



PROJET TUTEURE APL2.1

2021

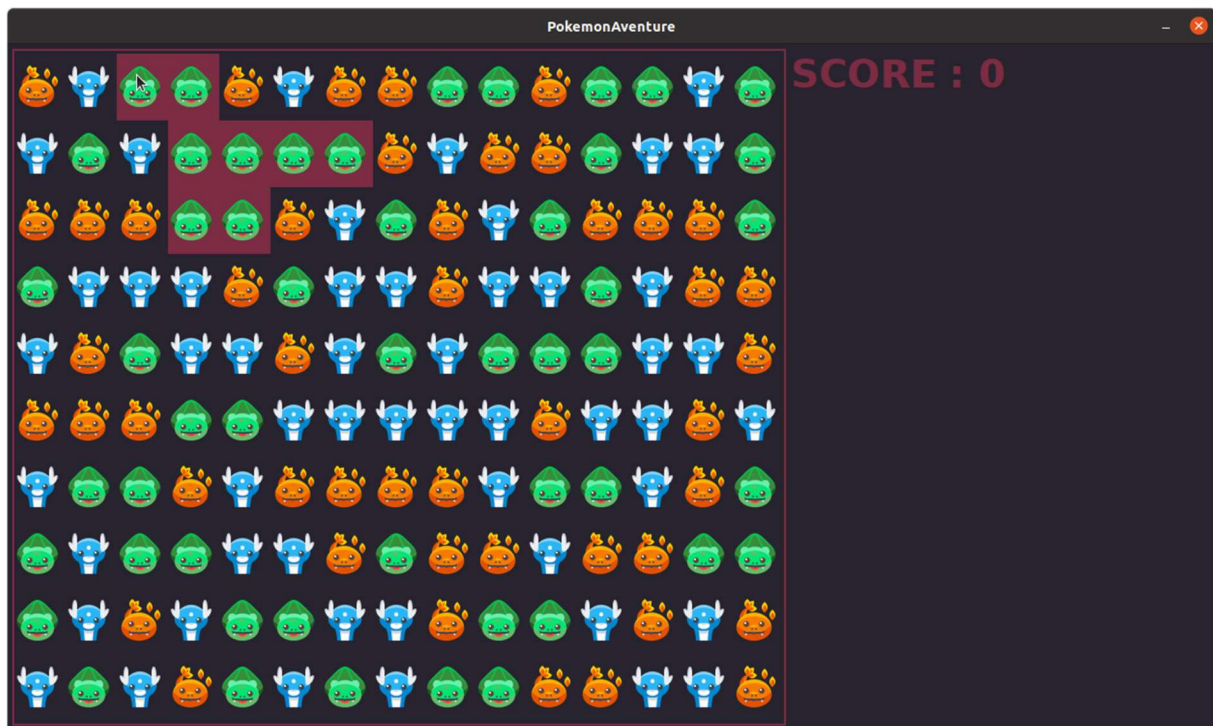
Table des matières

Introduction.....	3
Description du programme	3
Structure du programme.....	4
Algorithme identifiant les groupes.....	5
Procédé.....	6
Conclusion	7
Annexes	8

Introduction

Ce rapport traitera de la conception du jeu « PokémonAventure ». Ce jeu a été réalisé en JAVA et en binôme dans le cadre d'un projet de l'IUT.

Description du programme



Ecran de jeu principal

Ce jeu est une grille composée de 15 colonnes et 10 lignes de blocs. Le but est de faire évoluer son Pokémon en cliquant sur les groupes de blocs.

Afin de vider la grille, il suffit de cliquer sur une image qui fera disparaître tout son groupe. Une case seule, n'appartenant à aucun groupe, ne peut pas disparaître. La fin du jeu est déclenchée lorsque toutes les images restantes n'appartiennent plus à aucun groupe. Lorsqu'une image est supprimée, l'image au-dessus de cette dernière prend sa place, provoquant ainsi l'effet que les images tombent au fur et à mesure du jeu. Lorsqu'une colonne est entièrement vidée, la colonne à sa droite prend sa place. Au lancement du jeu, le joueur a deux choix : générer une grille aléatoirement ou importer une grille préparée dans un fichier.

Le score du joueur varie en fonction du nombre de bloc supprimé. Ainsi, plus il y a de bloc dans un groupe supprimé plus le gain de point sera élevé. Le joueur commence avec le Pokémon « Salamèche », puis obtient « Réptincel » puis « Dracaufeu ».


Structure du programme

Le programme s'exécute à partir du fichier « Main.java ». Dans cette classe on appelle la méthode « Fenetre » issu de la classe « Fenetre.java ».

Cette classe initialise une fenêtre graphique que l'on a modifié à notre guise (Logo de l'application, dimension, titre, ...).

