulté avec attention des documents sur l'autisme de type Asperger et discuté longuement avec sa mère qui a répondu à mes questions.

J'ai constaté que [nom de l'étudiant], dans cet environnement familier (puisque je le vois chez lui), peut entrer en relation pour apprendre de nouveaux concepts. Bien sur, des difficultés relationnelles surgissent régulièrement mais elles sont traitées au fur et à mesure calmement avec l'aide de sa maman qui le connaît bien et peut m'expliquer ce qui se passe. Ces difficultés se font plus rares puisque je comprends de mieux en mieux son fonctionnement.

La programmation est son « centre d'intérêt spécifique » au sens de l'Autisme Asperger. Il comprend rapidement les nouveaux concepts que je lui présente d'une façon étonnante vu son jeune âge.

Il programme depuis plusieurs années déjà, a réalisé seul un certain nombre de petits projets et est déjà décidé à devenir un professionnel dans ce domaine.

## B. Contexte d'organisation du stage

Le projet initial du stage du Brevet des collèges devait être fait au centre social [nom du centre social] sur la proposition de l'animateur de l'espace multimédia. Il avait proposé 1h par jour sans préciser de contenu précis. L'envie de [nom de l'étudiant] (et son projet pour le stage / au Brevet des collèges) était de s'améliorer en programmation / son projet professionnel or l'animateur du centre social ne fait pas de programmation.

J'ai donc logiquement fait une proposition (cf ci-dessous).

[nom de l'étudiant] a déjà acquis seul un certain nombre de compétences (cf ci-dessous) en programmation. Il utilise activement internet : a créé une chaîne sur Youtube pour parler de ses passions (jeux vidéos, programmation). Il crée des tutoriels sur les outils qu'il utilise pour fabriquer des programmes (Scratch, Godot Engine). Il est en relation avec d'autres programmeurs (généralement légèrement plus âgés) via différents types de forums. Il a notamment rencontré, sur le discord du CNED, un autre enfant qui programme également.

Il est intéressant de noter que ces échanges ne se bornent pas toujours à l'utilisation d'un ordinateur car il a pu voir « en réel » certain d'entre eux, notamment, un autre enfant habitant près de Rennes qui est déjà venu chez lui.

J'ai pu noter que sa mère est très attentive et se tient bien au courant des personnes avec lesquelles il est en relation et de leurs échanges.

## C. Objectifs du stage et moyens mis en œuvre

### i. Projet de [nom de l'étudiant]

[nom de l'étudiant] a opté pour le Parcours Avenir pour continuer sa réflexion sur son **avenir scolaire et professionnel et, en particulier, sur le métier d'analyste-programmeur.**

« Une capacité d'analyse importante est demandée. En effet, l'analyste-programmeur doit bien comprendre les besoins et attentes des futurs utilisateurs du programme. Il doit ensuite faire preuve de rigueur, de logique, d'inventivité et de précision dans ses tâches. Sans ces qualités, le travail deviendra tout simplement impossible. Une bonne compréhension de l'anglais est indispensable, car à haut niveau, l'accès à une documentation pointue se fait uniquement dans cette langue. Enfin, le travail en équipe est aussi important que le travail en autonomie. ».

(extrait de la page : <https://www.regionsjob.com/observatoire-metiers/fiche/analyste-programmeur> )

[nom de l'étudiant] a une bonne compréhension de l'anglais. Un exemple de vidéo qu'il comprend et peut traduire : <https://www.youtube.com/watch?v=2CeurT8vjU0>.

Ceci est, bien sur, une reformulation de nos échanges. Cela a guidé la forme du stage que je lui ai proposé.

### ii. moyens mis en œuvre

**Actuellement**, [nom de l'étudiant] apprend à utiliser le logiciel **Godot Engine**.

C'est un programme professionnel qui sert principalement à créer des jeux vidéo (pour information, c'est l'équivalent, en logiciel libre, du programme privé Unity). Le logiciel permet de créer des jeux en 2D, 3D, sur ordinateurs, smartphones, internet...

Cet outil un destiné à des professionnels, il est pourvu d'outils conséquents pour accélérer la tâche des programmeurs. C'est un avantage pour un professionnel mais cela masque aussi certains concepts (puisque beaucoup de choses sont déjà construites) qui sont indispensables à un professionnel.

C'est pourquoi, j'ai utilisé un autre système de programmation dans le cadre du stage : **Processing** (cf <https://processing.org/>) qui est un logiciel destiné, entre autre, à l'apprentissage, avec une interface très simple (pour ne pas se disperser dans des détails inutiles quand on apprend). Ce programme s'appuie sur des langages de programmation répandus (Java, Python, Javascript). Nous avons utilisé la version qui utilise le langage Java.

D'autre part, comme un programmeur professionnel ne travaille pas seul mais fait partie d'une équipe avec d'autres programmeurs, en particulier. J'ai aussi utilisé le programme **Violet UML Editor** pour faire découvrir à [nom de l'étudiant] un système de représentation graphique couramment utilisé pour représenter l'organisation d'un programme. On peut faire une analogie avec le fait d'établir des plans pour construire une maison. Ces plans servent à concevoir mais aussi à échanger des informations entre les différents corps de métiers. C'est la même chose avec les graphiques de type UML dans une équipe de création d'applications informatiques (jeux vidéo ou autre).

