Application ANDROID / iOS : Jeu Snake

SINAMA Richard
LE LIDEC Tristan

L3 Informatique

2 mai 2021

Plan

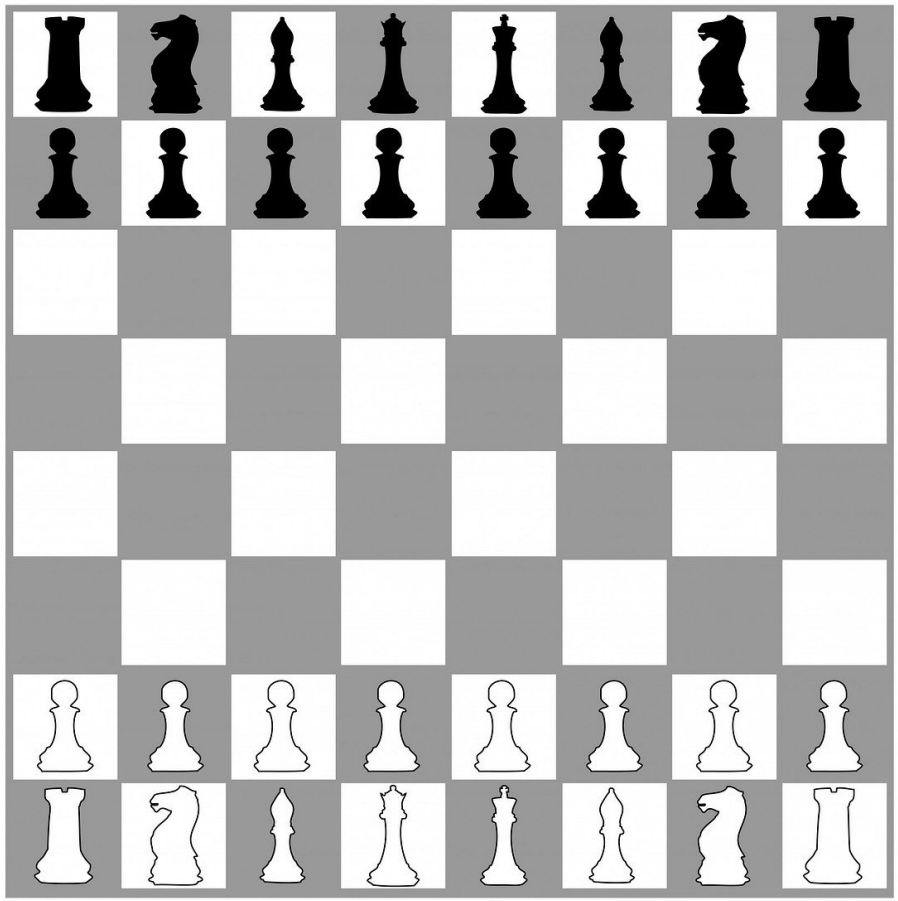
Introduction

Réalisation d'une application mobile.

