

Titre : Boule Magique

Auteur (projet) : Stephen Fafournoux

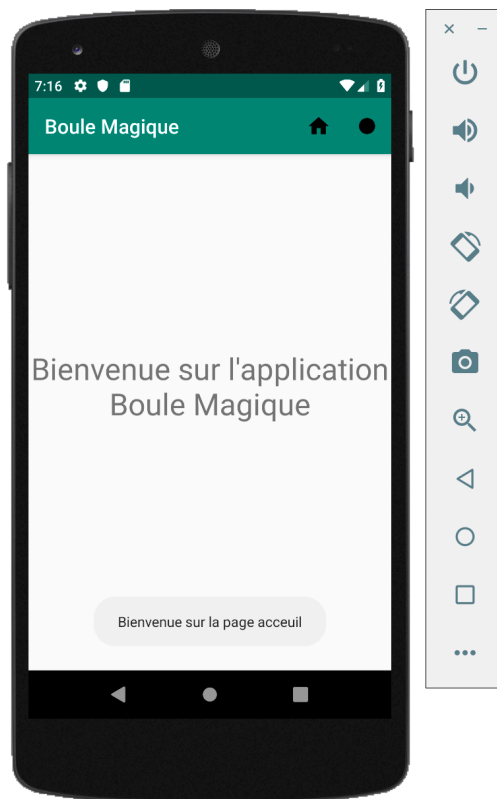
Auteur (readme) : Stephen Fafournoux & Nicolas Rochette

Résumé projet :

Ce projet permet de mettre en oeuvre tout ce que l'on a vu durant les cours en développement une application qui met en oeuvre différentes fonctionnalités.

Description de l'application:

Cette application permet de choisir la vitesse de la balle qui va être affichée, un menu est proposé pour accéder aux différentes pages.



Cette capture d'écran est la page d'accueil de l'application avec le menu en haut. Un message s'affiche lorsqu'on arrive sur la page.

Ici on utilise AnkoLogger pour afficher le messages toast à l'écran et informer sur quelle page on se trouve.

AppCompatActivity permet de créer de nouvelles activity et passer d'une page à une autre.

```

package com.example.stephen.projet1

import ...

class MainActivity : AppCompatActivity(), AnkoLogger {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
    }

    override fun onCreateOptionsMenu(menu: Menu?): Boolean {
        val inflater = menuInflater
        inflater.inflate(R.menu.menu, menu)
        return true
    }

    override fun onOptionsItemSelected(item: MenuItem?): Boolean {
        return when(item?.itemId){
            R.id.BouleMagique -> {
                toast("Bienvenue sur la page boule magique")
                var intent = Intent( packageContext: this, NumberPicker::class.java)
                finishActivity( requestCode: 0)
                startActivity(intent)
                return true
            }
            else -> {print("Bonjour")
                true
            }
        }
    }
}

```

Ici on voit que l'on accède à une autre page via un bouton du menu en haut.

```

package com.example.stephen.projet1

import ...

class NumberPicker : AppCompatActivity() {
    companion object {
        var number : Int = 1
    }

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.number_picker)

        number_picker.minValue = 1
        number_picker.maxValue = 5
        number_picker.wrapSelectorWheel = false
        var button = findViewById<Button>(R.id.Valider)
        number_picker.setOnValueChangedListener { picker, oldVal, newVal ->
            //Display the newly selected number to text view
            text_view.text = "Vitesse de la balle : $newVal "
        }
        button.setOnClickListener(object: View.OnClickListener {
            override fun onClick(view: View): Unit {
                // Handler code here.
                var intent = Intent( packageContext: this@NumberPicker, ActivityBall::class.java)
                number = number_picker.value
                startActivity(intent)
            }
        })
    }

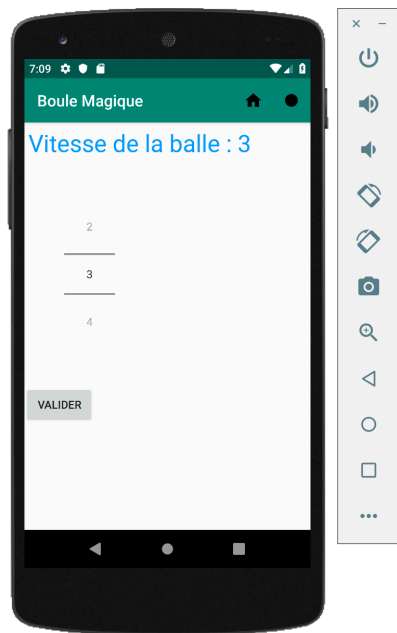
    override fun onCreateOptionsMenu(menu: Menu?): Boolean {
        val inflater = menuInflater
        inflater.inflate(R.menu.menu, menu)
        return true
    }

    override fun onOptionsItemSelected(item: MenuItem?): Boolean {
        return when(item?.itemId){
            R.id.Accueil -> {
                toast("Bienvenue sur la page accueil")
                var intent = Intent( packageContext: this, MainActivity::class.java)
                finishActivity( requestCode: 0)
                startActivity(intent)
                return true
            }
            else -> {print("Bonjour")
                true
            }
        }
    }
}