Je lui ai présenté et me suis appuyé dans mes explications sur le « diagramme de classes » la représentation la plus utilisée parmi les différentes représentation UML

( U.M.L : Unified Modeling Language / Langage de Modélisation Unifié )

### iii. Les objectifs du stage :

- vérifier les concepts de programmation déjà acquis
- aborder de nouveaux concepts de programmation
- découverte du “diagramme de classes” UML pour aider à la prise de recul, la fabrication d'un plan et le travail en équipe

*Ceci en s'appuyant sur l'enthousiasme de [nom de l'étudiant] pour la programmation.*

## II. Compétences déjà acquises et/ou à vérifier

Un point sur les programmes déjà utilisés par [nom de l'étudiant] et ses réalisations

[nom de l'étudiant] a réalisé des tutoriels vidéo sur ces programmes.  
(voir le lien vers sa chaîne Youtube : [lien supprimé] )

### A. Scratch

#### i. Texte sur Scratch écrit par [nom de l'étudiant]...

... à l'occasion de la réalisation d'un CV pour expliquer à l'animateur du [centre social] quels étaient ses centres d'intérêts et connaissances dans le domaines de l'informatique afin d'organiser un programme adapté allant dans la démarche et volonté de [nom de l'étudiant] de faire un stage pour en apprendre plus sur le métier qu'il veut faire :

« **Avantages:** prise en main rapide et facile même pour des personnes ne connaissant pas le codage, langue multiple, rapide, partage de projet (plusieurs personnes simultanément sur le même projet), intégration de variable\_liste\_photo\_sprite\_fond écran, personnalisations quasi infinies, tutoriel adapté pour tous, interface simple et facile d'accès, utilise flash player pour charger le créateur de jeux ou le générateur du jeu (installation facile et rapide), logiciel scratch 2 pour les créations offline.

**Inconvénients:** aucun code manuel disponible (réduit les possibilités de créations), créations de dessin trop pixelisés et pas assez précis (comparable à paint), bug de déplacement des sprites (recréation d'un personnage complexe 15min) . »

## ii. programme fait par [nom de l'étudiant]

[nom de l'étudiant] a réalisé un programme pour l'association [références supprimées]



**Page sur internet regroupant différents liens / [nom de l'étudiant] : [lien supprimé]**

Présentation de consonnes et de voyelles pour former des syllabes.

j'en déduis les compétences acquises (en particulier) :

([nom de l'étudiant] a fait une présentation lors de la sortie de l'application. L'association a convié le député, les journalistes, et parents adhérents.)

- compréhension de la notion de séquences d'instructions (les instructions sont exécutées dans l'ordre)
- utilisation de variables (affectation et utilisation des valeurs contenues dans des variables)
- choix en fonction du contenu de variables (instructions du type « si... alors... sinon »)
- utilisation de « collections » (type de variables particulier permettant d'enregistrer un grand nombre d'autres variables souvent appelé « liste » ou « tableau »)
- envoi de messages pour exécuter un groupe d'instruction donné en fonction des besoins

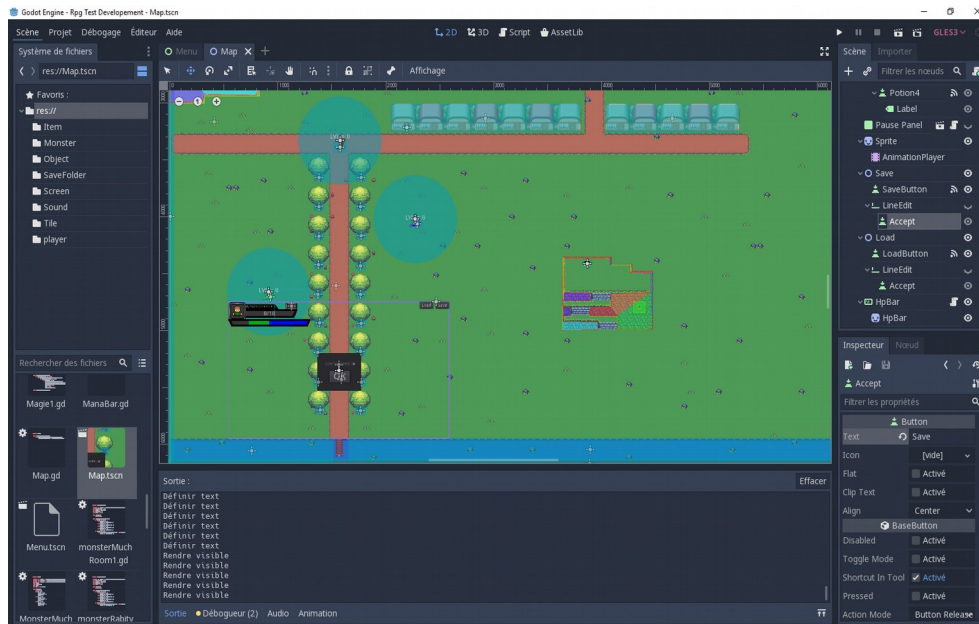
## B. Godot Engine

[nom de l'étudiant] est en train de fabriquer un « RPG » (Role Playing Game).

