

Code du casse-brique

```
import ddf.minim.*; // Importation de la bibliothèque Minim, et utiliser du son
PFont police; // Déclaration du nom d'un font
Minim minim; // Permission d'avoir accès à la librairie
AudioPlayer NightVision; // Déclaration du nom de la chanson
int rectangleX=300; // Longueur du rectangle
int Ballex=365; // Position de la balle en abscisse
int Balley=535; // Position de la balle en ordonnée
int vy=-5; // Vitesse initiale de mouvement de la balle en ordonnée
int vx=5; // Vitesse initiale de mouvement de la balle en abscisse
int BriqueX=10; // Quantité de brique en abscisee
int BriqueY=5; // Quantité de brique en ordonnée
int result=0; // Le score du jeu
int mouvement; // Le déplacement de la plateforme
Brique brique[]=new Brique[BriqueX*BriqueY]; // Déclaration du tableau Brique, avec comme
dimensions BriqueX et BriqueY
String A = "SCORE: ";
String B = "WELCOME TO OUR ";
String C = "BRICK BREAKER \\ 'o' "; // Accessoires au jeu
PImage background; // Importation de l'image de fond de la partie jeu

void setup() {

    minim = new Minim(this); // A quoi correspond minim, this=ce programme
    NightVision = minim.loadFile("NightVision.mp3"); // Ici, pour la musique de fond
    NightVision.loop(999999999); // La musique de fond sera jouée en boucle
    background=loadImage("BackGroundGame.jpg"); // Image de fond de la partie jeu
```

```

size(1000, 600); // Dimensions de la fenêtre du jeu
}
void draw () {
background(0); // Background général
cadreEtResult(); // Permet d'afficher els accessoires au jeu
plateforme(); // La plaquette
balle (); // La balle
collisionBalle(); // collision plaquette/balle
bordure(); // La plaquette est bornée
constructeur(); // Placement des briques
NightVision.play();
for (int i=0; i<BriqueX*BriqueY; i++) { // Boucle permettant la création des briques
if (!brique[i].getbroken()) { // Si la brique est détruite
brique[i].collision(); // Application de la fonction collision
if (!brique[i].getbroken()) // Si la brique n'est pas détruite
fill(255, 255, 255); // La brique sera blanche
rect(brique[i].getBriqueX(), brique[i].getBriqueY(), 50, 20); // Création d'une brique, dont la
position sera donnée dans la classe brique
}
}
}

void cadreEtResult() {
police = loadFont("BankGothicBTMedium.vlw"); // Loading d'une police correspondant à la variable
"police"
smooth();
textFont(police); // Appliquer la police
textSize(15); // Taille police
text(B, 820, 50, -20); // Position et dimensions du texte B
textSize(15);
text(C, 820, 70, -20);
}

```

```

    textSize(15);
    text(A, 860, 120);
    textSize(15);
    text(result, 950, 120);
    stroke(0, 0, 255);
    strokeWeight(5);
    noFill();
    image(background, 0, 0);//Loading de l'image de fond du jeu
    rect(0, 0, 800, 800);// Création d'un cadre de jeu
}
void constructeur() {
    int numero=0;
    for (int i=0; i<BriqueX; i++) { //Délimitation en abscisse...
        for (int j=0; j<BriqueY; j++) { // ... et en ordonnée, tant que i et j ont respectivement inférieurs à
            BriqueX, BriqueY
            brique[numero]= new Brique(70+(65)*i, 20+(35)*j); // Création du tableau contenant un certain
            nombre de briques, décalage+(espace entre 2 briques)*ordonnée ou abscisse
            numero=numero+1;// Brique suivante
        }
    }
}

void mouseMoved() {
    mouvement=mouseX;
    if (rectangleX+50>mouvement) { // Si l'utilisateur presse gauche
        rectangleX=rectangleX-4;// la plateforme va vers la gauche
        if (rectangleX+100<=0) {
            rectangleX=rectangleX+0;
        }
    } else if (rectangleX+50<mouvement) { // Si l'utilisateur presse droite
        rectangleX=rectangleX+4;// la plateforme va vers la gauche
        if (rectangleX>=700) {

```

```
    rectangleX=rectangleX+0;
  }
}
void plateforme() {
  noStroke();
  smooth();
  fill(255, 255, 255);
  rect(rectangleX, 550, 100, 15); // Création de la plateforme, avec seulement l'abscisse qui varie
}
```

```
void bordure() { // Délimitation de la position de la plateforme dans la partie du jeu
  if (rectangleX<=0) {
    rectangleX=5 ;
  } else if (rectangleX>=700) {
    rectangleX=695;
  }
}
```

```
void balle () {
  noStroke();
  smooth();
  fill(255, 255, 255);
  ellipse(Ballex, Balley, 30, 30);
  if (keyCode==32) {
    Ballex=Ballex+vx;
    Balley=Balley+vy;
  }
  if (Ballex>=784) {
```

```
vx=-5;
} else if (Ballex<=10) {
vx=5;
}
```

```
if (Balley>=615) {
Ballex=365;
Balley=535;
rectangleX=300;
} else if (Balley<=15) {
vy=5;
}
}
```

```
void collisionBalle() {
if ((Ballex>rectangleX+50)&&(Ballex<rectangleX+110)) {
if ((Balley>535)&&(Balley<550)) {
vy=-5;
vx=5;
}
} else if ((Ballex>rectangleX)&&(Ballex<rectangleX+60)) {
if ((Balley>535)&&(Balley<550)) {
vy=-5;
vx=-5;
}
}
}
```

```
class Brique {
int x, y;
boolean broken = false;
```

```

Brique(int x0, int y0) {
    x=x0;
    y=y0;
}
boolean getbroken() {
    return broken;
}
int getBriqueX() {
    return x;
}
int getBriqueY() {
    return y;
}
void collision() {
    if(broken==true){
        x=5000;
        y=5000;
        result=result + 1;
    }
    if (Ballex+15>x && Ballex-15<x+50 && Balley+15>y && Balley-15<y+20) {
        broken=true;
        if (Ballex-15<x) {
            if (Balley-15<y) {
                if (vx<0) {
                    vy=-5;
                } else if (vy<0) {
                    vx=-5;
                } else
                if (((Ballex+15)-(x))>((Balley+15)-(y))) {
                    vy=-5;
                } else {

```

```

    vx=-5;
}
} else if (Balley+15> y+20) {
    if (vx<0) {
        vy=5;
    } else if (vy>0) {
        vx=-5;
    } else
    if (((Ballex+15)-(x))>((Balley-15)-(y+20))) {
        vy=5;
    } else {
        vx=-5;
    }
} else {
    vx=-5;
}
} else if (Ballex+15>x+50) {
    if (Balley-15<y) {
        if (vx>0) {
            vy=-5;
        } else if (vy<0) {
            vx=5;
        } else
        if (((Ballex-15)-(x+50))>((Balley+15)-(y))) {
            vy=-5;
        } else {
            vx=5;
        }
    } else if (Balley+15> y+20) {
        if (vx>0) {
            vy=5;

```

```
} else if (vy>0) {  
    vx=5;  
} else  
if (((Ballex-15)-(x+50))>((Balley-15)-(y+20))) {  
    vy=5;  
} else {  
    vx=5;  
}  
} else {  
    vx=5;  
}  
} else {  
    if (Balley-15<y) {  
        vy=-5;  
    } else if (Balley+15>y+20) {  
        vy=5;  
    }  
}  
}  
}  
}
```