Plan

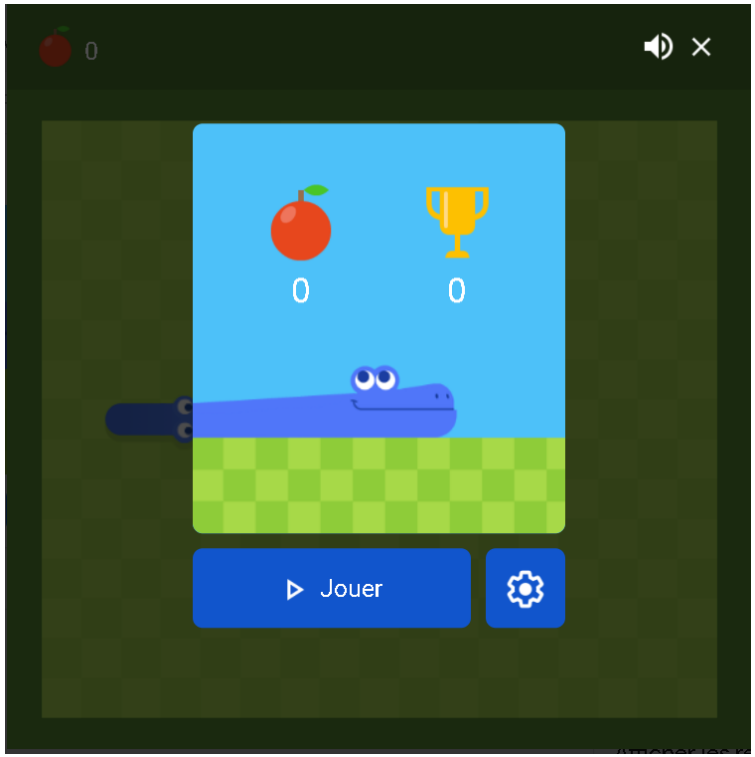
Jeu d'échecs

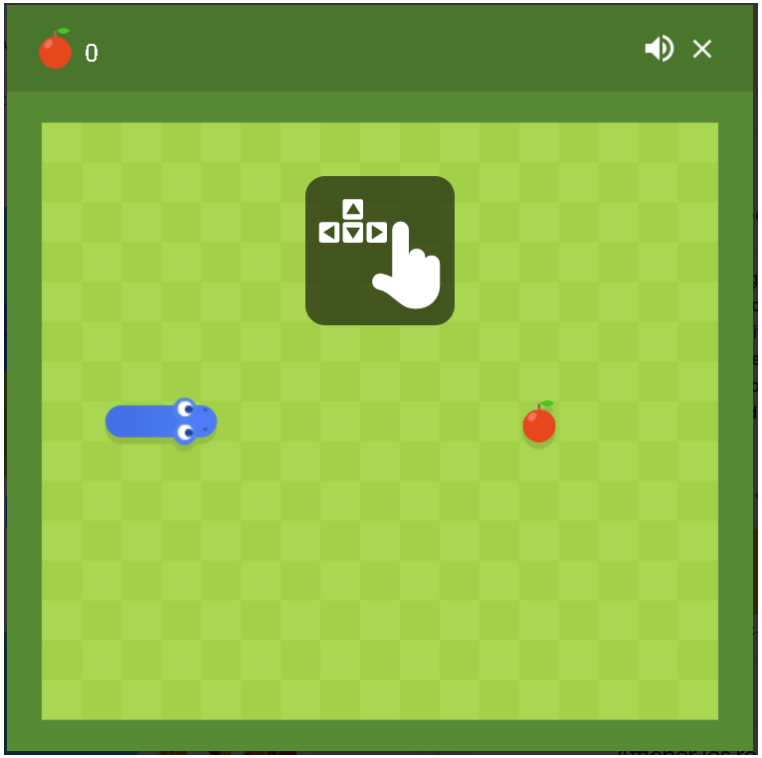
Notre jeu d'échec est un échec



Jeu Snake

Jeu de snake de Google

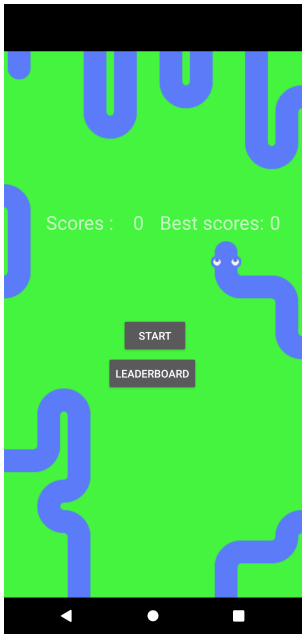


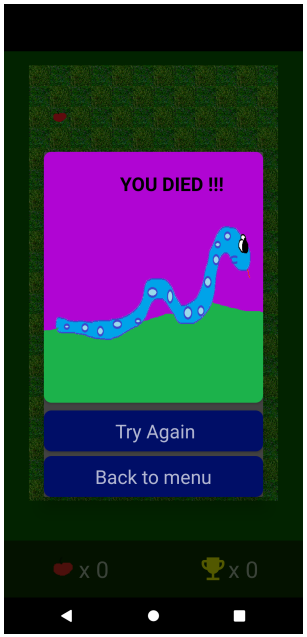


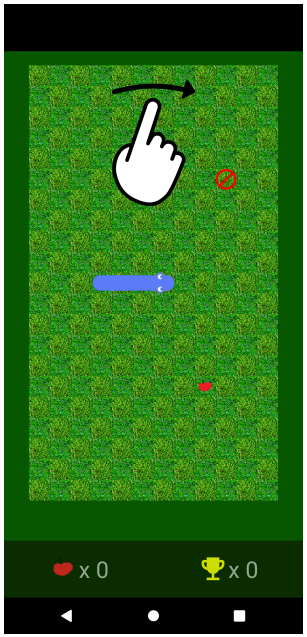
Plan

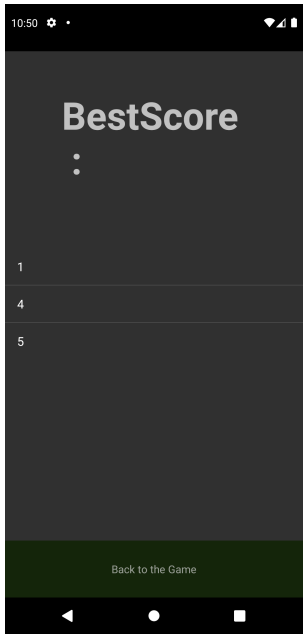
Notre jeu

Notre touche personnelle :









Logique du jeu

Le but du jeu est de contrôler un serpent afin de manger des pommes et d'esquiver les obstacles pour obtenir le meilleur score.

Le Snake

Création de l'objet snake :

```
public Snake(Bitmap bm, int x, int y, int length) {
    this.bm = bm;
    this.length = length;
    bm_body_bottom_left = Bitmap.createBitmap(bm, x: 0, y: 0, width: bm.getWidth()/14, bm.getHeight());
    bm_body_bottom_right = Bitmap.createBitmap(bm, x: bm.getWidth()/14, y: 0, width: bm.getWidth()/14, bm.getHeight());
    bm_body_horizontal = Bitmap.createBitmap(bm, x: 2*bm.getWidth()/14, y: 0, width: bm.getWidth()/14, bm.getHeight());
    bm_body_top_left = Bitmap.createBitmap(bm, x: 3*bm.getWidth()/14, y: 0, width: bm.getWidth()/14, bm.getHeight());
    bm_body_top_right = Bitmap.createBitmap(bm, x: 4*bm.getWidth()/14, y: 0, width: bm.getWidth()/14, bm.getHeight());
    bm_body_vertical = Bitmap.createBitmap(bm, x: 5*bm.getWidth()/14, y: 0, width: bm.getWidth()/14, bm.getHeight());