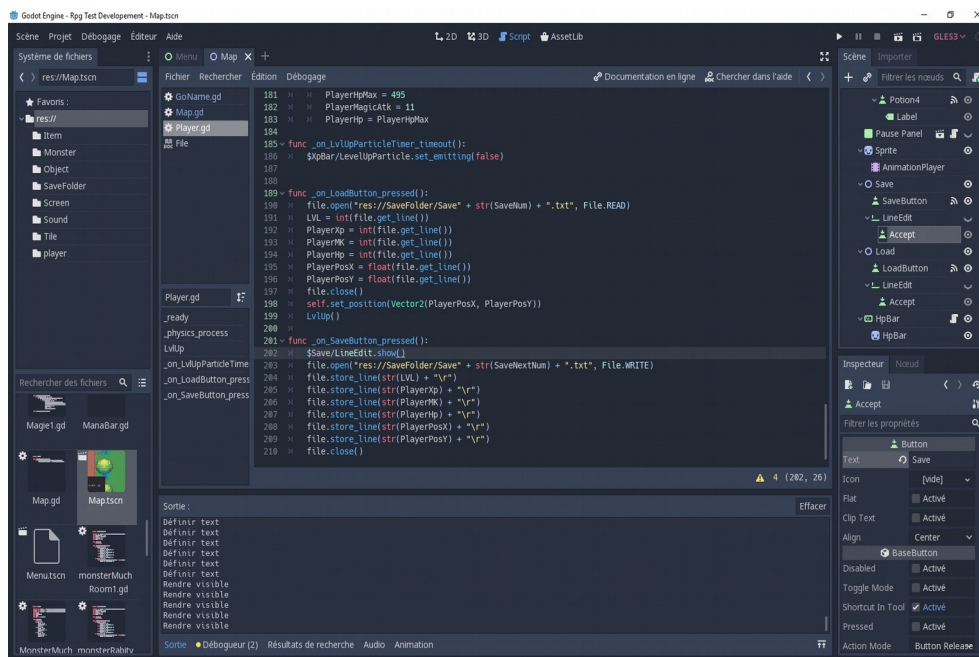
### i. Description du projet

Le personnage principal est déplaçable avec les flèches du clavier, il peut tirer sur des ennemis. Il y a un début d'interface graphique. Les ennemis ont leurs propres déplacements... Il a écrit cela seul. Je l'ai juste un peu aidé sur la syntaxe de certaines instructions.

- **Son jeu dans le programme :**



- **Une partie de son code dans le programme :**



**Possibilité de télécharger le jeu ici :**

Page sur internet regroupant différents liens / [nom de l'étudiant] : [lien supprimé]

## ii. Compétences

j'en déduis les compétences acquises (en particulier) :

- Capacité à utiliser un logiciel professionnel (donc très complexe) ! [nom de l'étudiant] a déjà une bonne compréhension de l'interface (ce qui n'est pas rien dans ce type de logiciel !).
- Les mêmes qu'avec Scratch mais avec une mise en œuvre plus complexe et plus riche.

## III. Déroulement du stage

Voir le récapitulatif des compétences dans le tableau en annexe.

### A. Jour 1

Présentation de Processing, prise en main, tests et vérification des compétences de base par l'écriture de tous petits programmes

Compétences :

- Programmation en Séquence et la console
- Variables

### B. Jour 2

Je lui montre le résultat à obtenir : l'affichage d'un rectangle au centre de l'écran se remplissant de façon aléatoire par des points. J'appelle ce programme « pointillisme dans un rectangle ».

**Compétences acquises** : système de coordonnées, utilisation de variables dans le code pour ne pas avoir à réécrire directement à chaque fois que l'on change quelque chose dans le programme.

**Difficulté** : baisse d'enthousiasme constatée voire ennui. Conclusion : l'approche pédagogique « classique » : montrer ce que l'on va obtenir et lui demander de le réaliser (en l'accompagnant pour lui donner au fur et à mesure les informations dont il a besoin) n'est pas efficace. Il n'y a plus l'enthousiasme constaté avec ses projets personnels d'où réflexion de ma part pour un changement de l'approche prévue.

**Notes de [nom de l'étudiant]**: « à la fin de la journée mon maître de stage m'a lancé le défi de réaliser le pointillisme dans un rectangle. A ce moment je ne comprenais pas encore complètement comment fonctionnait Processing et suite aux explications d'Eric et de recherches j'y suis arrivé.

Explications techniques :

J'ai créé plusieurs variables float randiX, randiY, hauteurrect et largeurrect\*.

On va créer une fonction setup suivie d'un rafraîchissement d'image puis on va changer la taille de l'écran, nous allons mettre rectMode en center ( rectMode(CENTER) )pour que le rectMode soit centré sur l'axe X et Y (hauteurrect, largeurrect) on va définir la couleur des objets que nous allons créer, on va remplir le carré de la couleur noire ( un carré blanc avec un intérieur noir) , définir hauteurrect et largeurrect à 360 px ( pixel) et on va dire que la position de rectangle est égal à x (la largeur de l'écran/2) et y (la hauteur de l'écran/2) et maintenant nous allons définir la Hauteur et Largeur du rectangle Largeur=largeurrect ; Hauteur= hauteurrect



Maintenant il va falloir faire apparaître des points aléatoirement sur l'écran c'est là qu'interviennent randiX et randiY. randiX est = random(largeurrect) randiY est = random(hauteurrect) maintenant qu'ils ont été définis nous allons pouvoir faire apparaître les points aléatoirement dans le carré.

```
point(randiX+width/2-largeurrect/2,randiY+height/2-hauteurrect/2)
```

Ainsi les point apparaissent aléatoirement à l'infini dans le carré.

\*rectangle »

## C. Jour 3

Je commence la séance en lui expliquant le changement d'approche du cours par rapport à ce que j'ai constaté le jour précédent. Je lui demande de décider lui-même d'un projet simple de jeu vidéo. Il choisit de faire un jeu avec un vaisseau spatial qui tire sur des ennemis.

La séance se déroule dans une tout autre ambiance : retour de l'enthousiasme et de la motivation. ([nom de l'étudiant] a besoin d'être acteur, investi pour réussir à rentrer dans une activité.)

- **Programme 3-1** : Affichage d'un carré blanc en bas de l'écran qui figure l'emplacement du vaisseau spatial du joueur à des coordonnées données (utilisation de variables).
- **Programme 3-2** : Récupération de l'information dans le programme lors de l'appui sur des touches du clavier

Le programme écrit dans la console qu'il a détecté l'appui sur les touches « flèche droite » ou « flèche gauche »

**Compétences acquises** : prise en compte du clavier, utilisation de la console , notion de boucle infinie dans les jeux vidéos

- **Programme 3-3** : Déplacement du vaisseau en fonction de l'appui sur les flèches droites ou gauches

Le vaisseau (un simple carré pour l'instant) se déplace en fonction de l'appui sur les flèches droite et gauche

**Compétences acquises**: utilisation de variables booléenne, de variables de type entier (pour la position du vaisseau), d'instructions de prise de décision du type « si...alors...sinon »

- **Programme 3-4** : Nécessité de commenter le code

Des explications sont ajoutées en clair dans le code afin mieux se repérer régulièrement par les programmeurs. J'avais l'intention d'aborder ce point mais je n'ai pas eu à le faire. [nom de l'étudiant] a ajouté, de lui-même, des annotations.

