

**Projet Java**

**Rapport 3: Java GUI**

**SNAKE GAME**

**Classe : 17VP**

**Notre groupe:**

**Nguyễn Thị Quỳnh Hoa – 1752017**

**Lưu Phạm Mạnh Hà – 1752001**

**Nguyễn Hoàng Anh – 1652002**



## Tableau de contenu

L'histoire de versions	3
Introduction	4
Analyse et conception	5
Implémentation	6
Exemples de données	7
Résultat	8
Plan	9
Références	10

---

**L'histoire de versions**

Le nombre ordinal	Le travailler	Le réalisateur	La date
1	Analyse et conception	1652002 - Nguyễn Hoàng Anh	25/05/2020
2	Implémentation	1652002 - Nguyễn Hoàng Anh	25/05/2020
3	Résultat	1752017 - Nguyễn Thị Quỳnh Hoa	26/05/2020
4	Démonstration	1652002 - Nguyễn Hoàng Anh	30/05/2020
5	Introduction	1752017 - Nguyễn Thị Quỳnh Hoa	30/05/2020
6			

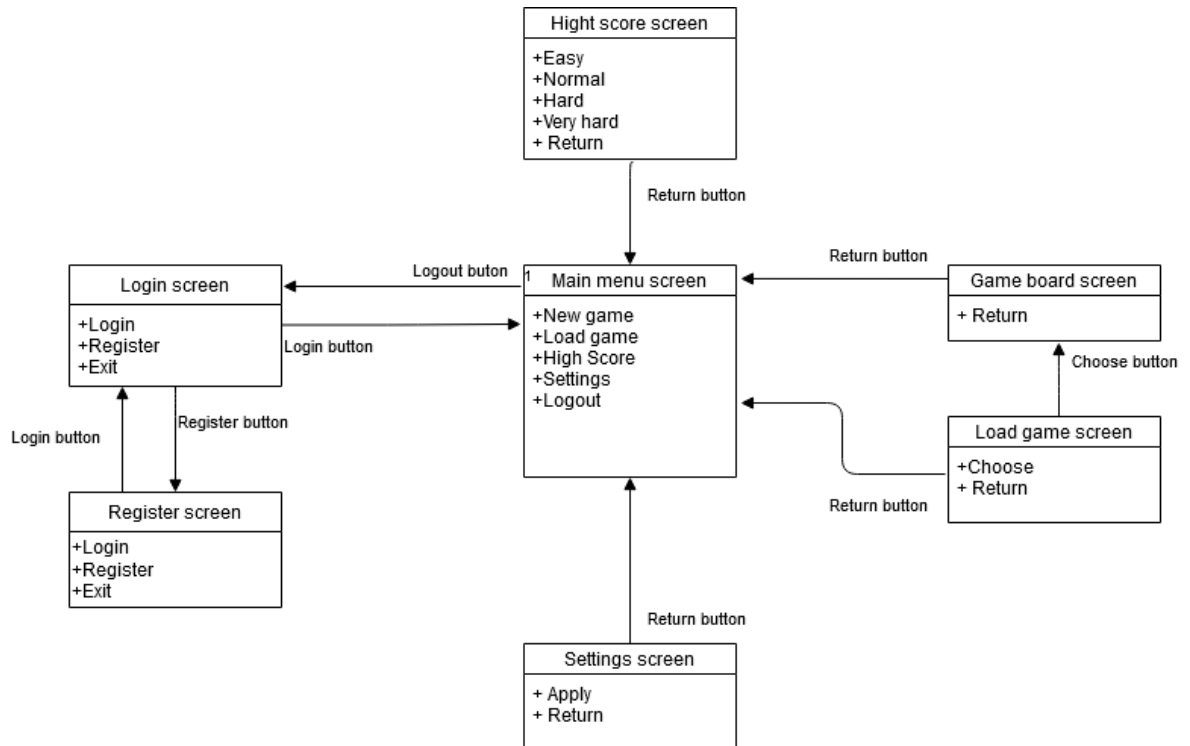
## Introduction

*[Vous présentez ici le but de votre rapport, expliquer comment vous désignez l'interface graphique, outils supplémentaires utilisés, les caractéristiques de votre application (une application simple, un simulateur ou un jeux graphique, etc.). A partir de ces caractéristiques, donnez des exigences pour l' interface graphique]*

Le but de ce rapport est de montrer la disposition des dessins et la navigation entre les pages de notre projet.

