

Le Pendu

Dorine Fontaine, L3 informatique

1^{er} mai 2021



Résumé

Dans le cadre de la licence d'informatique, dans l'unité Développement Mobile, il nous a été demandé de développer une application simpliste respectant quelques critères de base. La réalisation de ce projet a été pour moi une occasion supplémentaire d'enrichir mes compétences en programmation et de découvrir la programmation sur AndroidStudio et sous IOS. Le choix de l'application s'est porté sur un pendu.

Avant toute chose, je tiens à remercier tous ceux qui ont participé à la mise en œuvre de ce projet, en particulier M. PAYET, enseignant de la matière, et certain de mes camarades de classe.

Table des matières

1	Introduction	3
2	Description générale de l'application	3
2.1	Contenu	3
2.2	Design	5
3	Interactions et Données au sein du jeu	5
3.1	Android	6
3.2	iOS	7
4	Amélioration	7
5	Conclusion	7

1 Introduction

Depuis quelques années avec le développement de nouvelle technologie, les jeux ont envahi nos téléphones. Même si ceux-ci ont souvent une connotation négative en particulier les jeux vidéo, il aide à stimuler le cerveau et lui permet d'accroître sa logique et sa mémoire. Les jeux auraient même un effet bénéfique sur le cerveau. Lorsque l'opportunité m'a été donnée de réaliser une application, l'idée du jeu du Pendu m'est toute de suite venu. Le Pendu est un incontournable dans la catégorie jeu de mots. Il permet au utilisateur de tester leur connaissances linguistiques et de les enrichir. De même de par sa faible complexité, il est accessible a tous, au grand comme au plus petit.

Nous aborderons dans ce rapport les thèmes suivant :

1. La description général de l'application
2. L'interaction et Données au sein du jeu
3. L'amélioration

2 Description générale de l'application

L'application du pendu a été réalisée sur les environnements de développement d'Android Studio et de Xcode, avec l'appuie de la documentation de chacune des deux plateformes [1] [3] et des transparents de cour mis à disposition.

L'application, disponible en version paysage et portrait sous Android et uniquement en version portrait sous IOS, possède les fonctionnalités classique d'un pendu.

2.1 Contenu

L'application s'ouvre sur l'interface du menu, deux choix s'offre à lui : Jouer ou Thème. Dans l'onglet Thème, l'application donne la possibilité à l'utilisateur de choisir le thème sur lequel il souhaite se défier, tandis que s'il choisit l'onglet Jouer les mots proposés seront de thème aléatoires.



FIGURE 1 – Interface du menu

Le Pendu génère aléatoirement des mots correspondant à un thème désignée. L'application possède huit thèmes différents. Elle est également composée d'un menu principale, d'un menu listant les huit thèmes, d'un menu permettant de jouer et de deux fenêtres pop-up permettant la gestion de la victoire.

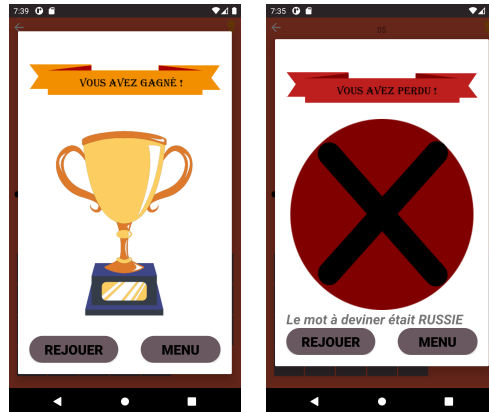


FIGURE 2 – Pop-up de gestion de la victoire

L'application sous Android possède plus de fonctionnalité que celle codé sur Xcode. En effet sous Android, un pop-up supplémentaire s'ajoute à ceux de la gestion de la victoire. Celui-ci correspond à un pop-up d'aide , permettant à l'utilisateur de voir une lettre ou d'obtenir un indice sur le mot en échange de pièce.



FIGURE 3 – Pop-up d'aide

À tout instant de l'application, l'utilisateur à la possibilité de revenir au menu et de changer de thème.

2.2 Design

De nombreuses images illustrent le jeu du Pendu. Du logo, en passant par les illustrations de la liste, à l'état du pendu, tous ces éléments, on été réalisée sur le logiciel de dessin Inskape[4]. Le choix des couleurs a été fait par l'usage d'une palette de couleur via le site Coolor [2].

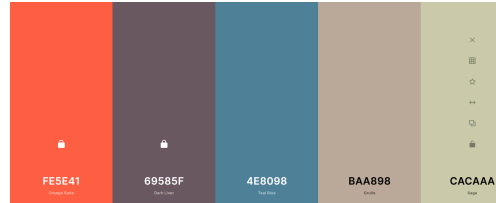


FIGURE 4 – Palette de couleur de l'application

3 Interactions et Données au sein du jeu

Le jeu comporte de nombreuses activités et pop-up qui communiquent entre eux. La communication dans l'application est illustrée ci-dessous.