**Compétence acquise** : Comprendre la nécessité de commenter son code

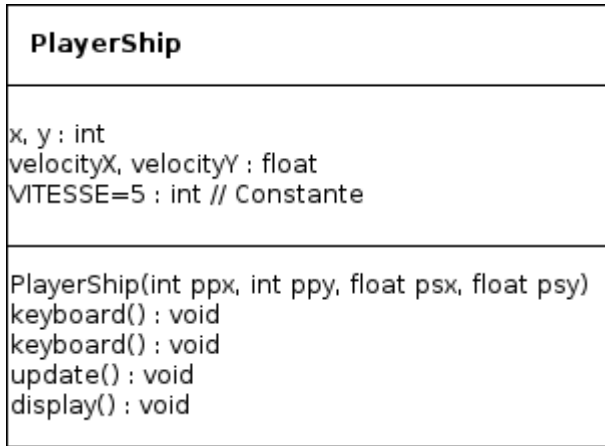
## D. Jour 4

- **Programme 4-1** : Vaisseau avec une classe

Il s'agit d'obtenir exactement la même chose que dans le programme 3-3 mais en déplaçant le code du vaisseau dans une « classe » à part. Le vaisseau spatial devient un objet à part pouvant être utilisé éventuellement plusieurs fois.

**Première introduction de la notation UML** (des diagrammes de classes) pour la représentation graphique de l'objet PlayerShip (le vaisseau spatial du jeu). Utilisation du programme : Violet UML Editor

**Diagramme de classes (UML) :**

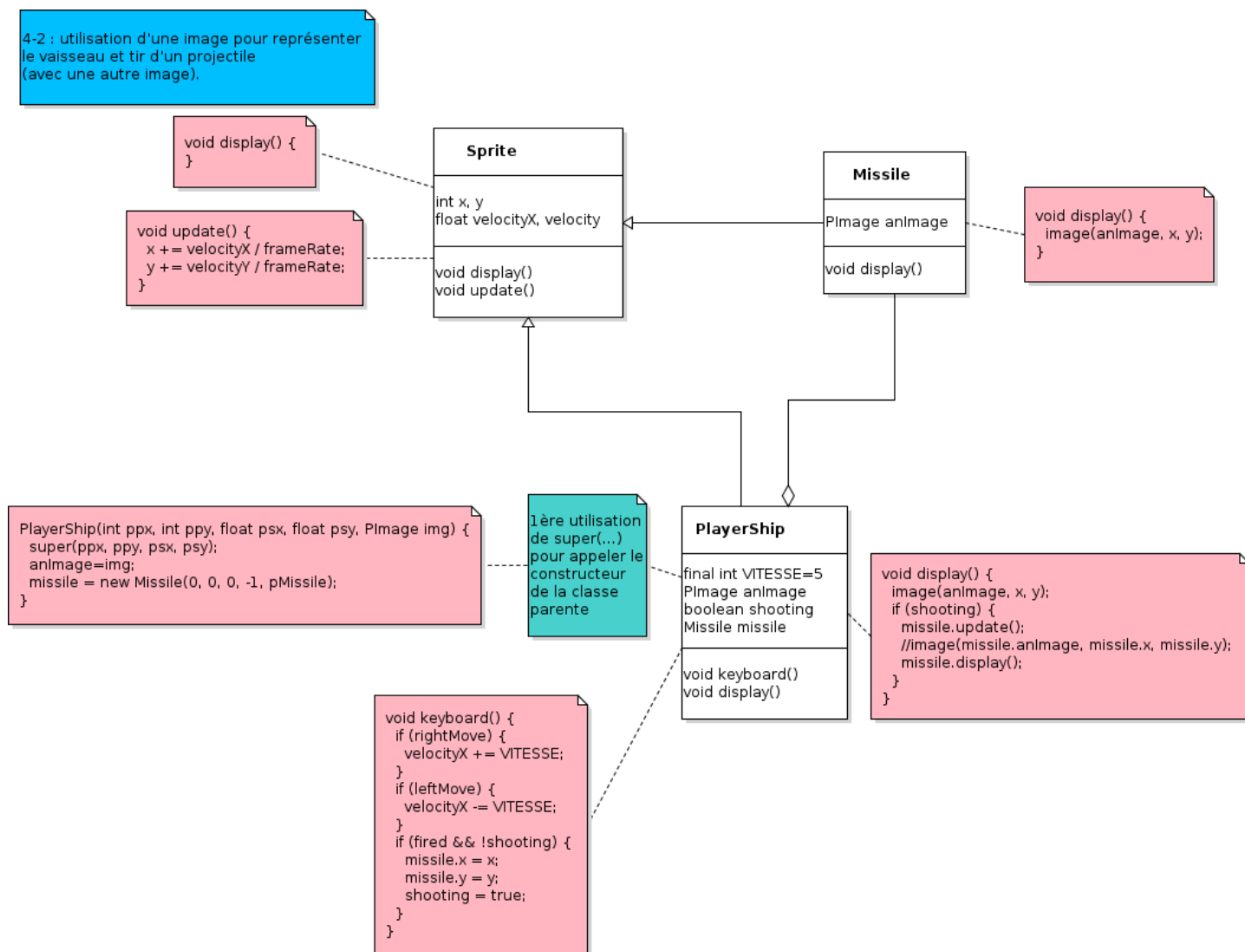


**Compétence à confirmer** : Création d'un objet (au sens de la « programmation objet ») à partir d'une classe et sa représentation graphique en UML.