Nous avons conçu l'interface en utilisant la navigation pour gérer la navigation des pages étroitement liées. La conception évite les tâches excessives et trop complexes pour l'utilisateur lors de l'utilisation de l'interface. Minimisez les opérations requises par les utilisateurs (saisie de données, confirmation de demande, ..) Assurez la cohérence de la conception de l'interface. Le design le plus minimaliste possible.

## Analyse et conception



## Implémentation

*[Présentez ici comment vous transférer ce que vous avez désigné dans la partie précédente en utilisant Java]*

Utilisation ActionListener passer d'un écran à l'autre

```
login.btn_register.addActionListener(new ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        setTitle("Register Page");  
        changePanel(panel_register);  
    }  
});
```

```
login.btn_login.addActionListener(new ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        setTitle("Main Menu");  
        changePanel(panel_main_menu);  
    }  
});
```

```
register.btn_login.addActionListener(new ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        setTitle("Home Screen");  
        changePanel(panel_login);  
    }  
});
```

```
main_menu.btn_new_game.addActionListener(new ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        setTitle("Game Boar");  
        setBounds( x: 200, y: 0, width: width+160, height);  
        changePanel(panel_game_board);  
        sound.stop();  
        sound = setMusic("Sound\\\\\\game_sound.wav");  
        setVolume(sound, vol);  
    }  
});
```

```
main_menu.btn_load_game.addActionListener(new ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        setTitle("Load Game");
        changePanel(panel_load_game);
    }
});

main_menu.btn_high_score.addActionListener(new ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        setTitle("High Score");
        changePanel(panel_high_score);
    }
});

main_menu.btn_settings.addActionListener(new ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        setTitle("Settings");
        changePanel(panel_settings);
    }
});
```

```
main_menu.btn_settings.addActionListener(new ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        setTitle("Settings");
        changePanel(panel_settings);
    }
});

main_menu.btn_logout.addActionListener(new ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        setTitle("Home Screen");
        changePanel(panel_login);
    }
});
```

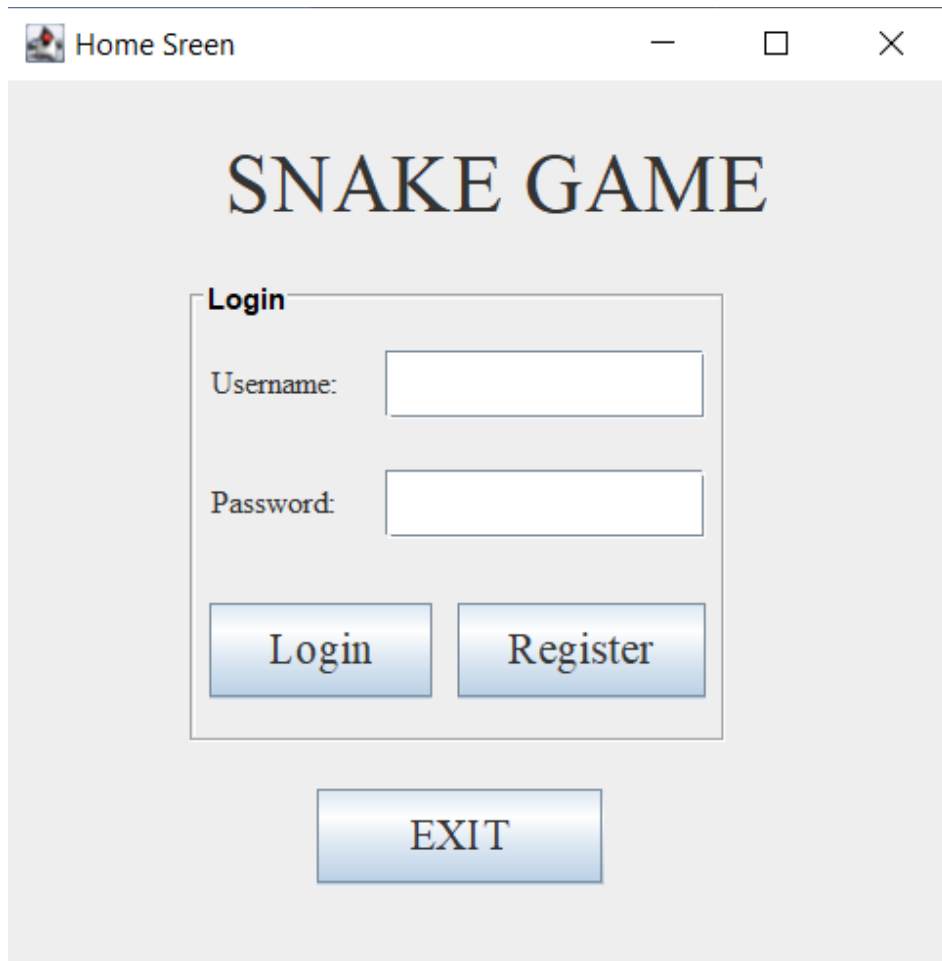
Utilisation Reszie ajustez la taille de l'écran



```
settings.Resize(width ,height);  
main_menu.Resize(width ,height);  
login.Resize(width,height);  
register.Resize(width, height);  
high_score.Resize(width, height);  
load_game.Resize(width, height);  
game_board.Resize(width, height);
```

