Projet de :
Bazile Oliver: 12313244
Nizar Abak-Kali: 11290569

Programmation Nomade :
Mr Yfithen Kacimi

Casse-tête rapport

Sommaire :

1. Introduction

Dans le cadre du cours de Programmation Nomade, nous avons choisi le projet casse-tête en Android . Ce projet est un puzzle où le but consiste à glisser différentes pièces et d'essayer de toutes les faire entrer dans un bloc , le tout sous un temps imparti.
Il nous a été demandé dans le cadre du projet de créer trois niveaux statiques , chaque niveau contenant un certain nombre de pièce à intercaler en un temps imparti pour finir le puzzle . Ensuite, il nous a été demandé de gérer le Drag and Drop pour chaque pièce. Enfin gérer les fonctionnalités du jeu,faire un menu, les conditions du jeu (perdre et gagner) ,la gestion du son .

Lors de ce rapport nous présenterons ,dans un premier temps ,l'architecture du programme ,puis dans un second temps , un brève explication des classes et leur intérêt , enfin une présentation des principales difficultés rencontrées .

2. Développement

a. L'architecture de l'application

Dans cet partie nous présentons les différentes activités qui composent notre applications .

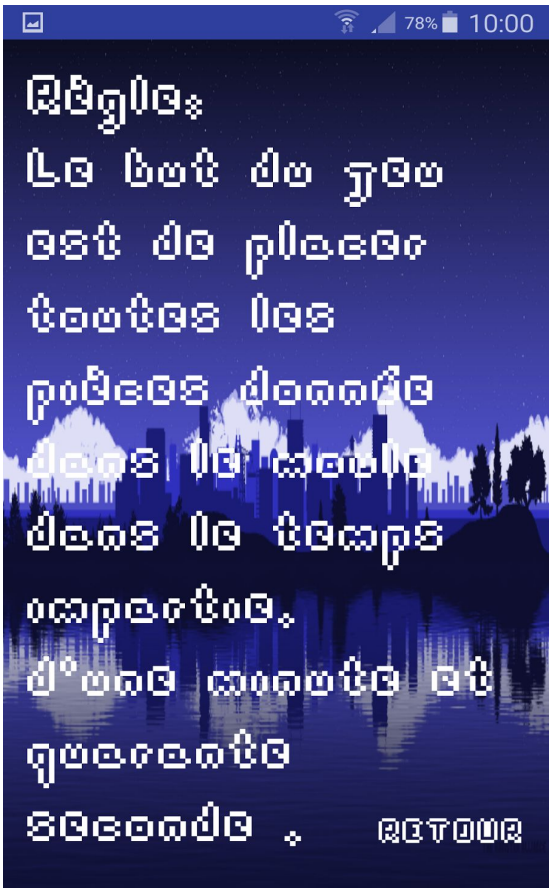
- StartActivity :

Il s'agit de l'activité de départ au lancement de l'application . C'est le menu du jeu . Il utilise le Layout activity_start . C'est aussi lors dans cet activité que le lecteur multimedia est créer et que la musique du jeu est lancé .On peut voir trois boutons : START pour lancer le jeu ,REGLE pour afficher les règles ,CREDIT pour afficher les informations de l'application .


79% 09:57
CASSE TETE
START
REGLE
CRÉDIT

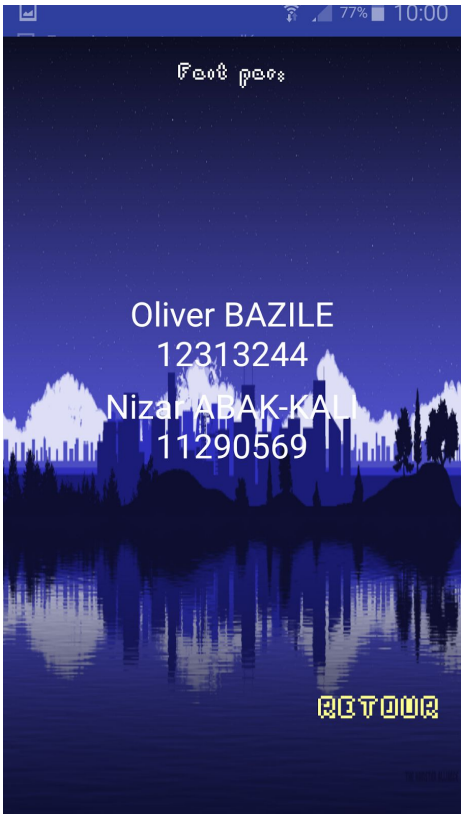
- RegleActivity:

C'est une Activity qui nous informe sur les règles du jeu. Cet activité utilise le layout activity_regle qui est composé d'un relativeLayout et d'un TextView contenant les règles du jeu et un bouton pour pouvoir retourner dans le menu principale.



- <u>AboutActivity :</u>

Il s’agit de l’activité contenant les information sur les développeurs de l’application . Cet activité utilise le layout activity_about qui est composé d’un relativeLayout et des TextView contenant les informations tel que nos et nos numéros étudiants.

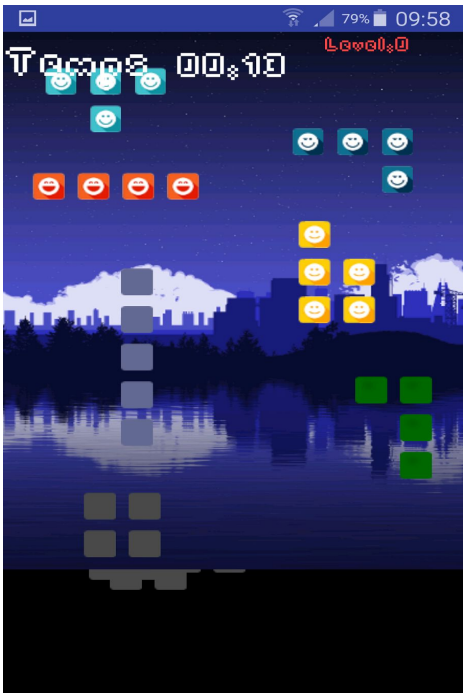


- PlayActivity :

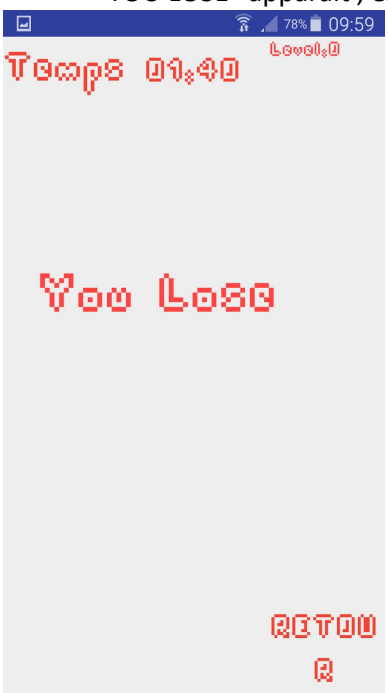
C’est l’activité qui lance le jeu casse tête. Elle gère le son du jeu , le timer , et la fin de jeu . Le layout de cette activité est est composé d’un relativeLayout, d’une surfaceView nommé CassetêteView2 comportant l’animation et les formes du jeu.

Les états de l’application :

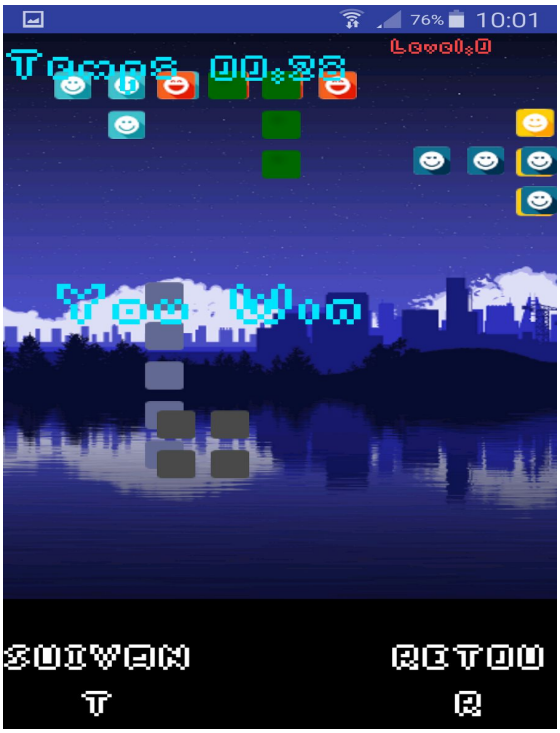
- en jeu :