La « programmation objet » est un exemple de règles et d'outils permettant d'obtenir des « parties de codes » indépendantes les unes des autres et réutilisables dans un programme plus vaste. On peut comparer cela à la création de « briques de code » qui peuvent s'imbriquer les unes dans les autres. c'est un concept fondamental utilisé dans la plupart des langages de programmation.

- **Programme 4-2** : utilisation d'une image pour représenter le vaisseau et tir d'un projectile (avec une autre image).. Concept d'héritage de classe et de collections abordés (et leur représentation dans le graphique)

## Diagramme de classes (UML) :



**Compétences en cours d'acquisition** : héritage de classe et collections (et leur représentation dans le graphique)

Utilisation de la notion « d'héritage » de classes : l'objet du vaisseau et du missile partagent un certain nombre de fonctions qui sont créés dans la classe parente Sprite (écrite donc, une seule fois).

Graphisme UML (cf ci-dessus) pour la représentation de l'organisation du programme. Pour aider à sa compréhension, le code est ajouté au graphisme.

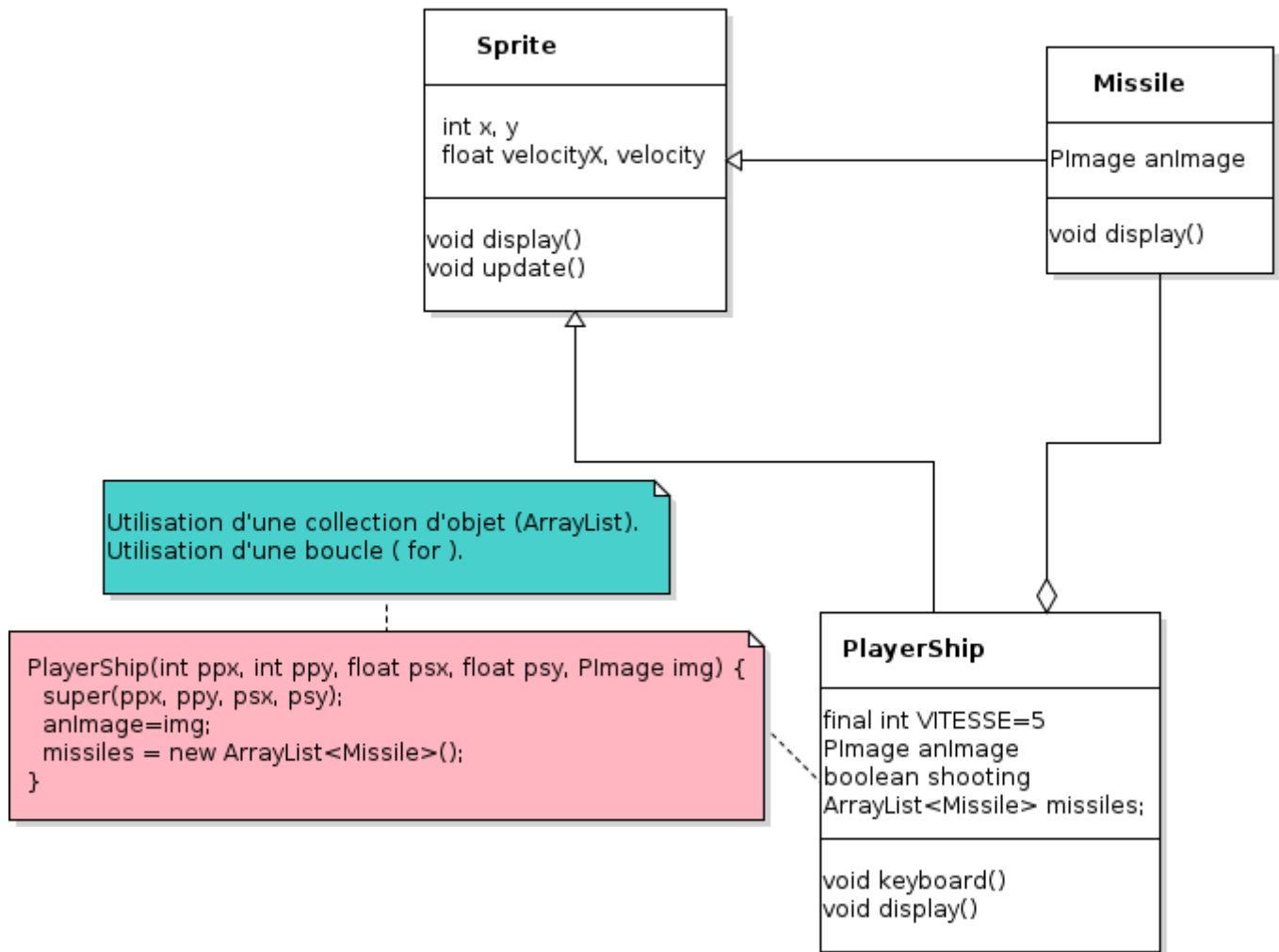
**N.B.** Ces compétences sont abordées avec [nom de l'étudiant]. Pour les confirmer, il faudra voir s'il est capable de les utiliser seul, par la suite, pour créer quelque chose d'autre. Ce sont des concepts abstraits qui ne peuvent être réellement comprise qu'avec un certain temps de pratique individuelle.

- **Programme 4-3** : Tir de plusieurs missiles par le vaisseau

Utilisation de la notion de « collection » qui permet d'enregistrer plusieurs objets dans une liste pour créer et utiliser plusieurs missiles pendant le déroulement du jeu.