## Démonstration

*[Donnez ici la capture des écrans de votre application et donnez les explication pour chaque écran capturé]*



Register bouton: Passer de l'écran de connexion à l'écran d'enregistrement

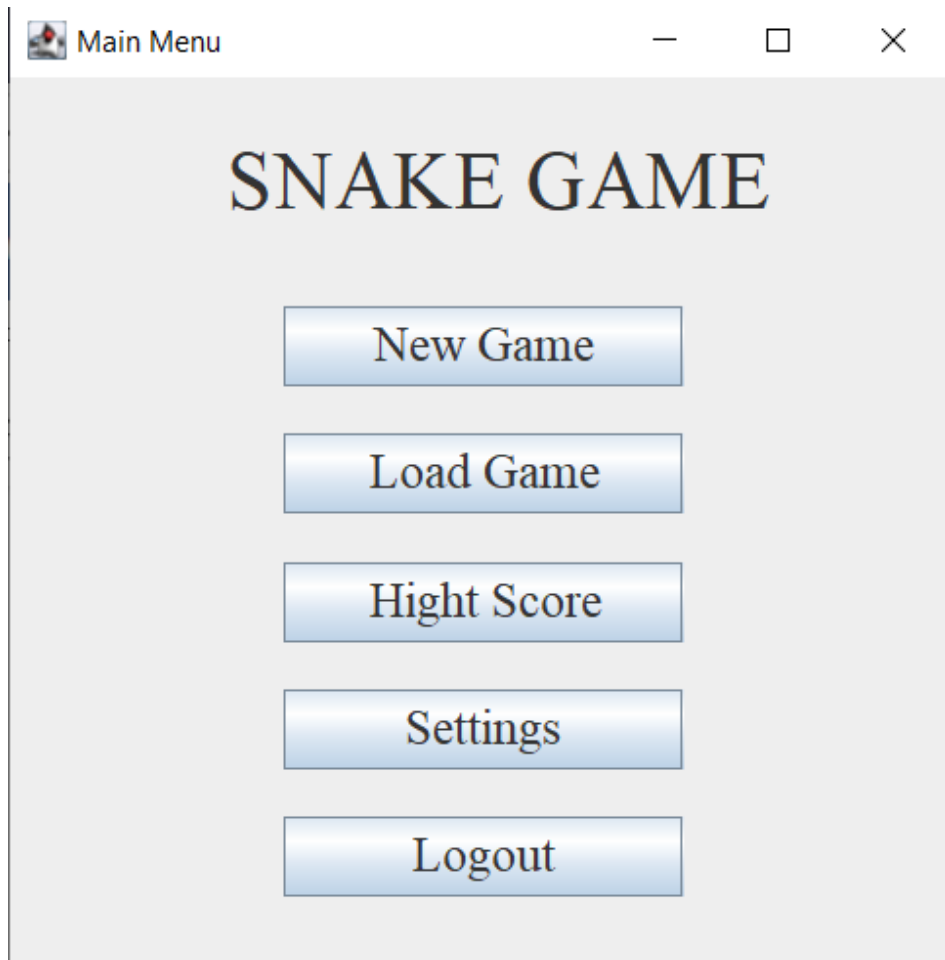
Login bouton(Login écrans): Passer de l'écran de connexion à l'écran principal

Exit bouton: quitter le programme

The image shows a Java Swing window titled "Register Page". Inside the window, the text "SNAKE GAME" is displayed at the top. Below it, there is a "Register" section containing three text input fields labeled "Username:", "Password:", and "Player Name:". Underneath these fields are two buttons labeled "Login" and "Register". At the bottom of the window, there is a large button labeled "EXIT".