    bm_head_down = Bitmap.createBitmap(bm, x: 6*bm.getWidth()/14, y: 0, width: bm.getWidth()/14, bm.getHeight());
    bm_head_left = Bitmap.createBitmap(bm, x: 7*bm.getWidth()/14, y: 0, width: bm.getWidth()/14, bm.getHeight());
    bm_head_right = Bitmap.createBitmap(bm, x: 8*bm.getWidth()/14, y: 0, width: bm.getWidth()/14, bm.getHeight());
    bm_head_up = Bitmap.createBitmap(bm, x: 9*bm.getWidth()/14, y: 0, width: bm.getWidth()/14, bm.getHeight());
    bm_tail_up = Bitmap.createBitmap(bm, x: 10*bm.getWidth()/14, y: 0, width: bm.getWidth()/14, bm.getHeight());
    bm_tail_right = Bitmap.createBitmap(bm, x: 11*bm.getWidth()/14, y: 0, width: bm.getWidth()/14, bm.getHeight());
    bm_tail_left = Bitmap.createBitmap(bm, x: 12*bm.getWidth()/14, y: 0, width: bm.getWidth()/14, bm.getHeight());
    bm_tail_down = Bitmap.createBitmap(bm, x: 13*bm.getWidth()/14, y: 0, width: bm.getWidth()/14, bm.getHeight());
    setMove_right(true);
    arrPartSnake.add(new PartSnake(bm_head_right, x, y));
    for (int i = 1; i < length-1; i++){
        this.arrPartSnake.add(new PartSnake(bm_body_horizontal, x: this.arrPartSnake.get(i-1).getX()-GameView.sizeElementMap, y));
    }
    arrPartSnake.add(new PartSnake(bm_tail_right, x: arrPartSnake.get(length-2).getX()-GameView.sizeElementMap, arrPartSnake.get(length-2).getY()));
}
```