**Compétences acquises** : Comprendre ce qu'est une collection (ici de type « ArrayList »), comprendre le principe d'une boucle « For »

#### 4-3 : Tir de plusieurs missiles par le vaisseau



## E. Jour 5

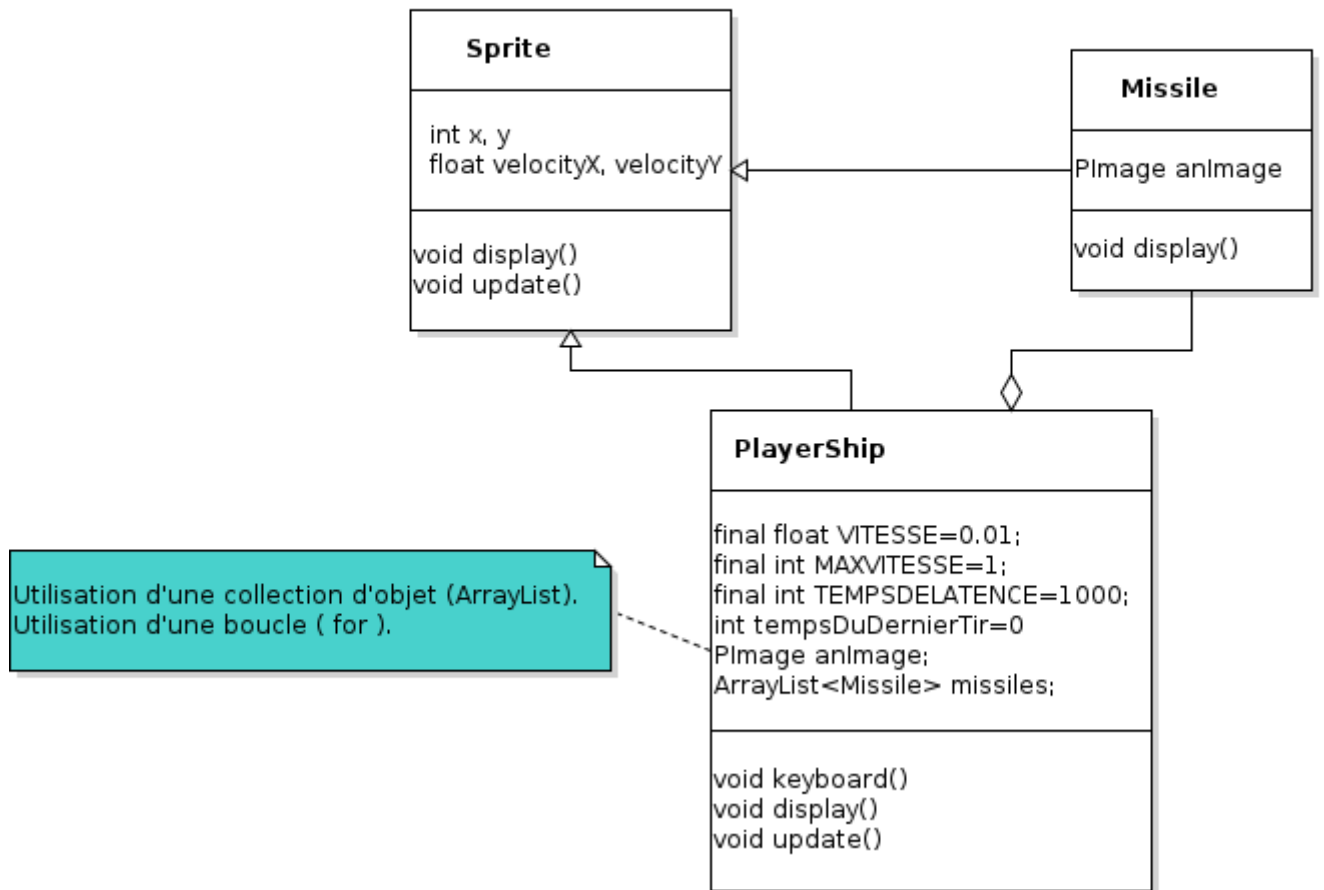
- **Programme 5-1** : déplacement du vaisseau dans les 4 directions avec les flèches du clavier  
Il s'agit d'une petite modification du programme pour commencer.

- **Programme 5.2** : Introduction d'un temps de latence entre le tir de 2 missiles  
Dans le programme précédent, le tir de missiles par le vaisseau a lieu en permanence. De nouveaux missiles sont donc créés des dizaines de fois par secondes (selon la vitesse d'affichage de l'ordinateur). Dans ce programme un temps minimum entre la création de 2 missiles est introduit.

Un objet représentant un astéroïde est ajouté au début du jeu d'une façon aléatoire en haut de l'écran grâce à la même classe d'objets utilisée pour le vaisseau du joueur.

(Pour l'instant, la collision entre l'astéroïde et les missiles n'est pas traitée).

**Pas de nouvelle compétence : confirmation des compétences déjà acquises**



## F. « Jour 6 »

Après le stage, [nom de l'étudiant] a complété son programme avec mon aide. Je lui ai fourni un code lui permettant de détecter des collisions pour faire disparaître les astéroïdes touchés par un missile. Il a aussi rajouté une image en arrière plan.