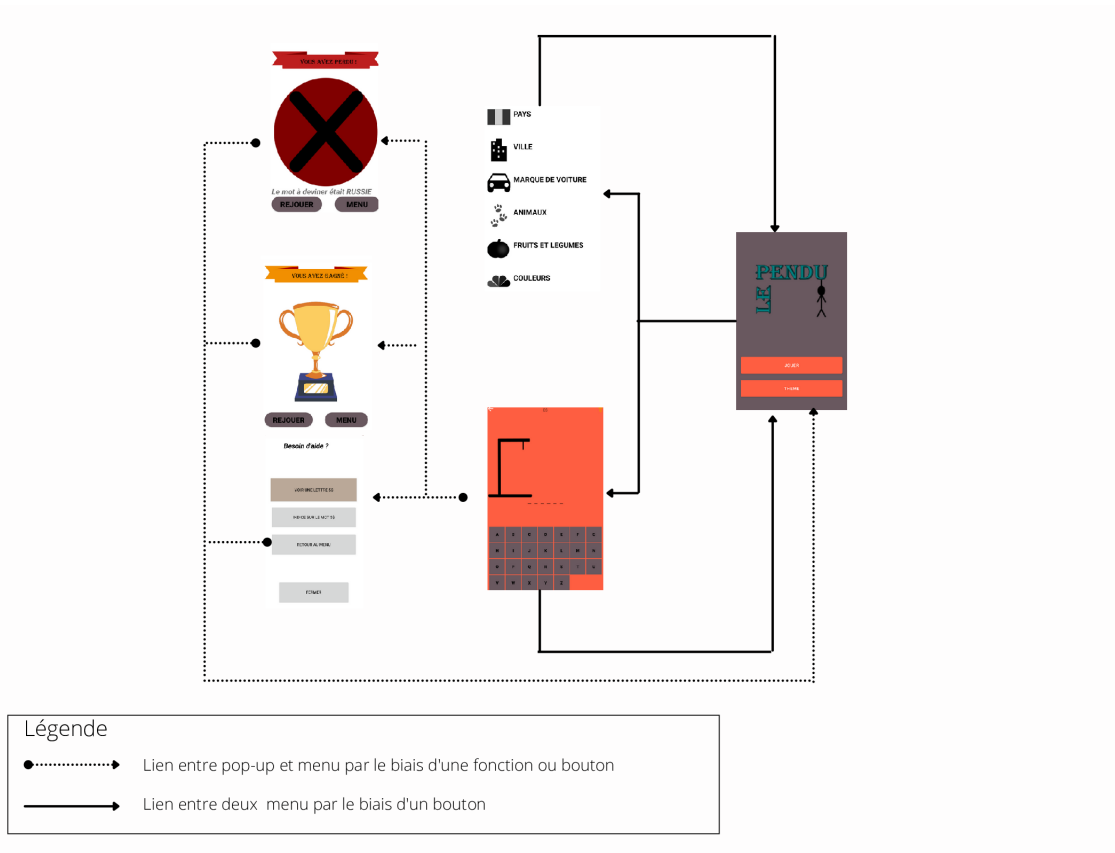


FIGURE 5 – Communication dans l'application

3.1 Android

La communication de données sous Android se fait via l'utilisation d'un Bundle. Dans l'application, selon le thème choisis par l'utilisateur, le menu jeu doit proposer une liste de mots en correspondance avec le thème. Voici quelques extraits de code illustrant la communication entre la liste et le jeu.

1 - Envoie de l'information vers le menu par la liste

```
Intent i = new Intent(this,Activity_ menu.class);
i.putExtra("theme","PAYS.txt");
startActivity(i);
```

2 - Réception de l'information par le menu

```
Bundle word_ array;
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_ menu);
    word_ array = getIntent().getExtras();

    //On récupère le nom du fichier dans lequel se trouve les mots
    if (word_ array != null) myArray = word_ array.getString("theme");
}
```

Sur la demande du menu jeu, les pop-ups de l'application se lancent, que ce soit via une fonction ou par l'intermédiaire d'un bouton.

3 - Lancement du pop-up de gestion de victoire par la demande du menu jeu

```
private void closeWinDialog(Boolean win) {
    Intent j = new Intent(this, Activity_ menu.class);

    if (win ) dialog_ result.setContentView(R.layout.victoire);
    else {
        dialog_ result.setContentView(R.layout.defaite);
        TextView mot = dialog_ result.findViewById(R.id.textView4);
        mot.setText("Le mot à deviner était "+ word);
    } dialog_ result.show();
}
```

3.2 iOS

Sous iOS, la transmission d'information se fait via des Segue.

1- Envoi de l'information vers le menu par la liste

```
func tableView(_ tableView : UITableView, didSelectRowAt indexPath : IndexPath){
    index = indexPath.row
    performSegue(withIdentifier : "theme", sender : nil)
}
override func prepare(for segue : UIStoryboardSegue, sender : Any?) {
    if segue.identifier == "theme" {
        let VCDestination = segue.destination as! MenuViewController
        VCDestination.theme = index
    }
}
```

2 - Réception de l'information par le menu

```
class MenuViewController : UIViewController {
    var theme = 0
    override func viewDidLoad() {
        super.viewDidLoad()
    }
    override func prepare(for segue : UIStoryboardSegue, sender : Any?) {
        if segue.identifier == "jouer"{
            let VCDestination = segue.destination as! ViewController
            VCDestination.Theme = theme } }
    @IBAction func infoTheme(_ sender : UIButton) {
        performSegue(withIdentifier : "jouer", sender : nil)
    } }
}
```

4 Amélioration

L'application demandée devait être conçue dans un délai imparti, c'est pourquoi certaines parties du jeu nécessitent encore des améliorations. Pour commencer, l'implémentation d'une base de données, pour amoindrir le nombre d'éléments dans le jeu et faciliter la lisibilité du code. Une fonctionnalité de changement de langue a aussi été prévue, mais n'a pas eu le temps de voir le jour. Et enfin la mise en œuvre de différents niveaux, comme par exemple un niveau chronométré ou un niveau en anglais pour tester sa culture linguistique. De même, une amélioration des contraintes sur les objets de l'application, et la création de la version paysage de l'application sur Xcode.

5 Conclusion

En conclusion, le développement de cette application a été pour moi très enrichissant. Je suis satisfaite de ce que j'ai pu accomplir sur les deux plateformes même si des améliorations peuvent être apportées.

Références

- [1] Android studio.
- [2] Coolors.
- [3] Xcode.
- [4] Xcode.