

# Brick Shooter

Projet B1 Informatique

# Membre de l'Équipe

- LOUAN NERRIERE
  - 19 ANS
  - B1 INFORMATIQUE
- ALEXIS PHELIPPON
  - 21 ANS
  - B1 INFORMATIQUE

# Sommaire



Introduction



Gestion de  
Projet



Logiciels  
Utilisés



Organisation  
du Code



Démonstration



Évolution des  
Compétences



FAQ

# Introduction

---

## LES ATTENTES DES FONCTIONNALITÉS DU PROJET :



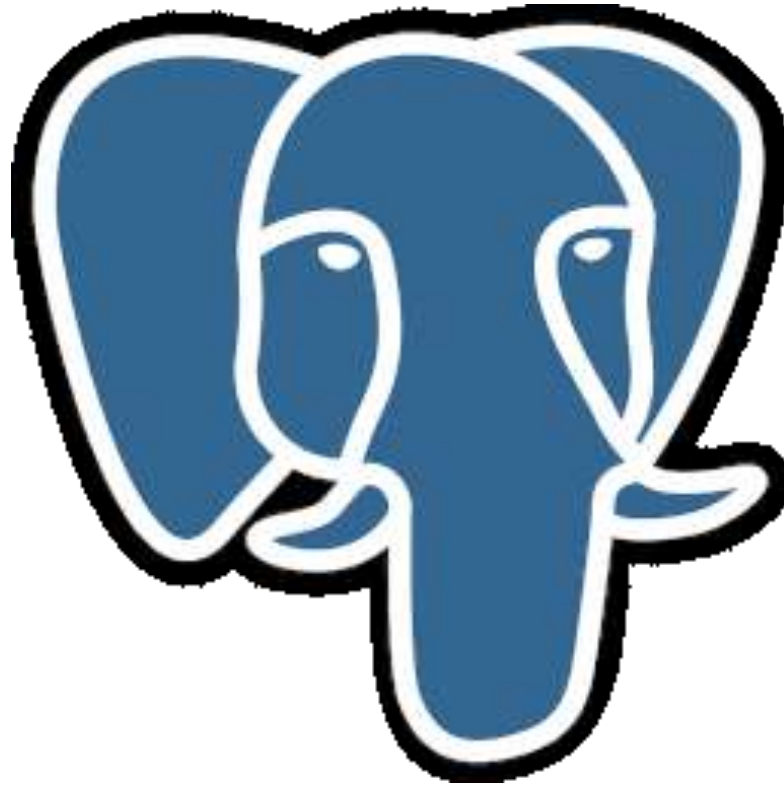
### Fonctionnalités

- cinq écrans : *Difficulté : 2*
  - menu principal avec accès aux autres écrans
  - écran de jeu avec possibilité de retour
  - écran de fin de partie
  - écran d'instructions avec possibilité de retour
  - écran d'équipement avec possibilité de retour
- l'écran d'instructions devra présenter à l'utilisateur toutes les données nécessaires à la bonne utilisation du logiciel *Difficulté : 1*
- modèle de données :
  - un joueur possédant : *Difficulté : 2*
    - des crédits
    - une vitesse de tir
    - une vitesse de déplacement
    - une vitesse de défilement
  - des types de bloc possédant : *Difficulté : 2*
    - un type de donnée du Joueur, peut être nul
    - une valeur, peut être positive ou négative ou nul
    - une probabilité d'apparition (il vaut mieux que le bloc n'affectant pas le joueur soit celui de plus forte probabilité)
- pour les types de bloc, avoir au minimum :

# Gestion de projet

- Prise en compte des compétences individuelles.
- Décision des outils informatiques à utiliser.
- Attribution des tâches pour chaque personne du groupe.
- Utilisation d'un logiciel de versionnage et de stockage.