Il est possible de télécharger le programme pour le tester sur la page internet :

[lien supprimé]

## IV. Annexes

### A. Compétences

	<i>Compétences</i>	<i>Acquisition</i>	<i>Notes</i>	<i>Programme en Processing</i>
Compétence de base	Séquences d'instructions	Acquis	Les instructions sont effectuées dans leur ordre d'écriture	Jour 1
Compétence de base	Utilisation de la console	Acquis	Affichage de texte dans une fenêtre simple (utilisée pour trouver des erreurs dans un programme ou pour des tests)	Jour 1 Programme 3-2
Compétence de base	Variables : booléennes, entiers, flottantes...	Acquis	Pour enregistrer des informations (comme les positions d'une image à l'écran)	Jour 1
Compétence de base	système de coordonnées	Acquis	Similaire à la même notion en mathématique	Jour 2
Compétence de base	prise en compte du clavier	Acquis		Programme 3-2
Compétence de base	boucle infinie dans les jeux vidéos	Acquis	Pour répéter pendant un jeu les mêmes instructions (le déplacement des images par exemple)	Programme 3-2
Compétence de base	Prise de décision : « si...alors...sinon »	Acquis		Programme 3-3
Compétence de base	Nécessité des commentaires	Acquis		Programme 3-4
Compétence de base	Collections	Acquis sur le principe, à confirmer dans la maîtrise du code	Variables particulières pour enregistrer une liste d'objet (les différents missiles à l'écran par exemple)	Programme 4-2 & 4-3
Compétence de base	Boucle « For »	En cours d'acquisition	Pour parcourir une collection (répéter les mêmes instructions pour une série d'objet se trouvant dans une liste)	Programme 4-3
Compétence de niveau intermédiaire	Classe (dont sa représentation en UML)	En cours d'acquisition	Système permettant d'écrire une seule fois des instructions pour une série d'objets identiques (les astéroïdes par exemple)	Programme 4-1
Compétence de niveau intermédiaire	Héritage de classe	En cours d'acquisition	Permet d'avoir une partie commune de code pour des objets différents.	Programme 4-2

## B. Le code des programmes

### i. Jour 2

#### jour2\_pointillisme\_dans\_rectangle

```
float randiX;
float randiY;

float hauteurRect;
float largeurRect;

void setup() {
  background(0);
  size(500, 500);
  rectMode(CENTER);
  stroke(255);
  fill(0);
  hauteurRect = 360;
  largeurRect = 360;
  rect(width/2, height/2, largeurRect, hauteurRect);
}

void draw() {
  randiX = random(largeurRect);
  randiY = random(hauteurRect);
  point(randiX+width/2-largeurRect/2,randiY+height/2-
hauteurRect/2);
}
```

### ii. Jour 3

#### a) 3-2

```
//fonctions de base
boolean leftMove,rightMove;

void setup() {
  size(1080,720);
  frameRate(1000);
```

```
}
void draw() {
  background(10);
  rectMode(CENTER);
  rect(width/2,600,75,75);
  println("leftMove = " + leftMove + " rightMove = " +
rightMove);
  text("Frame rate: " + int(frameRate), 10, 20);
}
void keyPressed(){
  if(keyCode == LEFT){
    leftMove = true;
  }
  if(keyCode == RIGHT){
    rightMove = true;
  }
}
void keyReleased(){
  if(keyCode == LEFT){
    leftMove = false;
  }
  if(keyCode == RIGHT){
    rightMove = false;
  }
}
```

#### b) 3-3

```
//Déplacement du vaisseau
boolean leftMove, rightMove;
int playerPosX, playerPosY;

void setup() {
  size(1080, 720);
  frameRate(1000);
  playerPosX = width/2;
```

```

    playerPosY = 600;
}
void draw() {
    background(10);
    rectMode(CENTER);
    rect(playerPosX, playerPosY, 75, 75);

    println("leftMove = " + leftMove + " rightMove = " +
rightMove);
    text("Frame rate: " + int(frameRate), 10, 20);
    if (leftMove) {
        playerPosX -= 1;
    }
    if (rightMove) {
        playerPosX += 1;
    }
}
void keyPressed() {
    if (keyCode == LEFT) {
        leftMove = true;
    }
    if (keyCode == RIGHT) {
        rightMove = true;
    }
}
void keyReleased() {
    if (keyCode == LEFT) {
        leftMove = false;
    }
    if (keyCode == RIGHT) {
        rightMove = false;
    }
}

```

### c) 3-4

// Ajouts de commentaires

boolean leftMove, rightMove;

int playerPosX, playerPosY, Delta, preMillis;

```

float playerSpeed;
//Setup le code de base
void setup() {
    size(1080, 720);
    frameRate(1000);
    playerPosX = width/2;
    playerPosY = 600;
    playerSpeed = 0.60;
}
//Répéter ce code frameRate fois
void draw() {
    Delta = millis() - preMillis;
    preMillis = millis();
    background(10);
    rectMode(CENTER);
    rect(playerPosX, playerPosY, 75, 75);

    println("leftMove = " + leftMove + " rightMove = " +
rightMove);
    text("Frame rate: " + frameRate, 10, 20);
    text("Delta: " + Delta, 10, 40);
    if (leftMove) {
        playerPosX -= playerSpeed * Delta;
    }
    if (rightMove) {
        playerPosX += playerSpeed * Delta;
    }
}
//Si la touche keyCode LEFT est pressé leftMove = true, Si la
touche keyCode RIGHT...
void keyPressed() {
    if (keyCode == LEFT) {
        leftMove = true;
    }
    if (keyCode == RIGHT) {
        rightMove = true;
    }
}
}

```



```
//Si la touche keyCode LEFT est laché leftMove = false, Si la
touche keyCode RIGHT...
```

```
void keyReleased() {
  if (keyCode == LEFT) {
    leftMove = false;
  }
  if (keyCode == RIGHT) {
    rightMove = false;
  }
}
```

