

SMILEY :

```
PImage Dessin;

void setup() {
  size(300, 300);
  Dessin = loadImage("smiley.png");
}

void draw() {
  image(Dessin, 0, 0);
}
```

DEMOGET :

```
PImage Dessin;

void setup() {
  size(300, 300);
  Dessin = loadImage("smiley.png");
}

void draw() {
  image(Dessin, 0, 0);
}
```

ANIMV1 :

```
PImage Paysage;
PImage Balle;
int x, y, dx, dy;

void setup() {
  size (320, 200);
  Paysage = loadImage("paysage.jpg");
  Balle = loadImage("balle.png");
  image(Paysage, 0, 0);
  x = 0;
  y = 0;
  dx = 1;
  dy = 1;
}

void draw() {
  image(Balle, x, y);
  x = x + dx;
  y = y + dy;
}
```

ANIMV2 :

```
PImage Paysage;
PImage Balle;
PImage Fondballe;
int x, y, oldx, oldy, dx, dy;

void setup() {
  size (320, 200);
  Paysage = loadImage("paysage.jpg");
  Balle = loadImage("balle.png");
  image(Paysage, 0, 0);
  x = 0;
  y = 0;
  oldx = 0;
  oldy = 0;
  dx = 1;
  dy = 1;
  Fondballe = get(x, y, 20, 20);
}

void draw() {
  image(Fondballe, oldx, oldy);
  oldx = x;
  oldy = y;
  Fondballe = get(x, y, 20, 20);
  image(Balle, x, y);
  x = x + dx;
  y = y + dy;
}
```

ANIMV3 :

```
// Création des variables
PImage Paysage;
PImage Balle;
PImage Fondballe;
int x, y, oldx, oldy, dx, dy;

void setup() {
  // Définition de la fenêtre graphique
  size (320, 200);
  // Lecture des images
  Paysage = loadImage("paysage.jpg");
  Balle = loadImage("balle.png");
  // Affichage du décor
  image(Paysage,0,0);
  // Initialisation des variables
  // x et y correspondent aux coordonnées de la balle à un instant t
  x = 0;
  y = 0;
  // oldx et oldy sont les coordonnées précédentes de la balle
  oldx = 0;
  oldy = 0;
  // dx et dy sont les déplacement de la balle selon x et selon y
  dx = 1;
  dy = 1;
  // Fondballe est la portion de décor qui se trouve derrière la balle
  Fondballe = get(x, y, 20, 20);
}

void draw() {
  // On remplace le fond à l'ancien emplacement de la balle
  image(Fondballe, oldx, oldy);
  // Puis on enregistre les nouvelles coordonnées de la balle
  oldx = x;
  oldy = y;
  // On copie la partie du fond se trouvant derrière la balle
  Fondballe = get(x, y, 20, 20);
  // Puis on affiche la balle
  image(Balle, x, y);
  // On calcule ses nouvelles coordonnées : elle avance horizontalement de dx,
  // et elle avance verticalement de dy
  x = x + dx;
  y = y + dy;
  // Si la coordonnée x est hors de la fenêtre, alors on change de sens selon x
  // "width" contient la largeur de la fenêtre graphique
  if ((x >= width - 20) || (x <= 0)) {
    dx = -dx;
  }
  // Si la coordonnée y est hors de la fenêtre, alors on change de sens selon y
  // "height" contient la hauteur de la fenêtre graphique
  if ((y >= height - 20) || (y <= 0)) {
    dy = -dy;
  }
}
```



SQUASH :

```
// Squash v1.0
```

```
// Déclaration des variables
```

```
PImage balle, raquette, fondballe, fondraquette;
```

```
int x, y, dx, dy, oldx, oldy;
```

```
int xraquette, yraquette, oldxraquette, oldyraquette;
```

```
void setup() {
```

```
    // Initialisations
```

```
    oldx = 0;
```

```
    oldy = 0;
```

```
    x = 0;
```

```
    y = 0;
```

```
    dx = 2;
```

```
    dy = 2;
```

```
    oldxraquette = 300;
```

```
    oldyraquette = 450;
```

```
    xraquette = 300;
```

```
    yraquette = 450;
```

```
    balle = loadImage("balle2.png");
```

```
    raquette = loadImage("raquette.png");
```

```
    size(700,500);
```

```
    fondballe = get(x, y, 20, 20);
```

```
    fondraquette = get(xraquette, yraquette, 80, 20);
```

```
}
```

```
void draw() {
```

```
    // Efface les positions précédentes de la balle et de la raquette
```

```
    image(fondballe, oldx, oldy);
```

```
    image(fondraquette, oldxraquette, oldyraquette);
```

```
    // récupère le fond des nouvelles positions de la balle et de la raquette
```

```
    oldx = x;
```

```
    oldy = y;
```

```
    oldxraquette = xraquette;
```

```
    oldyraquette = yraquette;
```

```
    fondballe = get(x, y, 20, 20);
```

```
    fondraquette = get(xraquette, yraquette, 80, 20);
```

```
    // Affiche la balle et la raquette
```

```
    image(balle, x, y);
```

```
    image(raquette, xraquette, yraquette);
```

```
    // Déplacement de la balle
```

```
    x = x + dx;
```

```
    y = y + dy;
```

```
    // Rebond éventuel de la balle sur le haut du terrain
```

```
    if (y <= 0) {
```

```
        dy = -dy;
```

```
    }
```

```
    // Rebond éventuel de la balle sur la gauche ou la droite du terrain
```

```
    if ((x < 0) || (x > width - 20)) {
```

```
        dx = -dx;
```

```
    }
```

```
    // Déplacement de la raquette
```

```
    if (keyPressed) {
```

```
        if (key == CODED) {
```

```
            // On se déplace avec les touches flèche vers la gauche et flèche vers la droite
```

```
            if ((keyCode == LEFT) && (xraquette > 0)) {
```

```
                xraquette = xraquette - 2;
```

```
            }
```

```
            if ((keyCode == RIGHT) && (xraquette < width - 80)) {
```

```
                xraquette = xraquette + 2;
```

```
            }
```

```
        }
```

```
    }
```

```
    // La raquette touche la balle ?
```

```
    if ((y + 20 == yraquette) && (x >= xraquette) && (x + 20 <= xraquette + 80)) {
```

```
        // Si oui la balle rebondit
```

```
        dy = -dy;
```

```
    }
```

```
    // On perd la balle ?
```

```
    if (y > yraquette) {
```

```
        // On remet la balle en haut
```

```
        x = 0;
```

```
        y = 0;
```

```
    }
```

```
}
```

TEST DE PLUSIEURS TOUCHES SIMULTANEMENT :

```
boolean[] touches = new boolean[128];

void setup() {
    .....
}

void draw() {
    .....
}

void keyPressed(){
    touches[keyCode]=true;
}

void keyReleased(){
    touches[keyCode]=false;
}

boolean verifieClavier(char c){
    int cc = int(c);
    if(cc > 96 && cc < 123) {
        cc-=32;
    }
    return touches[cc];
}

boolean verifieClavier(int c){
    return touches[c];
}
```