Logiciels Utilisés



# Organisation du code

Création d'une aire de jeu en 2D avec l'élément CANVAS

```
1  //-----SÉLECTION DE L'ÉLÉMENT CANVAS-----//
2  const cvs = document.getElementById("breakout");
3  const ctx = cvs.getContext("2d");
4
5  //-----AJOUT DES BORDURES AU CANVAS-----//
6  cvs.style.border = "2px solid rgba(37, 204, 247,0.8)";
7
8  //-----DESSIN DES LIGNES DANS LE CANVAS-----//
9  ctx.lineWidth = 2;
```

//-----CRÉATION DES CLASSES-----//

```
/*class Brique {  
    constructor (largeur, longueur, durabilite, proba_spawn, effet, proba_effet, x_pos, y_pos){  
        this.largeur = largeur;  
        this.longueur = longueur;  
        this.durabilite = durabilite;  
        this.proba_spawn = proba_spawn;  
        this.effet = effet;  
        this.proba_effet = proba_effet;  
        this.x_pos = x_pos;  
        this.y_pos = y_pos;  
    }  
}*/
```

```
/*class Paddle {  
    constructor (credit, vitesse_tir, vitesse_deplacement, vitesse_defilement, x_pos, y_pos){  
        this.credit = credit;  
        this.vitesse_tir = vitesse_tir;  
        this.vitesse_deplacement = vitesse_deplacement;  
        this.vitesse_defilement = vitesse_defilement;  
        this.x_pos = x_pos;  
        this.y_pos = y_pos;  
    }  
}*/
```

```
class Shoot {  
    constructor(id, cooX, cooY) {  
        this.id = 0;  
        this.cooX = cooX;  
        this.cooY = cooY;  
    }  
}
```

## Création des classes



//

```
function update() {  
    movePaddle();  
    shootBrickCollision();  
    createShoot();  
    updateTirs();  
    gameOver();  
    levelUp();  
}
```