### iii. Jour 4

#### a) 4-2 : tir d'un seul projectile

- **Programme Principal**

```
PlayerShip player;
//Missile bullet;
PImage pShip, pMissile;
//Fonctions de base
boolean leftMove, rightMove, fired;

//Setup le code de base
void setup() {
  size(1080, 720);
  frameRate(1000);
  pShip = loadImage("PlayerShip.png");
  pMissile = loadImage("PlayerShoot.png");
  player = new PlayerShip(width/2, 600 , 0, 0, pShip);
  //bullet = new Missile(width/2, 600 , 0, 0, pMissile);
}

//Répéter ce code frameRate fois
void draw() {
  background(10);
  //println("leftMove = " + leftMove + " rightMove = " +
rightMove);
  text("Frame rate: " + frameRate, 10, 20);
  player.display();
```

```
player.update();
player.keyboard();
//bullet.display();
}
```

```
//Si la touche keyCode LEFT est pressé leftMove = true, Si la
touche keyCode RIGHT...
```

```
void keyPressed() {
  if (keyCode == LEFT) {
    leftMove = true;
  }
  if (keyCode == RIGHT) {
    rightMove = true;
  }
  if (key == ' ') {
    fired = true;
  }
}
```

```
//Si la touche keyCode LEFT est laché leftMove = false, Si la
touche keyCode RIGHT...
```

```
void keyReleased() {
  if (keyCode == LEFT) {
    leftMove = false;
  }
  if (keyCode == RIGHT) {
    rightMove = false;
  }
  if (key == ' ') {
    fired = false;
  }
}
```

- Les définitions des classes

```
class Sprite {
  int x, y;
  float velocityX, velocityY;

  Sprite(int ppx, int ppy, float psx, float psy) {
    x = ppx;
```

```

y = ppy;
velocityX = psx;
velocityY = psy;
}

void display() {
}

void update() {
    x += velocityX / frameRate;
    y += velocityY / frameRate;
}
}

class Missile extends Sprite {
    PImage anImage;

    Missile(int ppX, int ppy, float psx, float psy, PImage bullet)
    {
        super(ppX, ppy, psx, psy);
        anImage = bullet;
    }

    void display() {
        image(anImage, x, y);
    }
}

class PlayerShip extends Sprite {
    final int VITESSE=5;
    PImage anImage;
    boolean shooting = false;
    Missile missile;

    PlayerShip(int ppX, int ppy, float psx, float psy, PImage
img) {

```

```

super(ppX, ppy, psx, psy);
anImage=img;
missile = new Missile(0, 0, 0, -1, pMissile);
}

void keyboard() {
    if (rightMove) {
        velocityX += VITESSE;
    }
    if (leftMove) {
        velocityX -= VITESSE;
    }
    if (fired && !shooting) {
        missile.x = x;
        missile.y = y;
        shooting = true;
    }
}

void display() {
    image(anImage, x, y);
    if (shooting) {
        missile.update();
        //image(missile.anImage, missile.x, missile.y);
        missile.display();
    }
}
}

```

### **b) 4-3 Tir de plusieurs missiles**

```

PlayerShip player;
//Missile bullet;
PImage pShip, pMissile;
//Fonctions de base
boolean leftMove, rightMove, fired;

//Setup le code de base

```

```

void setup() {
  size(1080, 720);
  frameRate(1000);
  pShip = loadImage("PlayerShip.png");
  pMissile = loadImage("PlayerShoot.png");
  player = new PlayerShip(width/2, 600 , 0, 0, pShip);
  //bullet = new Missile(width/2, 600 , 0, 0, pMissile);
}
//Répéter ce code frameRate fois
void draw() {
  background(10);
  //println("leftMove = " + leftMove + " rightMove = " +
rightMove);
  text("Frame rate: " + frameRate, 10, 20);
  player.display();
  player.update();
  player.keyboard();
  //bullet.display();
}
//Si la touche keyCode LEFT est pressé leftMove = true, Si la
touche keyCode RIGHT...
void keyPressed() {
  if (keyCode == LEFT) {
    leftMove = true;
  }
  if (keyCode == RIGHT) {
    rightMove = true;
  }
  if (key == ' ') {
    fired = true;
  }
}
//Si la touche keyCode LEFT est laché leftMove = false, Si la
touche keyCode RIGHT...
void keyReleased() {
  if (keyCode == LEFT) {

```

```

    leftMove = false;
  }
  if (keyCode == RIGHT) {
    rightMove = false;
  }
  if (key == ' ') {
    fired = false;
  }
}
(Les définitions des classes sont omises)

```

#### iv. Jour 5 :

Ajout d'un temps d'attente entre le tir de 2 projectiles

```

PImage pShip, pMissile, imgAsteroid;
//Fonctions de base
boolean leftMove, rightMove, downMove, upMove, fired;

//Setup le code de base
void setup() {
  size(1080, 720);
  frameRate(1000);
  pShip = loadImage("PlayerShip.png");
  pMissile = loadImage("PlayerShoot.png");
  imgAsteroid = loadImage("Asteroid.png");
  player = new PlayerShip(width/2, 600, 0, 0, pShip);
  asteroid = new PlayerShip(random(0, width), 0, 0, 0.25,
imgAsteroid);
}
//Répéter ce code frameRate fois
void draw() {
  background(10);
  //println("leftMove = " + leftMove + " rightMove = " +
rightMove);
  text("Frame rate: " + frameRate, 10, 20);
  player.display();
  player.update();

```

```

player.keyboard();
asteroid.display();
asteroid.update();
}
//Si la touche keyCode LEFT est pressé leftMove = true, Si la
touche keyCode RIGHT...
void keyPressed() {
  if (keyCode == LEFT) {
    leftMove = true;
  }
  if (keyCode == RIGHT) {
    rightMove = true;
  }
  if (keyCode == DOWN) {
    downMove = true;
  }
  if (keyCode == UP) {
    upMove = true;
  }
  if (key == ' ') {
    fired = true;
  }
}
}
}
//Si la touche keyCode LEFT est laché leftMove = false, Si la
touche keyCode RIGHT...
void keyReleased() {
  if (keyCode == LEFT) {
    leftMove = false;
  }
  if (keyCode == RIGHT) {
    rightMove = false;
  }
  if (keyCode == DOWN) {
    downMove = false;
  }
  if (keyCode == UP) {
    upMove = false;
  }
  if (key == ' ') {
    fired = false;
  }
}
}
}

```