

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
int etat=1; //état des briques (1= construction des briques)
Brique brique[]=new Brique[BriqueX*BriqueY]; // Déclaration du tableau Brique, avec comme
dimensions BriqueX et BriqueY
String A = "SCORE: "; // permettant une amorce du 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(); //cela permet de faire jouer la musique de fond
  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
  reset(); //Placement des briques
  victoire(); // en cas de victoire
  if (etat==1 || etat ==2) {
    for (int i=0; i<BriqueX*BriqueY; i++) { //Boucle permettant la création des briques
      fill(255, 255, 255); // La brique sera blanche
      brique[i].collision(); // Application de la fonction collision
      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 reset() {
  if (etat==1) { // Si les briques sont créées alors
    int numero=0;// Brique numéro 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 cotenant 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) { // Si la plateforme va vers la gauche du cadre elle reviendra a 5 d'abscice
    rectangleX=5 ;
  } else if (rectangleX>=700) { // De même pour la droite
    rectangleX=695;
  }
}

```

```

void balle () {
  noStroke();
  smooth();
  fill(255, 255, 255);
  ellipse(Ballex, Balley, 30, 30); // Coordonnées de la balles (position, dimensions)

```

```

  if (key==ENTER) { // Si la touche ENTER est activée, la balle bouge
    Ballex=Ballex+vx;
    Balley=Balley+vy;
  }

```

```

  if (Ballex>=784) { // Si la balle atteint le bord droit, la balle rebondit
    vx=-5;
  } else if (Ballex<=10) { // Si la balle atteint le bord gauche, la balle rebondit
    vx=5;
  }

```

```

  if (Balley>=615) { // Si la balle plonge, elle revient à son point de départ sur la plateforme
    Ballex=365;
    Balley=535;
    rectangleX=300; // La plateforme est aussi déplacée sur son point de départ
  } else if (Balley<=15) { // Si la balle atteint le bord haut, la balle rebondit
    vy=5;
  }
}

```

```

void collisionBalle() {
  if ((Ballex>rectangleX+50)&&(Ballex<rectangleX+110)) { // Si la balle touche la partie droite de
la plateforme, rebond de la balle
    if ((Balley>535)&&(Balley<550)) {
      vy=-5;
      vx=5;
    }
  } else if ((Ballex>rectangleX)&&(Ballex<rectangleX+60)) { // Si la balle touche la partie gauche
de la plateforme, rebond de la balle
    if ((Balley>535)&&(Balley<550)) {
      vy=-5;

```

```

    vx=-5;
}
}
}

```

```

class Brique {
int x, y;// Déclaration de variables pour utiliser les positions de la brique
Brique(int x0, int y0) {
    x=x0;
    y=y0;
}
int getBriqueX() {
    return x;
}
int getBriqueY() {
    return y;
}
}

```

```

void collision() {
    if (Ballex+15>x && Ballex-15<x+50 && Balley+15>y && Balley-15<y+20) { // Quand la balle
rencontre la plateforme
        if (Ballex-15<x) { // Rencontre sur bord gauche de la brique
            if (Balley-15<y) { // Rencontre sur bord haut de la brique
                if (vx<0) { // Rebond
                    vy=-5;
                    destruction();
                } else if (vy<0) {
                    vx=-5;
                    destruction();
                } else
                if (((Ballex+15)-(x))>((Balley+15)-(y))) { // Selon où arrive la balle, le rebond sera différent
                    vy=-5;
                    destruction();
                } else {
                    vx=-5;
                    destruction();
                }
            } else if (Balley+15> y+20) { // Rencontre sur bord bas de la brique
                if (vx<0) { // Rebond
                    vy=5;
                    destruction();
                } else if (vy>0) {
                    vx=-5;
                    destruction();
                } else
                if (((Ballex+15)-(x))>((Balley-15)-(y+20))) {
                    vy=5;
                    destruction();
                } else {
                    vx=-5;
                    destruction();
                }
            }
        }
    }
}

```

```

} else {
    vx=-5;
    destruction();
}
} else if (Ballex+15>x+50) { // Rencontre sur bord gauche de la brique
if (Balley-15<y) { // Rencontre sur bord gauche de la brique
    if (vx>0) { // Rebond
        vy=-5;
        etat=2;
        destruction();
    } else if (vy<0) {
        vx=5;
        destruction();
    } else
        if (((Ballex-15)-(x+50))>((Balley+15)-(y))) { // Selon où arrive la balle, le rebond sera
différent
            vy=-5;
            destruction();
        } else {
            vx=5;
            destruction();
        }
    }
} else if (Balley+15> y+20) { // Rencontre sur bord bas de la brique
if (vx>0) { // Rebond
    vy=5;
    etat=2;
    destruction();
} else if (vy>0) {
    vx=5;
    destruction();
} else
    if (((Ballex-15)-(x+50))>((Balley-15)-(y+20))) { // Selon où arrive la balle, le rebond sera
différent
        vy=5;
        destruction();
    } else {
        vx=5;
        destruction();
    }
} else {
    vx=5;
    destruction();
}
} else {
if (Balley-15<y) { // Rencontre sur bord haut de la brique
    vy=-5;
    destruction();
} else if (Balley+15> y+20) { // Rencontre sur bord bas de la brique
    vy=5;
    destruction();
}
}
}
}

```

```

    }
}
void destruction(){// La brique est déplacé a x=5000 et le score augmente de 1
    x= 5000;
    result = result +1;
}
}
void keyPressed() {
    if (key=='f') {// si on appuie sur 'f' on recommence,
        setup();
    }
    loop();
}
void victoire(){
    if(result == 50){//Si on détruit toutes les briques, alors le score se réinitialise, les briques se
reconstruisent, et la plateforme se remet au point de départ
        result=0;
        etat=1;
        Ballex=365;
        Balley=535;
        rectangleX=300;

        textSize(50);
        text("Win !", width/3, height/3+100);
        textSize(40);
        text("Press f to play again !", width/6, height/6+120);//Si partie terminé,f pour recommencer
        noLoop();
        result=0;
    }
    if (Balley>600){
        textSize(40);
        fill(0);
        text("Press f to play again !", width/6, height/6+120);//Si partie terminé,f pour recommencer
        noLoop();
        result=0;
        etat=1;
        Ballex=365;
        Balley=535;
        rectangleX=300;// La plateforme est aussi déplacée sur son point de départ
    }
}
}
}

```