Login button(Register écrans): Passer de l'écran d'enregistrement à l'écran de connexion

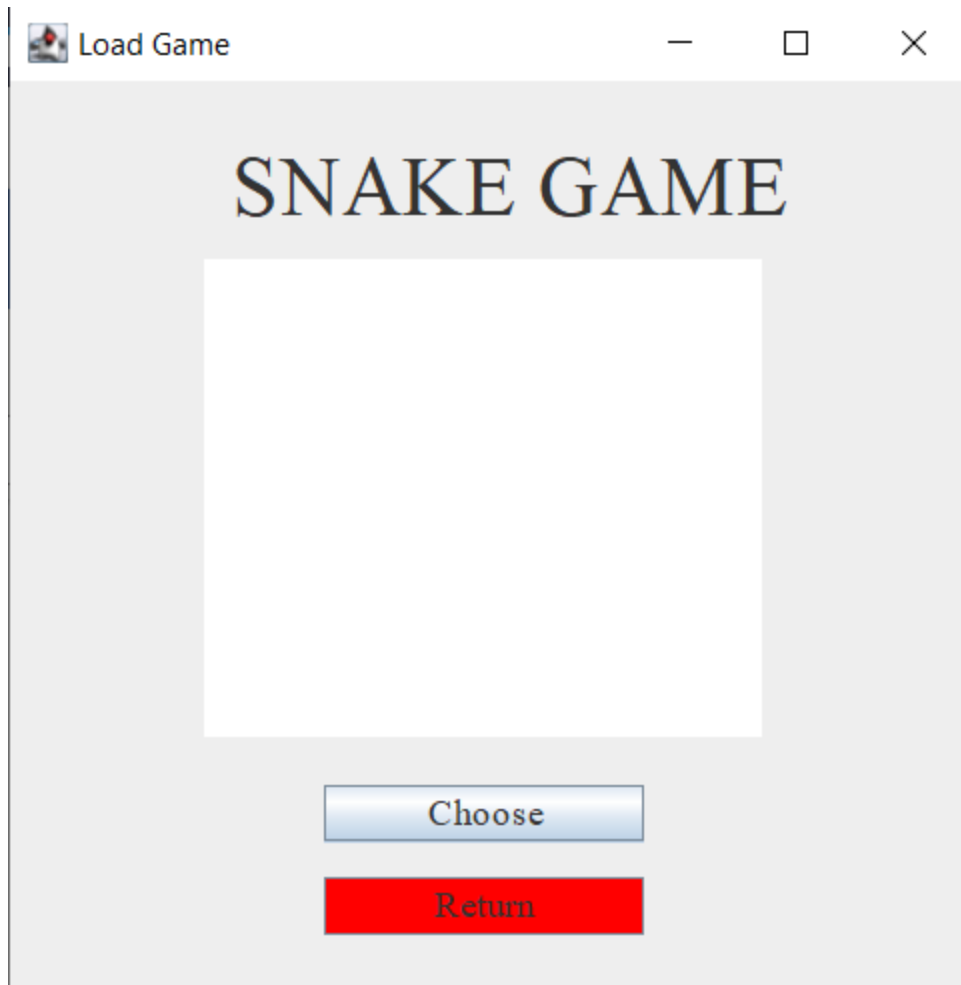