Le Snake

Update du snake :

```
public void update(){
    for(int i = length-1; i > 0; i--){
        arrPartSnake.get(i).setX(arrPartSnake.get(i-1).getX());
        arrPartSnake.get(i).setY(arrPartSnake.get(i-1).getY());
    }
    if(move_right){
        arrPartSnake.get(0).setX(arrPartSnake.get(0).getX()+GameView.sizeElementMap);
        arrPartSnake.get(0).setBm(bm_head_right);
    }else if(move_down){
        arrPartSnake.get(0).setY(arrPartSnake.get(0).getY()+GameView.sizeElementMap);
        arrPartSnake.get(0).setBm(bm_head_down);
    }else if(move_up){
        arrPartSnake.get(0).setY(arrPartSnake.get(0).getY()-GameView.sizeElementMap);
        arrPartSnake.get(0).setBm(bm_head_up);
    }else{
        arrPartSnake.get(0).setX(arrPartSnake.get(0).getX()-GameView.sizeElementMap);
        arrPartSnake.get(0).setBm(bm_head_left);
    }
```

Le Snake

Update du snake (suite) :

```
for (int i = 1; i < length - 1; i++){
    if(arrPartSnake.get(i).getrLeft().intersect(arrPartSnake.get(i+1).getrBody())
            &&arrPartSnake.get(i).getrBottom().intersect(arrPartSnake.get(i-1).getrBody())
            ||arrPartSnake.get(i).getrBottom().intersect(arrPartSnake.get(i+1).getrBody())
            &&arrPartSnake.get(i).getrLeft().intersect(arrPartSnake.get(i-1).getrBody())){
        arrPartSnake.get(i).setBm(bm_body_bottom_left);
    }else if (arrPartSnake.get(i).getrLeft().intersect(arrPartSnake.get(i+1).getrBody())
            &&arrPartSnake.get(i).getrTop().intersect(arrPartSnake.get(i-1).getrBody())
            ||arrPartSnake.get(i).getrLeft().intersect(arrPartSnake.get(i-1).getrBody())
            &&arrPartSnake.get(i).getrTop().intersect(arrPartSnake.get(i+1).getrBody())){
        arrPartSnake.get(i).setBm(bm_body_top_left);
    }else if (arrPartSnake.get(i).getrRight().intersect(arrPartSnake.get(i+1).getrBody())
            &&arrPartSnake.get(i).getrTop().intersect(arrPartSnake.get(i-1).getrBody())
            ||arrPartSnake.get(i).getrRight().intersect(arrPartSnake.get(i-1).getrBody())
            &&arrPartSnake.get(i).getrTop().intersect(arrPartSnake.get(i+1).getrBody())) {
        arrPartSnake.get(i).setBm(bm_body_top_right);
    }else if(arrPartSnake.get(i).getrRight().intersect(arrPartSnake.get(i+1).getrBody())
            &&arrPartSnake.get(i).getrBottom().intersect(arrPartSnake.get(i-1).getrBody())
            ||arrPartSnake.get(i).getrRight().intersect(arrPartSnake.get(i-1).getrBody())
            &&arrPartSnake.get(i).getrBottom().intersect(arrPartSnake.get(i+1).getrBody())){
        arrPartSnake.get(i).setBm(bm_body_bottom_right);
    }else if(arrPartSnake.get(i).getrLeft().intersect(arrPartSnake.get(i-1).getrBody())
            &&arrPartSnake.get(i).getrRight().intersect(arrPartSnake.get(i+1).getrBody())
            ||arrPartSnake.get(i).getrLeft().intersect(arrPartSnake.get(i+1).getrBody())
            &&arrPartSnake.get(i).getrRight().intersect(arrPartSnake.get(i-1).getrBody())){
        arrPartSnake.get(i).setBm(bm_body_horizontal);
    }else if(arrPartSnake.get(i).getrTop().intersect(arrPartSnake.get(i-1).getrBody())
            &&arrPartSnake.get(i).getrBottom().intersect(arrPartSnake.get(i+1).getrBody())
            ||arrPartSnake.get(i).getrTop().intersect(arrPartSnake.get(i+1).getrBody())
```

