TP4 : Les collisions

**Table des matières :**

[1. La théorie des collisions 1](#_Toc507663250)

[2. Gestion des collisions 2](#_Toc507663251)

[a) Identification des AABB 2](#_Toc507663252)

[b) Recadrage de l’AABB 2](#_Toc507663253)

[c) Gestion des collisions 2](#_Toc507663254)

# La théorie des collisions

Il existe de nombreux algorithmes pour gérer les collisions, nous allons étudier le plus simple, l’**AABB** ***Axis Aligned Bounding Box***. Comme son nom l’indique, il s'agit d'un rectangle dont les axes sont alignés avec le repère orthonormé de notre écran.

Axe des x

Axe des y

**AABB du personnage**

**AABB du décor (couche O\_layer)**



Une AABB est définie par ses coordonnées x,y (en haut à gauche), sa largeur W et sa hauteur H :



**CoordX\_perso1, CoordY\_perso1**

**W = 32px**

**H = 36px**

L’algorithme de test de collision entre deux AABB est le suivant :

Si (ABBB\_decor.X==(AABB\_perso.X+AABB\_perso.W)) alors collision\_droite = vrai

Si ((ABBB\_decor.X+AABB\_decor.X==AABB\_perso.X) alors collision\_gauche = vrai

Si (ABBB\_decor.Y==(AABB\_perso.Y+AABB\_perso.H)) alors collision\_bas = vrai

Si ((ABBB\_decor.Y+AABB\_decor.H)==AABB\_perso.Y) alors collision\_haut = vrai

# Gestion des collisions

## Identification des AABB

**Travail n°1 :**

A partir du programme Processing exemple1, codez les deux fonctions pour dessiner les AABB :

* la fonction ***void dessine\_AABB\_perso()*** qui affiche le rectangle délimitant l’AABB du personnage.
* la fonction ***void dessine\_AABB\_decor()*** qui affiche le rectangle délimitant les AABB du decor (la couche O\_layer).



## Recadrage de l’AABB

Pour permettre au personnage de passer au plus près des obstacles, il faut recadrer l’AABB du personnage :

La nouvelle AABB doit être légèrement inférieur à la taille d’une tuile de 16px. On prend donc une AABB de 12px/12px pour avoir une marge de 2px de chaque côté.



CoordX\_AABB\_perso1 = CoordX\_perso1+((32-16)/2)+2

 = CoordX\_perso1+10

**CoordX\_AABB\_perso1,**

**CoordY\_AABB\_perso1**

**H\_AABB\_perso1 = 12px**

**W\_AABB\_perso1 = 12px**

CoordY\_AABB\_perso1 = CoordY\_perso1+(36-16)+2

 = CoordY\_perso1+22



**Travail n°2 :**

A partir du programme précédent, codez la fonction qui dessine la nouvelle AABB recadrée du personnage :

* la fonction ***void dessine\_AABB\_perso\_tile()*** qui affiche le rectangle délimitant l’AABB du personnage.

## Gestion des collisions

A présent, l’objectif est de localiser l’AABB du personnage par rapport à la grille de tuiles (20x20) qui composent la carte. Chaque tuile ayant un numéro allant de 0 à 400.

Il faut déterminer si la future position de l’AABB ne sera pas en collision avec un obstacle du décor (O\_layer) :

* Si on se déplace vers la droite, il faut tester le coin haut droit et le coin bas droit de l’AABB :



**num\_tuile\_AABB\_coin\_HG**

**num\_tuile\_AABB\_coin\_BG**

* Si on se déplace vers la gauche, il faut tester le coin haut gauche et le coin bas gauche de l’AABB :





**num\_tuile\_AABB\_coin\_HD**

**num\_tuile\_AABB\_coin\_BD**

* Si on se déplace vers le haut, il faut tester le coin bas gauche et le coin bas droit de l’AABB :





**num\_tuile\_AABB\_coin\_HD**

**num\_tuile\_AABB\_coin\_HG**

* Si on se déplace vers le bas, il faut tester le coin haut gauche et le coin haut droit de l’AABB :





**num\_tuile\_AABB\_coin\_BD**

**num\_tuile\_AABB\_coin\_BG**

**Travail n°3 :**

A partir du travail précédent, réalisez le programme final :

Codez la fonction qui permet de localiser l’AABB du personnage en déterminant sur quelles tuiles les 4 coins de l’AABB se situent.

* Une fonction ***int localisation(int coordX, int coordY)*** qui renvoie le numéro de la tuile par rapport aux coordonnées du coin de l’AABB passées en paramètres.

Codez la fonction de test des collisions qui sera vérifiée à chaque déplacement sur les deux coins de l’AABB concernés (**exemple :** si on veut se déplacer vers la droite, on *test num\_tuile\_AABB\_coin\_HG+1* et on teste *num\_tuile\_AABB\_coin\_HG+1*) :

* Une fonction ***boolean test\_collision(int num\_tuile)*** qui renvoie « vrai » s’il y a collision.