Dans cette classe est ajouté un Panel. Celui-ci contient de nombreux éléments esthétiques tel que le titre ou encore les règles du jeu. C'est également dans ce Panel que les deux boutons « importer » et « jouer » ont été ajouté. Ces boutons permettent de faire un choix entre les deux possibilités suivantes :

- Si l'on clique sur le bouton « jouer », la fenêtre actuelle sera fermée et la méthode « Grille » sera appelé ainsi que la classe « RemplirCase » qui générera une grille aléatoirement.
- Si le joueur clique sur « importer » il devra choisir un fichier texte qui générera la grille en fonction du fichier par la méthode « RemplirCase ». Puis il sera dirigé vers la classe « Grille.java ».



```
RBVRBVRBVRBBRRV
RRVBRVBRRBVRVRV
RBVRBRVBVRRRRVB
RVBRVRVRBRRRVBR
BBRRBRRVBVRBBBR
VVRBVRRVVVRVBRB
RBVRBRBRVRRVRB
RBRRVRVRBRVRVRB
RVRRBVRBRVRVRB
RVBRVRBRBBBVRV
```

Exemple de grille générée par « RemplirCase »

La classe « Grille.java » initialise une nouvelle fenêtre dans laquelle nous avons créé deux parties. La première est dédiée à la grille de jeu et la seconde est dédiée au score et au changement de Pokémon. Dans cette classe il y a la méthode « fillGrille ». Cette dernière permet de dessiner la grille aléatoirement. Il y a également « dessinScoreInit », qui permet de dessiner la partie score du joueur au départ du jeu. Cette classe est aussi relié à la classe « Son.java ». C'est grâce à la méthode « run » issu de cette classe que l'on peut lire un fichier audio et donc mettre le générique de Pokémon au sein du jeux.

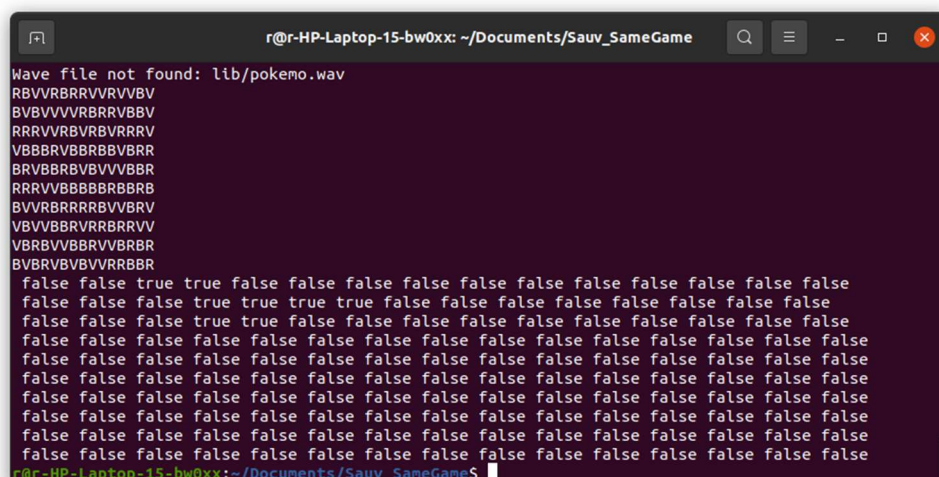
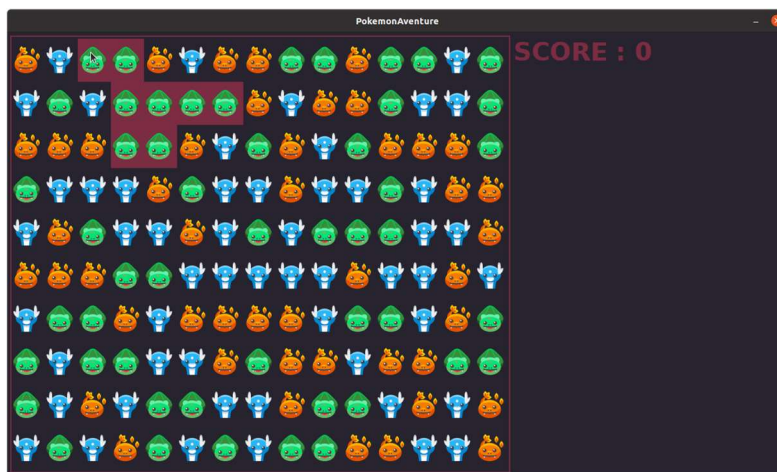
Tous les évènements de la souris au sein de « Grille.java » sont gérés par « EventGame.java » (voir : Algorithme identifiant les groupes).

Lorsque la fin est détectée, la fenêtre de « Grille.java » est alors fermée et on ouvre une nouvelle fenêtre grâce à la classe « FenetreFin.java ». A la manière de « Fenetre.java », celle-ci est modifiée pour nous convenir et est composé d'un Panel nommé « MenuFin.java ». Ce panel dispose de différent élément de décoration et dispose de deux boutons : un nommé « fermer » et l'autre « rejouer ». Le bouton « rejouer » ferme cette fenêtre et ouvre à nouveau « Fenetre.java » ce qui permet de recommencer une nouvelle partie. Le bouton « ferme » quant à lui permet d'arrêter l'exécution du programme.

(Voir diagramme de classe en annexes et dans le répertoire Git).