Le Snake

Update du snake (suite) :

```
            arrPartSnake.get(i).setBm(bm_body_horizontal);
        }else if(arrPartSnake.get(i).getrTop().intersect(arrPartSnake.get(i-1).getrBody())
                &&arrPartSnake.get(i).getrBottom().intersect(arrPartSnake.get(i+1).getrBody())
                ||arrPartSnake.get(i).getrTop().intersect(arrPartSnake.get(i+1).getrBody())
                &&arrPartSnake.get(i).getrBottom().intersect(arrPartSnake.get(i-1).getrBody())){
            arrPartSnake.get(i).setBm(bm_body_vertical);
        }else{
            if(move_right){
                arrPartSnake.get(i).setBm(bm_body_horizontal);
            }else if(move_down){
                arrPartSnake.get(i).setBm(bm_body_vertical);
            }else if(move_up){
                arrPartSnake.get(i).setBm(bm_body_vertical);
            }else{
                arrPartSnake.get(i).setBm(bm_body_horizontal);
            }
        }
    }
    if(arrPartSnake.get(length-1).getrRight().intersect(arrPartSnake.get(length-2).getrBody())){
        arrPartSnake.get(length-1).setBm(bm_tail_right);
    }else if(arrPartSnake.get(length-1).getrLeft().intersect(arrPartSnake.get(length-2).getrBody())){
        arrPartSnake.get(length-1).setBm(bm_tail_left);
    }else if(arrPartSnake.get(length-1).getrBottom().intersect(arrPartSnake.get(length-2).getrBody())){
        arrPartSnake.get(length-1).setBm(bm_tail_down);
    }else{
        arrPartSnake.get(length-1).setBm(bm_tail_up);
    }
}
```

Passage entre les activités

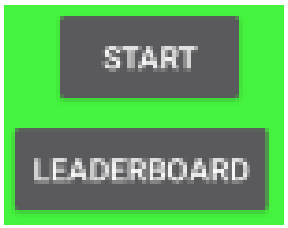
Utilisation des OnClickListener sur bouton afin de naviguer entre les activités :

```
start = (Button) findViewById(R.id.button_start);
start.setOnClickListener(new View.OnClickListener(){
    @Override
    public void onClick(View v) { openGame(); }
});
```

```
RelativeLayout rl_BackGame = findViewById(R.id.rl_backGame);
rl_BackGame.setOnClickListener(new View.OnClickListener(){
    @Override
    public void onClick(View v) { finish(); }
});
```

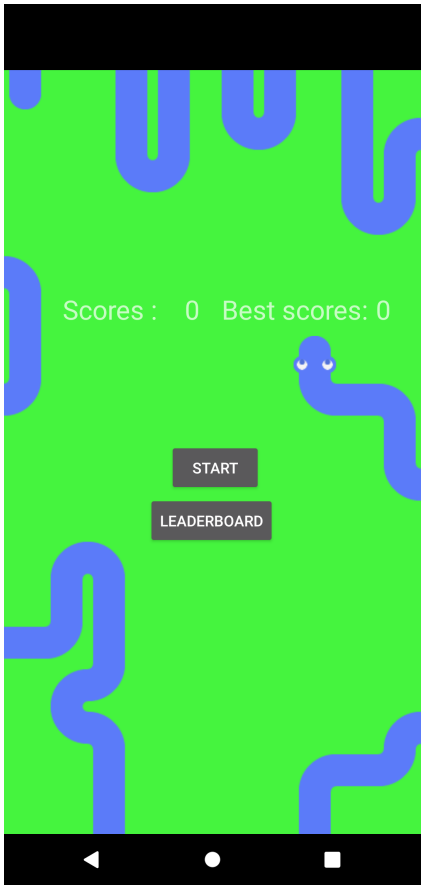
Passage entre les activités

Nos boutons et layouts de déplacements entre activités :



