



Piscine iOS Swift - Day 06

MotionCube

Résumé: Ce document contient le sujet du Day 06 de la piscine iOS Swift de [42](#)

Table des matières

I	Préambule	2
II	Consignes	4
III	Introduction	5
IV	Exercice 00 : TapGesture	6
V	Exercice 01 : Dynamic Behavior	7
VI	Exercice 02 : Gestures	9
VII	Exercice 03 : CoreMotion	11

Chapitre I

Préambule

Voici le monde de **Poincaré** tiré du livre *Science et l'hypothèse* chapitre 4 :

Supposons, par exemple, un monde renfermé dans une grande sphère et soumis aux lois suivantes :

La température n'y est pas uniforme ; elle est maxima au centre, et elle diminue à mesure qu'on s'en éloigne, pour se réduire au zéro absolu quand on atteint la sphère où ce monde est renfermé.

Je précise davantage la loi suivant laquelle varie cette température. Soit R le rayon de la sphère limite ; soit r la distance du point considéré au centre de cette sphère. La température absolue sera proportionnelle à $R^2 - r^2$.

Je supposerai de plus que, dans ce monde, tous les corps aient même coefficient de dilatation, de telle façon que la longueur d'une règle quelconque soit proportionnelle à sa température absolue.

Je supposerai enfin qu'un objet transporté d'un point à un autre, dont la température est différente, se met immédiatement en équilibre calorifique avec son nouveau milieu.

Rien dans ces hypothèses n'est contradictoire ou inimaginable.

Un objet mobile deviendra alors de plus en plus petit à mesure qu'on se rapprochera de la sphère limite.

Observons d'abord que, si ce monde est limité au point de vue de notre géométrie habituelle, il paraîtra infini à ses habitants.

Quand ceux-ci, en effet, veulent se rapprocher de la sphère limite, ils se refroidissent et deviennent de plus en plus petits. Les pas qu'ils font sont donc aussi de plus en plus petits, de sorte qu'ils ne peuvent jamais atteindre la sphère limite.

Si, pour nous, la géométrie n'est que l'étude des lois suivant lesquelles se meuvent les solides invariables, pour ces êtres imaginaires, ce sera l'étude des lois suivant lesquelles se meuvent les solides déformés par ces différences de température dont je viens de parler.

Sans doute, dans notre monde, les solides naturels éprouvent également des variations de forme et de volume dues à l'échauffement ou au refroidissement. Mais nous négligeons ces variations en jetant les fondements de la géométrie ; car, outre qu'elles sont très faibles, elles sont irrégulières et nous paraissent par conséquent accidentelles.

Dans ce monde hypothétique, il n'en serait plus de même, et ces variations suivraient des lois régulières et très simples.

D'autre part, les diverses pièces solides dont se composerait le corps de ses habitants, subiraient les mêmes variations de forme et de volume.

Je ferai encore une autre hypothèse ; je supposerai que la lumière traverse des milieux diversement réfringents et de telle sorte que l'indice de réfraction soit inversement pro-

portionnel à $R^2 - r^2$. Il est aisé de voir que, dans ces conditions, les rayons lumineux ne seraient pas rectilignes, mais circulaires.

Pour justifier ce qui précède, il me reste à montrer que certains changements survenus dans la position des objets extérieurs peuvent être corrigés par des mouvements corrélatifs des êtres sentants qui habitent ce monde imaginaire; et cela de façon à restaurer l'ensemble primitif des impressions subies par ces êtres sentants.

Supposons en effet qu'un objet se déplace, en se déformant, non comme un solide invariable, mais comme un solide éprouvant des dilatations inégales exactement conformes à la loi de température que j'ai supposée plus haut. Qu'on me permette pour abréger le langage, d'appeler un pareil mouvement déplacement non euclidien.

Si un être sentant se trouve dans le voisinage, ses impressions seront modifiées par le déplacement de l'objet, mais il pourra les rétablir en se mouvant lui-même d'une manière convenable, il suffit que finalement l'ensemble de l'objet et de l'être sentant, considéré comme formant un seul corps, ait éprouvé un de ces déplacements particuliers que je viens d'appeler non euclidiens. Cela est possible si l'on suppose que les membres de ces êtres se dilatent d'après la même loi que les autres corps du monde qu'ils habitent.

Bien qu'au point de vue de notre géométrie habituelle les corps se soient déformés dans ce déplacement et que leurs diverses parties ne se retrouvent plus dans la même situation relative, cependant nous allons voir que les impressions de l'être sentant sont redevenues les mêmes.

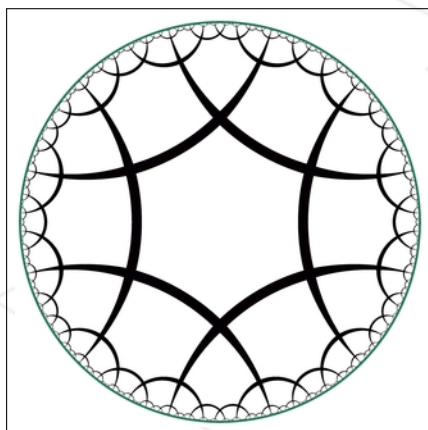
En effet, si les distances mutuelles des diverses parties ont pu varier, néanmoins les parties primitivement en contact sont revenues en contact. Les impressions tactiles n'ont donc pas changé.