```

On peut voir ici l'initialisation du numberPicker lors de la création de la deuxième activity. Un compagnon object permet de récupérer la valeur du numberPicker pour les prochaines activity. Ainsi, on pourra utiliser la valeur du numberPicker dans les autre activity.

Voici la seconde page qui permet de sélectionner la vitesse de la balle qui sera afficher dans la page suivante. Après avoir cliqué sur le bouton valider, la balle s'affiche avec la vitesse correspondante.



Cette capture d'écran représente la classe pour construire le cercle.

```
package com.example.stephen.projet1

import ...

data class MagicCircle(val maxX: Float, val maxY: Float, val vitesse: Int) : AppCompatActivity(), AnkoLogger {
    var cx: Float = 50F
    var cy: Float = 50F
    val rad: Float = 40F
    var delta = 7 * vitesse
    var dx = delta
    var dy = delta

    fun move(){
        when {
            cx < 0F -> dx = delta
            cx > maxX -> dx = -delta
            cy < 0F -> dy = delta
            cy > maxY -> dy = -delta
        }
        cx += dx
        cy += dy
    }
}
```

```

package com.example.stephen.projet1

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.view.View
import android.widget.NumberPicker

class BouleMagique : AppCompatActivity() {
    lateinit var mCircle : MagicCircle

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_boule_magique)
        mCircle = MagicCircle(width.toFloat(), height.toFloat(), NumberPicker.number)
    }

    constructor(context : Context) : this(context, null)

    constructor(context: Context?, attrs: AttributeSet?) : super(context, attrs){
        init()
    }
    private var mPaint = Paint()

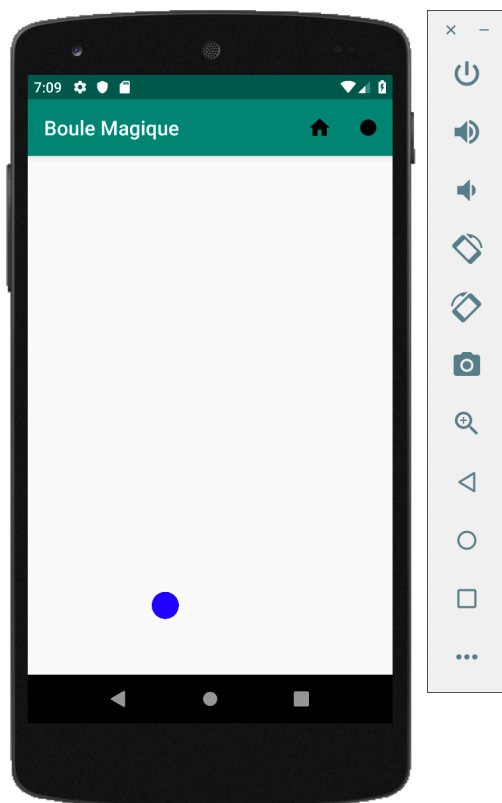
    private fun init(){
        mPaint.color = Color.BLUE
    }

    override fun onDraw(canvas: Canvas?){
        super.onDraw(canvas)
        mCircle.move()
        canvas?.drawCircle(mCircle.cx, mCircle.cy, mCircle.rad, mPaint)
        invalidate()
    }
}

```

Voici la capture d'écran du code qui construit le cercle et qui l'affiche. On peut voir que la vitesse du number picker est reprise pour construire le cercle.

Voici la page qui affiche le cercle.



Remarque sur le code en général

Les fragments ne sont pas présents pour faire le menu latéral. Mais cette application m'a permis de mieux comprendre comment fonctionne kotlin.

Remarque sur l'application en général

Cette application est assez simple, mais elle permet de voir les différents points demandés dans le cahier des charges hors mis les fragments. (Fonctionnalité de la bibliothèque anko, élément graphique et cercle à l'écran).

Les différentes activity peuvent aussi accéder aux données d'une autre via un companion object.

Le nom des variables et des classes sont pertinents et le code est bien lisible.