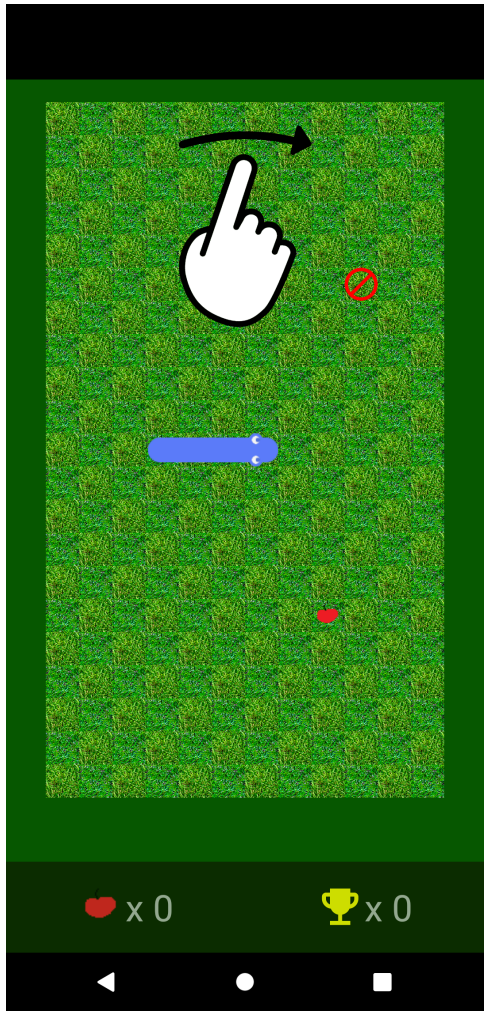
Le menu

Affichage du score et lien permettant d'accéder au leaderboard et du jeu principal :



La zone du jeu

Dessin de texture sur un canvas



Gestion du stop

Gestion du repositionnement toutes les x secondes du stop sur l'écran.

```
//Creating a class to reload the stop every 5 seconds
class stopload extends TimerTask {
    public void run() {
        stop.reset(arrGrass.get(objectPlacementRandom()[0]).getX(), arrGrass.get(objectPlacementRandom()[1]).getY());
    }
}
```

Sauvegarde des données

Récupération des données de scores dans une Liste à la fin de chaque partie.

```
if(bestScore > oldBestScore){
    MainGame.score.add(String.valueOf(bestScore));
}
try {
    sauvegarderDonnees();
} catch (IOException e){
    e.printStackTrace();
}
```

Sauvegarde des données

Fonction de sauvegarde dans un fichier :

```
public void sauvegarderDonnees() throws IOException {
    File file = new File(this.context.getFilesDir(), child: "stats.txt");
    // Si le fichier exste on le supprime
    if(file.exists()) file.delete();
    FileOutputStream stream = new FileOutputStream(file);
    try {
        // les stats sont stockés ligne par ligne
        int length = MainGame.score.size();
        for(int i=0; i < length-1; i++){
            stream.write((MainGame.score.get(i)+"\n").getBytes());
        }
        // Pas de retour à la ligne pour la dernière stat
        stream.write((MainGame.score.get(length-1)).getBytes());
    } catch (IOException e) { e.printStackTrace(); } finally {
        stream.close();
    }
}
```

Sauvegarde des données

Fonction de chargement du fichier :

```
public void chargerDonnees() throws IOException {
    File file = new File(getFilesDir(), child: "stats.txt");
    // Si il n'y a pas de sauvegarde on ne charge rien
    if(!file.exists()) return;
    // Lectures du flux de byte
    int length = (int) file.length();
    byte[] bytes = new byte[length];
    FileInputStream in = new FileInputStream(file);
    try { in.read(bytes); }
    catch (IOException e) { e.printStackTrace(); }
    finally { in.close(); }
    // Conversion des bytes en un seul string
    String contents = new String(bytes);
    // Les stats sont stockées ligne par ligne, on découpe et on ajoute en mémoire
    String[] data = contents.split( regex: "\n");
    for (int i=0; i < data.length; i++){ score.add(data[i]); }
}
```

Sauvegarde des données

Sauvegarde du meilleur dernier meilleur score dans les sharedPreferences

```
if(score > bestScore){
    bestScore = score;
    SharedPreferences sp = context.getSharedPreferences( name: "gamesetting", Context.MODE_PRIVATE);
    SharedPreferences.Editor editor = sp.edit();
    editor.putInt("bestscore", bestScore);
    editor.apply();
    MainActivity.bestscore.setText(bestScore+"");
    MainGame.txt_best_score.setText(bestScore+"");
}
```