Fonction du jeu

# Ajout de fonctionnalité

```
//-----CHOISIR LES ÉLÉMENTS SONORES-----
const soundElement = document.getElementById("sound");

soundElement.addEventListener("click", audioManager);

function audioManager() {
  // CHANGE L'IMAGE SON ON/OFF
  let imgSrc = soundElement.getAttribute("src");
  let SOUND_IMG = imgSrc == "img/SOUND_ON.png" ? "img/SOUND_OFF.png" : "img/SOUND_ON.png";

  soundElement.setAttribute("src", SOUND_IMG);

  //-----ACTIVE ET DÉACTIVE LES SONS-----
  WALL_HIT.muted = WALL_HIT.muted ? false : true;
  PADDLE_HIT.muted = PADDLE_HIT.muted ? false : true;
  BRICK_HIT.muted = BRICK_HIT.muted ? false : true;
  WIN.muted = WIN.muted ? false : true;
  LIFE_LOST.muted = LIFE_LOST.muted ? false : true;
}

// CLIQUEZ SUR LE BOUTON "PLAY AGAIN"
restart.addEventListener("click", function () {
  location.reload(); // recharge la page
})

//-----MONTRE QUE VOUS AVEZ GAGNEZ-----
function showYouWin() {
  gameover.style.display = "block";
  youwon.style.display = "block";
}

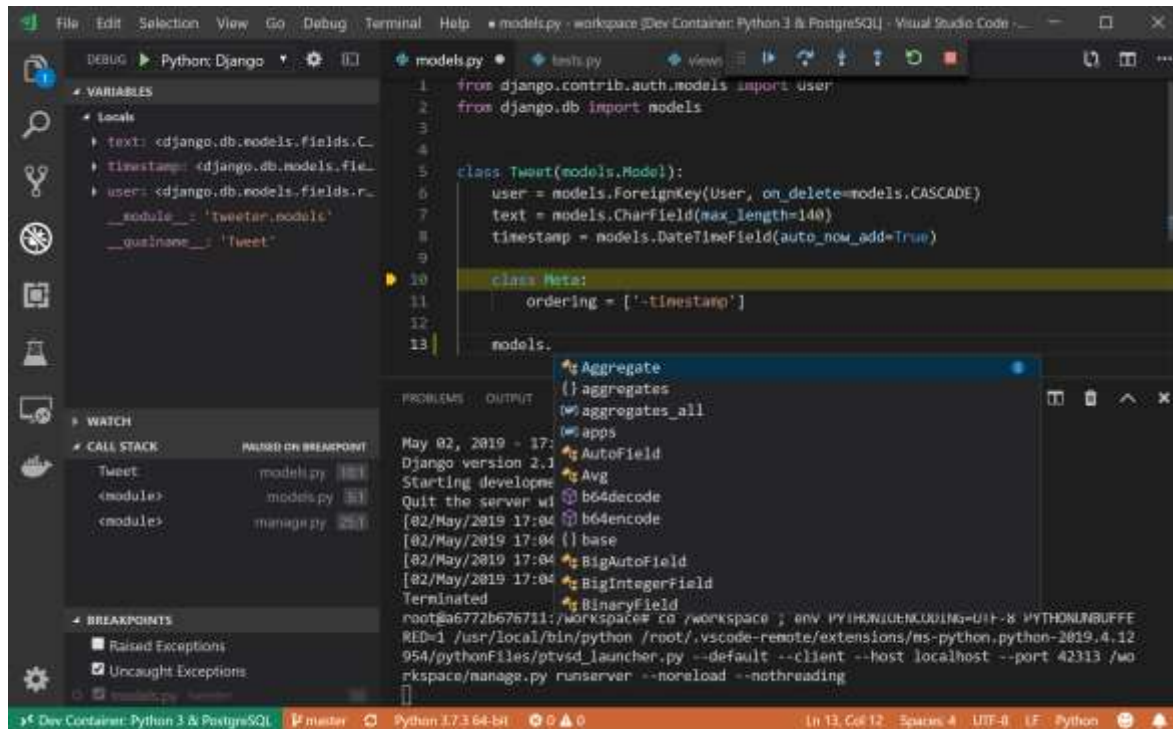
//-----MONTRE QUE VOUS AVEZ PERDU-----
function showYouLose() {
  gameover.style.display = "block";
  youlose.style.display = "block";
}
```

# Page HTML donnant accès au jeu

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8" />
5     <title>Casse-briques</title>
6     <link rel="stylesheet" href="css.css" />
7   </head>
8   <body>
9     <a href="../../Ball/index.html" class="bouton1">play ball</a>
10    <a href="../../Shoot/index.html" class="bouton2">play shoot</a>
11    <script></script>
12  </body>
13 </html>
14
```