Exit bouton: quitter le programme



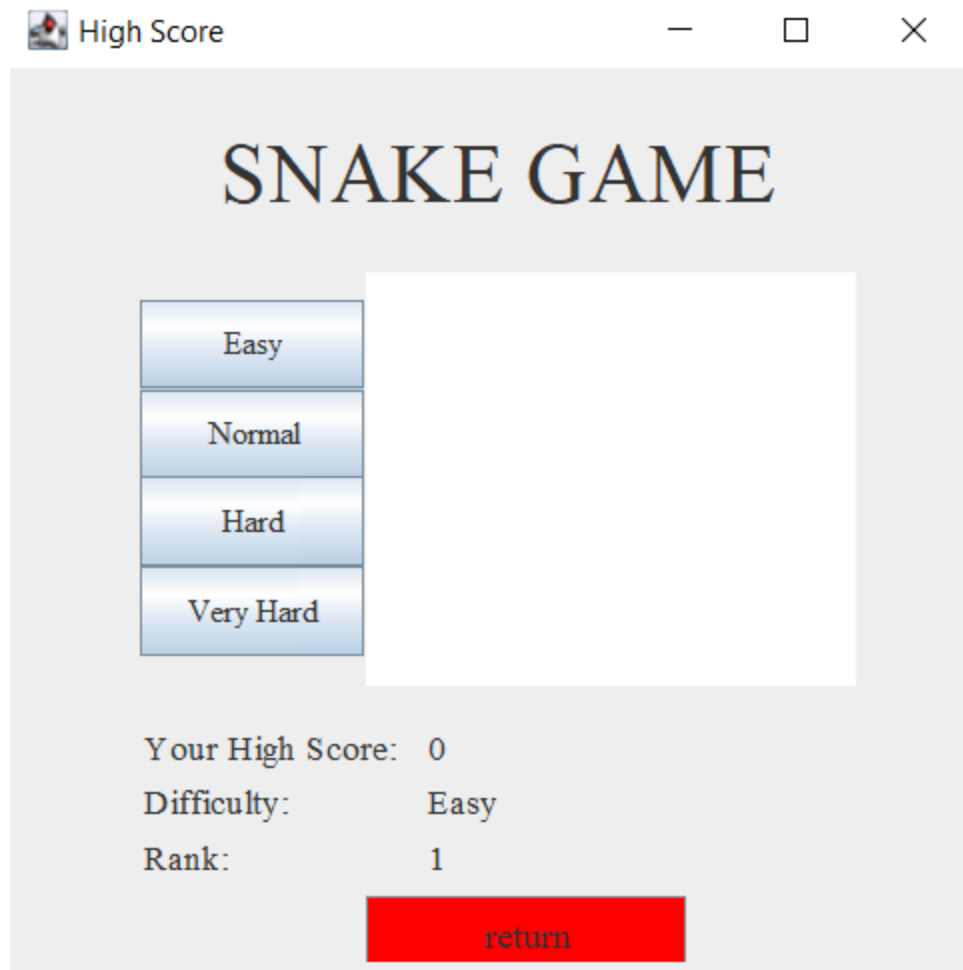
New game bouton: Passer de l'écran principal à l'écran de jeu