Affichage de la liste des meilleurs scores

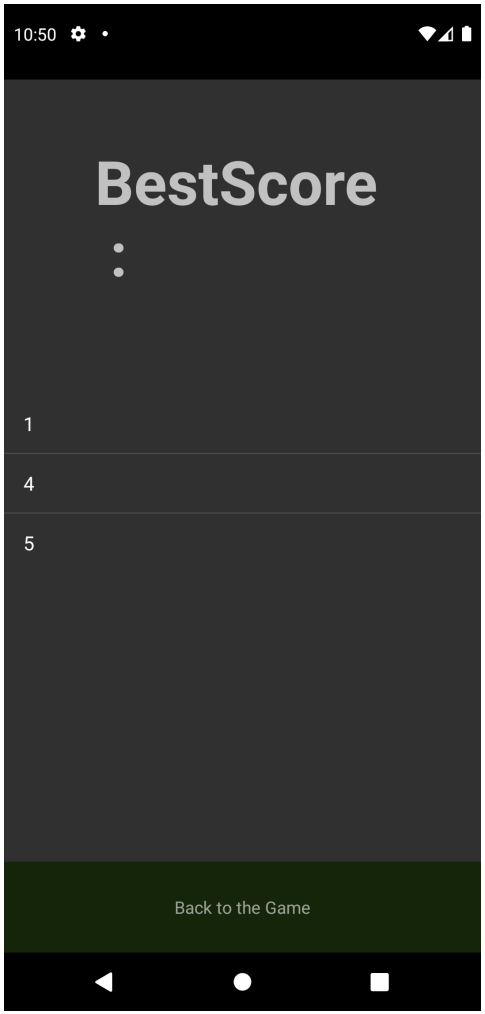
Affichage sur une activité des différents meilleurs scores obtenu depuis l'ouverture du jeu sous forme de liste.

```
<RelativeLayout
    android:layout_centerVertical="true"
    android:layout_centerHorizontal="true"
    android:layout_width="wrap_content"
    android:layout_height="200dp"
    android:layout_marginTop="100dp">
    <ListView
        android:id="@+id/listeScore"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">
    </ListView>
</RelativeLayout>
```

```
ListView listView = (ListView)findViewById(R.id.listeScore);
ArrayAdapter arrayAdapter = new ArrayAdapter( context: this,
                                android.R.layout.simple_list_item_1,
                                MainGame.score);
listView.setAdapter(arrayAdapter);
```

Affichage de la liste des meilleurs scores

Résultat :



Plan

Logique du jeu

Le même que Android mais avec un déplacement par bouton.

La zone de jeu

Création d'un tableau de x*y case qui représente le boardgame

```
func drawGrid(){
    let gridPath = UIBezierPath()

    for i in 0...SnakeBoard.rows{
        gridPath.move(to: CGPoint(x:originX, y:originY + CGFloat(i) * cellSide))
        gridPath.addLine(to: CGPoint(x:originX + CGFloat(SnakeBoard.cols) * cellSide, y:originY + CGFloat(i) * cellSide))
    }

    for i in 0...SnakeBoard.cols{
        gridPath.move(to: CGPoint(x:originX + CGFloat(i) * cellSide, y:originY))
        gridPath.addLine(to: CGPoint(x:originX + CGFloat(i) * cellSide, y:originY + CGFloat(SnakeBoard.rows) * cellSide))
    }

    UIColor.lightGray.setStroke()
    gridPath.stroke()
}
```

Snake

Dessin de carrés représentants les différentes parties du serpent sur le boardgame.

```
func drawSnake(){

        if shadowSnake.isEmpty{
            return
        }
        let snakeHead = shadowSnake.first!

        UIColor.purple.setFill()
        UIBezierPath(roundedRect: CGRect(x: originX + CGFloat(snakeHead.col) * cellSide,
                                         y: originY + CGFloat(snakeHead.row) * cellSide,
                                         width: cellSide, height: cellSide),
                                         cornerRadius: 6).fill()

        UIColor.green.setFill()
        for i in 1..<shadowSnake.count {
            let cell = shadowSnake[i]
            UIBezierPath(roundedRect: CGRect(x: originX + CGFloat(cell.col) * cellSide,
                                             y: originY + CGFloat(cell.row) * cellSide,
                                             width: cellSide, height: cellSide),
                                             cornerRadius: 6).fill()
        }
    }
```

Passage entre activité

Utilisation des UIViewController pour rediriger a l'aide d'un bouton.

```
import UIKit

class MenuViewController : UIViewController {

   @IBAction func Start(_ sender: Any) {
       self.performSegue(withIdentifier: "ViewController", sender: self)


       func viewDidLoad(){
       super.viewDidLoad()

   }
}

}
```

Plan

Contraintes

La création de notre projet développement mobile a été ralentit lors de notre changement de sujet, impliquant plusieurs contraintes :

- De Temps
- De matériel
- La gestion de rotation d'écran

Plan

Android

- Le menu de mort au lancement du jeu
- Le leaderboard sans donnéesk
- La gestion de rotation d'écran

iOS

- L'implémentation de la mort
- Le meilleurs scores

Plan

Conclusion

Pour conclure

-
-
-