

Rapport de l'application Jeu de math

ISSOUFFA Ambdoulouhi Ibnou et MEHONG SHIT LI Matthieu, L3 informatique

1^{er} mai 2021

Résumé

Dans ce projet d'application mobile, nous verrons l'intégration d'une application android et de sa version ios. Ces deux application sont tout deux responsive. Notament , une nouvelle manière d'utiliser les canvas.

1 Introduction

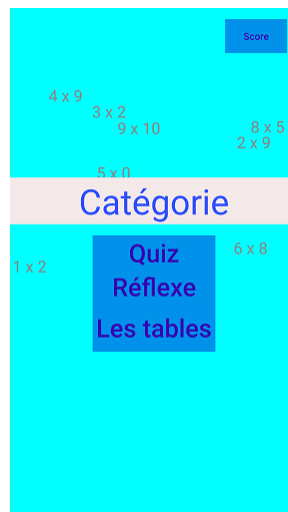
Nous avons tout fait du calcul mental durant notre enfance. Pour la plus part d'entre nous ce n'était pas vraiment agréable. Aujourd'hui, nous vous proposons notre application. Une application mobile, qui permet d'apprendre à retenir ses tables de multiplication en apprenant tout en jouant. Nous allons découvrir ensemble ce que nous avons codé.

2 Description générale de l'application

Vous avons 8 ecrans en tous :

1. Accueil
2. Catégorie
3. Quiz
4. Réflexe
5. Table de Multiplication
6. Score
7. Question du Quiz
8. Resultat au Quiz

Voici une capture d'écran de la page **catégorie** :

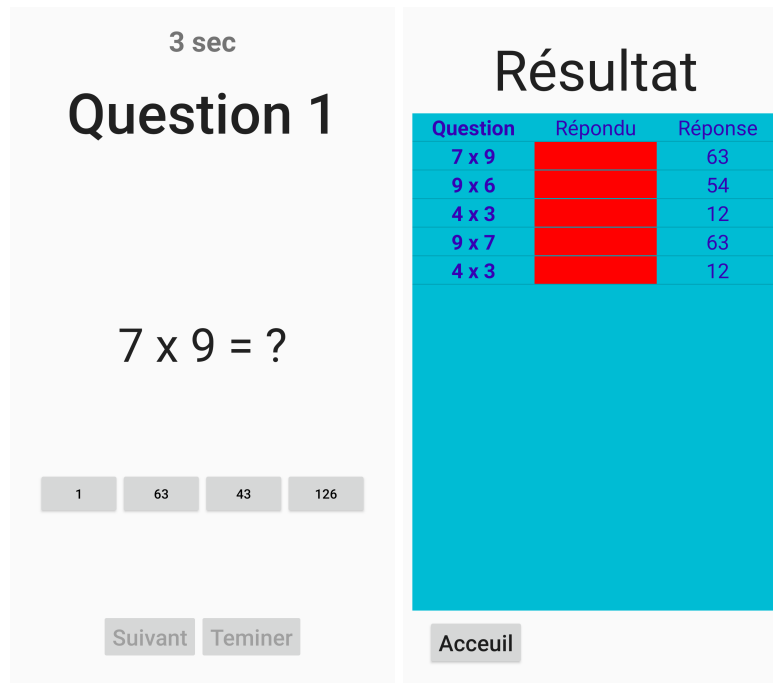


Sur cette page, nous pouvons accéder à :

- la page du **quiz** de calcul mental
- la page des score au jeu de **réflexe**
- la page du jeu de **réflexe**
- la page des **tables** de multiplication

3 Intégration des fonctionnalités

3.1 Le jeu de Quiz



Le jeu de quiz est la seule fonctionnalité qui est disponible sur les deux plateformes. Les questions sont défilées sur un seul écran. Cet écran est mis à jour à chaque fois que nous cliquons sur le bouton **suivant**. Il y a 5 questions en tout. L'utilisateur a 5 secondes par question. À la fin du quiz, une page **résultat** affiche les bonnes réponses de l'utilisateur.

3.1.1 Android

- Le timer, vient de cette page [1], et il a été adapté au code.
- Les boutons sauvegardent la dernière sélection de l'utilisateur en changeant de couleur.
- Les colonnes de répondu sont en vert si la réponse de l'utilisateur est correcte sinon elle est en rouge.
- La page résultat intègre les ListViews
- Les réponses aléatoires sont générées par l'algorithme `setAnswerList` ci-dessus :

```
private fun setAnswerList() {
    val list = ArrayList<Int>()
    val max = if (calcul.answer!=0) calcul.answer + calcul.answer / 2 else 10
    var piege: Int
    for (i in 1..2) {piege=getRandInt(max); list.add(piege)} // 2 réponses pièges
    piege = if (calcul.answer!=0) 2*calcul.answer else getRandInt(10)
    list.add(getRandInt(list.size-1),piege) // réponse piège
    list.add(getRandInt(list.size-1), calcul.answer) // la réponse
}
```

```

        answerList = list
    }

```

3.1.2 iOS

- Le timer, vient de cette page [2], et il a été adapté au code.
- Les boutons sauvegardent la dernière sélection de l'utilisateur car ce sont des `segmentController`.
- Les colonnes de réponse sont en vert si la réponse de l'utilisateur est correcte sinon elle est en rouge.
- La page résultat intègre les `TableView`
- Les réponses aléatoires sont générées par l'algorithme `setAnswerList` ci-dessus :

```

func fillListRep()-> Array<Int>{
    var l:[Int]=[]
    let max=(answer != 0) ? (answer+answer)/2 : 10
    var piege:Int
    for _ in 1...2 {
        piege=Int.random(in: 0...max)
        l.append(piege)
    }
    piege=(answer != 0) ? 2*answer : Int.random(in: 0...10)

    l.insert(piege, at: Int.random(in: 0...l.count-1))
    l.insert(answer, at: Int.random(in: 0...l.count-1))
    return l
}
}

```

4 Quelques points délicats/intéressants

5 Conclusion

Références

[1] <https://www.tutorialspoint.com/how-to-set-a-timer-in-android-using-kotlin>.

[2] <https://learnappmaking.com/timer-swift-how-to/>.