- Si perdus à cause du timer : si on perd CasseTeteView2 est mis en INVISIBLE , message “YOU LOSE” apparaît , enfin un bouton retour pour retourner sur le menu apparaît .



- Si on gagne: on peut voir afficher deux boutons : suivant qui nous envoie sur un autre niveau aléatoire et retour qui nous ramène au menu principale du jeu .



b. Les classes

Lors de cette partie nous présenterons les classes principales du jeu .

- Bloc.java :

C'est une classe qui consiste à récupérer où à initialiser la taille du bloc, la position en x et en y du bloc. Il permet aussi pour chaque bloc de vérifier si le bloc est fixer et si il est cliquable. Elle définit un bloc.

- Level.java:

Il s'agit d'une classe qui vas représenter un niveau . Cette classe à pour utilité d'être une classe général pour tout les niveaux . La classe à pour attribut une liste de bloc qui compose le niveau et le tableau qui va contenir toutes ces pièces .Cette classe a une méthode public "add_Block()" qui permet d'ajouter un bloc à la liste de bloc .

- CasseTeteView2.java:

Cette classe gère le jeu ainsi que les intéractions avec celui-ci . Cette classe hérite de la classe SurfaceView .Voici une présentations des méthodes les plus intéressantes :

adding_levels(): Méthodes dans la quelles on appelles les méthodes de création des niveaux statiques . Chaque méthode adding_énièmelevel crée des blocs que l'on remplit de manière statique puis que l'on ajoute à un level instancier , ce level est ensuite ajouté à liste de niveau "levels" .

initparameters() : il s'agit d'une méthode qui est appelé pour initialiser tout les paramètres du jeu mais c'est dans cette méthode qu'est choisi de manières aléatoire le niveau .

PaintCarte(): Dessine le bloc qui va contenir les blocs. Ce bloc est dessiné en fonction du tableau.

PaintBloc(): Pareille que pour PaintCarte sauf que pour chaque bloc on dessine son tableau spécifique .

InBox(): Fonction qui vérifie si la pièce et dans le tableau du niveau courant où en dehors.

YouWin(): Vérifie si toutes les bloc sont placés dans le tableau du niveau courant.

OnTouchEvent(MotionEvent event): méthode qui va gérer les interactions de l'utilisateur avec l'écran . Lorsque l'utilisateur touche un bloc la méthode change l'attribut est_touche du bloc à vrai ainsi on sait quel bloc est touché et donc quelle bloc bouger. Une fois le bloc positionné dans le tableau l'attribut est_fixe du bloc est passer à vrai.

c. Les difficulté rencontrées

Lors de ce projet nous avons rencontrés des **difficultés, parmi** celle-ci les problèmes de collision entre les pièces et **gestion** de la fin du jeu ont été les plus importantes .

3. Conclusion:

Lors de ce projet il nous à été demandé de faire un Casse-tête contenant trois niveaux déclarés de manière statique dans le code .Ce qui à été fait . Malgré cela, il subsiste certains point négatif dans notre projet . Les collisions entre les pièces n'est pas gérés , le tableau censé contenir les blocs existe physiquement mais seulement la première colonnes est visible pour une raison qui nous échappe . Finalement certaines fonctionnalités que nous voulions intégrer n'ont pas pus voir le jour , tel la génération automatique de niveau grâce à un algorithme, la gestion du volume du son , la création d'un score persistant dans une base de données SQLite.