```
1 body {
2   background-image: url("https://i.imgur.com/jXKzkEl.png");
3
4   margin: 0;
5   padding: center;
6   background-size: cover;
7   background-repeat: no-repeat;
8 }
9 a.bouton1 {
10  padding: 10px 50px;
11  background: #646262;
12  border-radius: 10px;
13  opacity: 0.4;
14  color: white;
15  position: absolute;
16  bottom: 90px;
17  right: 650px;
18  height: 26px;
19  width: 26px;
20 }
```

# Base de donnée



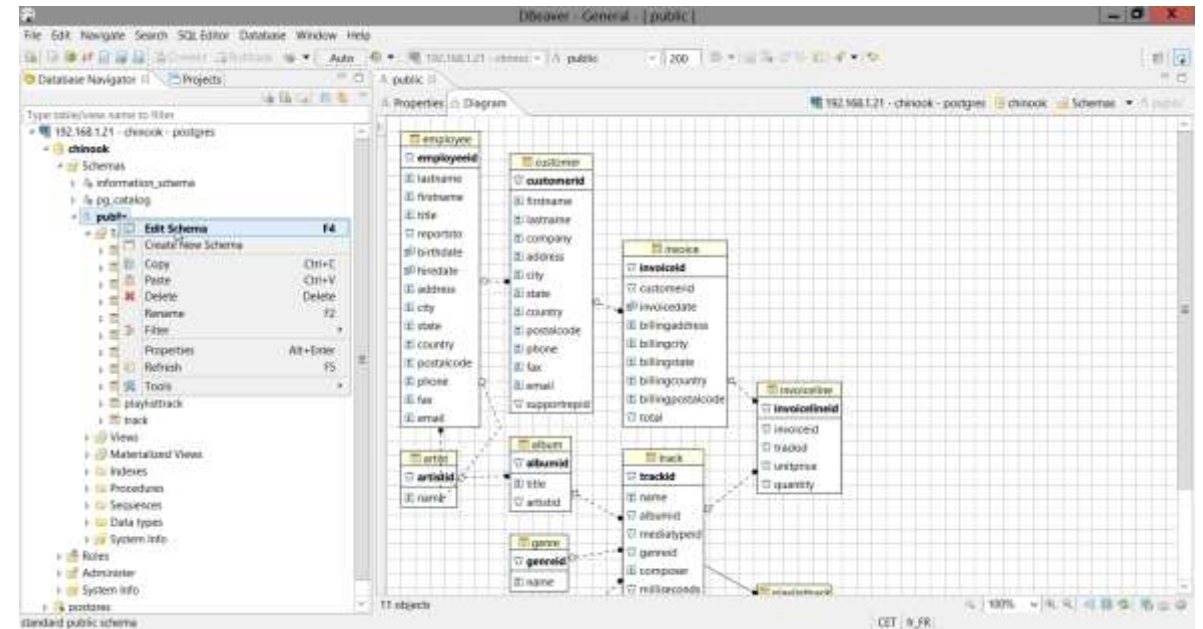
The screenshot shows the Visual Studio Code editor with a Django project. The main editor displays the `models.py` file, which defines a `Tweet` model. The model has a `user` field (a `ForeignKey` to the `User` model), a `text` field (a `CharField` with a maximum length of 140), and a `timestamp` field (a `DateTimeField` with `auto_now_add=True`). The `Meta` class for the model specifies an ordering of `['-timestamp']`.

```
1 from django.contrib.auth.models import User
2 from django.db import models
3
4
5 class Tweet(models.Model):
6     user = models.ForeignKey(User, on_delete=models.CASCADE)
7     text = models.CharField(max_length=140)
8     timestamp = models.DateTimeField(auto_now_add=True)
9
10 class Meta:
11     ordering = ['-timestamp']
12
13 models.
```

The left sidebar shows the `VARIABLES` pane with the following variables:

- `text`: `<django.db.models.fields.C...`
- `timestamp`: `<django.db.models.fie...`
- `user`: `<django.db.models.fields.r...`
- `__module__`: `'tweetar.models'`
- `__qualname__`: `'Tweet'`

The bottom pane shows the `PROBLEMS` and `OUTPUT` panes. The `PROBLEMS` pane shows a `RuntimeError` with the message `May 02, 2019 - 17:04: Django version 2.1.12 is not supported. The supported versions are 2.1.12 and 2.1.13.` The `OUTPUT` pane shows the command `python manage.py runserver --noreload --nothreading` and the output `root@6772b676711: /workspace# cd /workspace ; env PYTHONUNBUFFERED=1 /usr/local/bin/python /root/.vscode-remote/extensions/ms-python.python-2019.4.12/pythonFiles/ptvsd_launcher.py --default --client --host localhost --port 42313 /workspace/manage.py runserver --noreload --nothreading`.





DÉMONSTRATION





# Évolution des compétences

- Beaucoup d'auto-formation :
  - Open Class Room
  - MDN (Mozilla Developer Network)
  - W3School
  - Partage de l'expérience de nos camarades
- Logiciel :
  - JavaScript
  - CSS
  - HTML
  - Postgre SQL
  - GIT



# FAQ

Avez-vous des questions ?