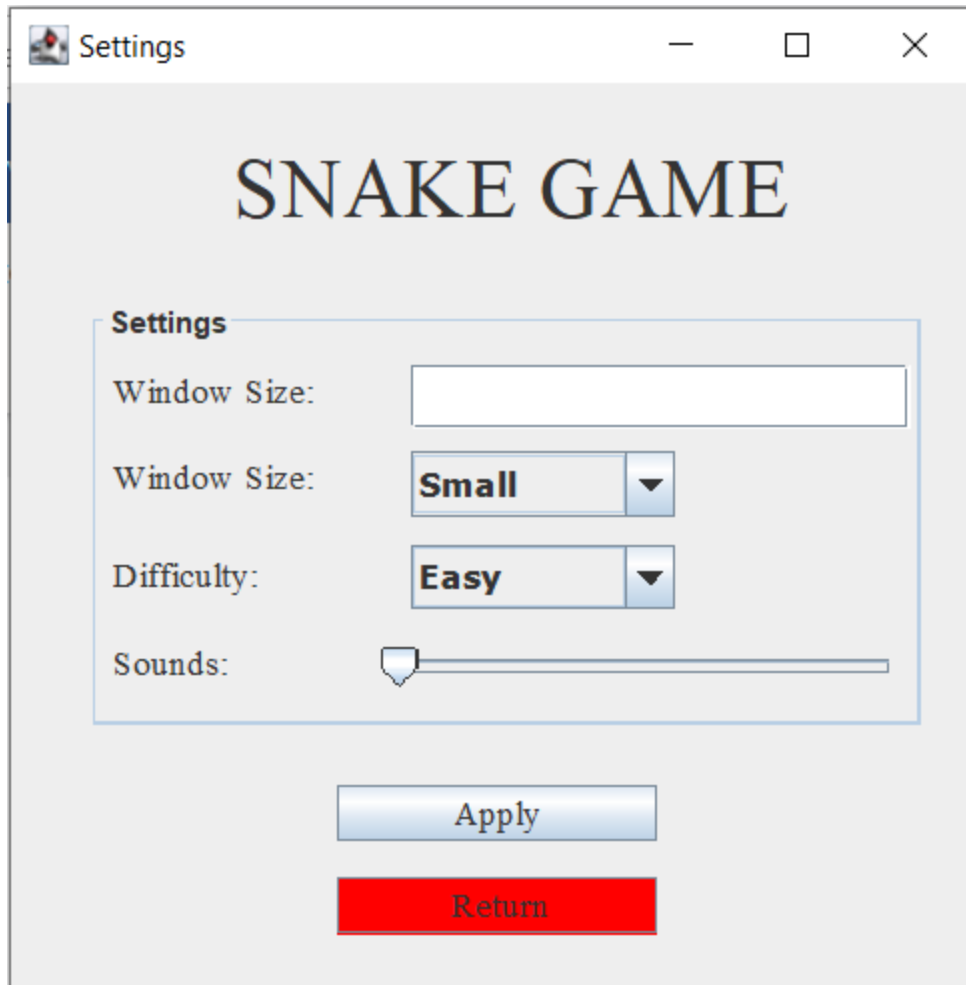
Load game bouton: Passer de l'écran principal à l'écran de chargement du jeu



High score bouton: Passer de l'écran principal à l'écran de score élevé



Settings bouton: Passer de l'écran principal à l'écran de réglage



Logout bouton: Passer de l'écran principal à l'écran d'enregistrement

Return bouton: revenir à l'écran principal



## Résultat

*[Expliquez ce que vous obtenez dans cette partie (GUI), des avantages, désavantages et les solutions prévues (si c'est possible)]*

Utilisez window builder:

Avantages :

- Eclipse WindowBuilder est un concepteur d'interface graphique Java bidirectionnel puissant et facile à utiliser qui permet de créer très facilement des applications d'interface graphique Java sans passer beaucoup de temps à écrire du code pour afficher des formulaires simples.
- Le développeur peut se déplacer en toute transparence entre un concepteur Drag n 'Drop et le code généré.
- En utilisant Eclipse WindowBuilder, le développeur appréciera de créer des applications basées sur l'interface graphique Java. On peut créer des fenêtres compliquées en quelques minutes en utilisant WindowBuilder.

Inconvénients :

- Le code n'a pas de formulaire à modifier.
- Pas d'annotation ne peut pas comprendre.

### Plan

Semaine	De jour - En jour	Le travailler	Le réalisateur
1	23 - 29/03/2020	Conception de l'interface	Hoàng Anh, Quỳnh Hoa.
2	30/03 - 05/04/2020	Conception de base de backend de jeu	Mạnh Hà.
3	06-12/04/2020	Jeu d'essai (Aucune interface)	Mạnh Hà, Quỳnh Hoa, Hoàng Anh.
4	13-19/04/2020	Jeu de base complet avec des interfaces	Mạnh Hà, Quỳnh Hoa, Hoàng Anh.
5	20-26/04/2020	Objets de jeu supplémentaires	Hoàng Anh, Quỳnh Hoa.
6	27/04 - 03/05/2020	Ajouter le menu	Mạnh Hà (Hoàng Anh, Quỳnh Hoa.)
7	04-10/05/2020	Ajouter la base de données	Mạnh Hà, Quỳnh Hoa, Hoàng Anh.
8	11-17/05/2020	Retester et terminer le jeu	Mạnh Hà, Quỳnh Hoa, Hoàng Anh.
9	18-24/05/2020		

17VP – programmation en java nom de groupe

---

10	25-31/05/2020		
----	---------------	--	--

## Références

*[Listez tous les ressources à utiliser dans votre projet, y compris les codes existantes, les algorithmes utilisés, les livres, les rapports, les liens à consulter, etc.]*