Algorithme identifiant les groupes

La méthode la plus importante du jeu est celle qui s'occupe de regrouper les images en fonction de si elles sont de la même couleur. Cette fonction est gérée par la méthode « Groupelcon » dans la classe « EventGame ». Cette méthode est utilisée de manière récursive : cela signifie que dans certains cas, elle s'appelle elle-même. Afin de détecter, la méthode utilise deux tableaux : le premier est le tableau de lettre généré par la classe « RemplirCase ». Ce tableau comporte une suite de « R », « V » et « B » rangé sur 10 lignes et 15 colonnes. Le second tableau est un tableau de booléen (TRUE/FALSE) initialisé par défaut à FALSE, d'une taille de 10 lignes et 15 colonnes. La méthode « Groupelcon » utilise également deux entiers en paramètres : la coordonnée x et la coordonnée y de la case dont il faut vérifier si elle fait partie d'un groupe. Lorsque la méthode est utilisée, celle-ci passe immédiatement l'état de la case à TRUE. Ensuite, elle va lire la lettre du tableau de lettre correspondant aux coordonnées qui lui ont été passé en paramètre. Une fois cette étape faite, elle vérifie si au-dessus, à droite, à gauche, ou en dessous, une case comportant la même lettre existe. Si c'est le cas, alors la méthode va s'appeler elle-même (principe de récursivité) en passant les paramètres de la case trouvée. Si jamais une case est seule, la case sera passé à l'état de TRUE mais sera immédiatement repassé à l'état de FALSE puisqu'elle n'aura aucun voisin. Une fois que la méthode a fini, on peut voir le groupe apparaitre avec la valeur TRUE sur le tableau de booléen.



Ici, on peut voir un exemple de l'utilisation de la méthode « Groupelcon ».

Procédé

Pour réaliser ce projet, nous avons opté pour une technique différente du dernier projet. La fois dernière, nous travaillions systématiquement en appel vocal, ensemble, sur une même partie du programme. C'était une technique efficace pour avancer mais la progression était lente. Ici, nous avons décidé de se répartir les tâches mais de nous entraider lors d'un problème. Le projet a donc été réalisé plus rapidement. Pour les parties compliquées du projet, on s'appelait afin de réfléchir ensemble et de mettre en commun nos idées, ce qui a porté ces fruits. Néanmoins, nous avons eu un problème important : nous nous sommes compte qu'une partie de notre projet était construite autour d'une classe « anonyme ». Supprimer cette classe a été long puisque tout notre programme était construit autour de cette classe anonyme. Malgré ce problème nous avons réussi à la modifier et à la supprimer en incluant les `MouseListener` (les événements liés à la souris) dans la classe « `EventGame.java` ». Une partie du code a dû être retravaillée.

```
//Cette fonction permet de faire pleins d'éléments en fo  
public void MouseClick() {  
    jeuxp.addMouseListener(new MouseAdapter() {  
        @Override  
        public void mousePressed(MouseEvent e) {  
            int casex = 0;  
            int casey = 0;  
        }  
    });  
}
```

Classe anonyme qui a été supprimée

Nous avons également rencontré un autre problème, nous disposons d'un fichier `Son.java` qui permet de lire un fichier nommé « `pokemo.wav` ». Ce fichier est une piste audio qui est lue le temps de notre jeu. Malheureusement, l'audio « `pokemo.wav` » est trop volumineux pour l'ajouter sur notre dépôt « `git` ». N'étant qu'un plus ; nous avons décidé de supprimer ce fichier et de le laisser uniquement sur les versions locales de nos ordinateurs.

Conclusion

Nicolas AINOUZ :

Ce projet était très intéressant, il m'a dans un premier temps permis de voir mon évolution en logique. En effet l'appréhension du jeu était moins présente par rapport aux premier projet réalisé en début d'année.

De plus, la bibliothèque graphique proposé par JAVA était un réel plaisir à manier et m'a permis d'expérimenter d'avantages de combinaison en étant libre de mes choix.

Je suis content de vous présenter ce projet car je le trouve plus aboutis que le premier d'une part d'un point de vu graphique mais aussi dans la jouabilité (le premier projet comportait quelques bugs). Avec Corentin nous avons décidés de garder le même binôme et nous nous sommes très bien organisés, nous avons avancés à notre rythme sans être submergé par le travail comme au précédent projet. Une belle réussite pleine de nostalgie !

Corentin RODIER :

Ce projet a été un réel plaisir pour ma part. J'ai bien aimé la partie graphique qui est plus aboutie que le dernier jeu réalisé (Blocus en C). Le point négatif et stressant de cette réalisation a été de devoir supprimer la classe anonyme : nous avons dû reprendre une partie du code pour pouvoir la supprimer. Dommage que l'on ne se soit pas rendu compte plus tôt que cette classe allait être gênante.

Mon petit coup de cœur a été la réalisation de tout les évènements liés à la grille. J'ai adoré créer la méthode Groupelcon et toutes les méthodes qui lui sont similaires.

Diagramme de classe