D'autre part, en tenant compte de l'hypothèse faite plus haut au sujet de la réfraction et de la courbure des rayons lumineux, les impressions visuelles seront aussi restées les mêmes.

Ces êtres imaginaires seront donc comme nous conduits à classer les phénomènes dont ils seront témoins et à distinguer parmi eux, les « changements de position » susceptibles d'être corrigés par un mouvement volontaire corrélatif.

S'ils fondent une géométrie, ce ne sera pas comme la nôtre, l'étude des mouvements de nos solides invariables; ce sera celle des changements de position qu'ils auront ainsi distingués, et qui ne sont autres que les « déplacements non euclidiens », ce sera la géométrie non Euclidienne.

Ainsi des êtres comme nous, dont l'éducation se ferait dans un pareil monde, n'auraient pas la même géométrie que nous.



Chapitre II

Consignes

- Seule cette page servira de référence : ne vous fiez pas aux bruits de couloir.
- Lisez attentivement l'intégralité du sujet avant de commencer.
- Le sujet peut changer jusqu'à une heure avant le rendu.
- Vos exercices seront corrigés par vos camarades de piscine.
- Le sujet fait foi, ne vous fiez pas toujours à la lettre aux demos qui peuvent contenir des ajouts supplémentaires non demandés.
- Vous devrez rendre une app par jour (sauf pour le Day 01) sur votre depot git, rendez le dossier du projet Xcode.
- Voici le manuel officiel de [Swift](#) et de [Swift Standard Library](#)
- Il est interdit d'utiliser d'autres librairies, packages, pods... avant le Day 07
- Vous avez une question ? Demandez à votre voisin de droite. Sinon, essayez avec votre voisin de gauche.
- Pensez à discuter sur le forum Piscine de votre Intra !
- Réfléchissez. Par pitié, par Odin ! Nom d'une pipe.



Les videos sur l'intra ont été tournées avant Swift 3. Enlevez le prefix "NS" que vous voyez devant les class/struct/function dans le code des videos pour les utiliser en Swift 3.



L'intra indique la date et l'heure de fermeture de vos dépôts. Cette date et heure correspond également au début de la période de peer-evaluation pour le jour de piscine correspondant. Cette période de peer-evaluation dure exactement 24h. Une fois ces 24h passées, vos notes peer manquantes seront complétées par des 0.

Chapitre III

Introduction

Avez vous déjà entendu "Mets pas tes doigts sur l'écran !" ? Bien sûr aujourd'hui cette phrase n'a plus de sens. La disparition des claviers au profit des écrans tactiles permet aux applications d'étendre leurs interfaces et de ne plus être limitées par les touches du clavier. En effet, lorsqu'on joue au snake sur un nokia 3310, on a juste besoin des 4 touches pour diriger le serpent, les autres sont tout à fait inutiles.


Apple à aussi équipé ses devices de différents capteurs comme un accéléromètre, un gyroscope, un capteur de proximité ou encore un baromètre.

Ainsi on peut récupérer des inputs utilisateur ou des données d'environnement grâce à l'écran et les capteurs du device.

Aujourd'hui vous allez apprendre à utiliser les **UIGestureRecognizer**, pour récupérer les actions de l'utilisateur sur l'écran, et **CoreMotion** pour l'orientation du device dans une application qui va vous faire jouer avec des petits carrés et des petits cercles. Ces formes seront soumises aux lois la physique comme la gravité, l'élasticité et les collisions en utilisant un **UIDynamicAnimator**.

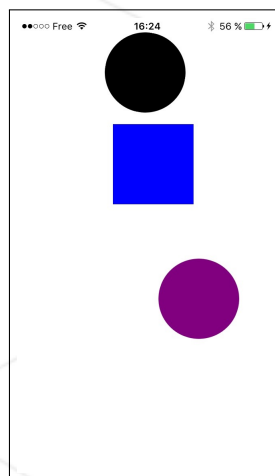
Chapitre IV

Exercice 00 : TapGesture

	Exercice : 00
TapGesture	
Fichiers à rendre : .xcodeproj et tous les fichiers necessaires	
Fonctions Autorisées : Swift Standard Library, UIKit	
Remarques : n/a	


Le but de cet exercice est de pouvoir ajouter une forme dans une vue vide en touchant l'écran. Pour cela vous allez créer une class qui hérite de la class **UIView** qui vas représenter une forme.

- Sa forme doit être soit un carré, soit un cercle.
- Sa taille doit être de 100 x 100.
- Sa forme et sa couleur doivent être aléatoires.
- Lorsqu'on touche l'écran, la forme doit apparaitre sous notre doigt.



Chapitre V

Exercice 01 : Dynamic Behavior

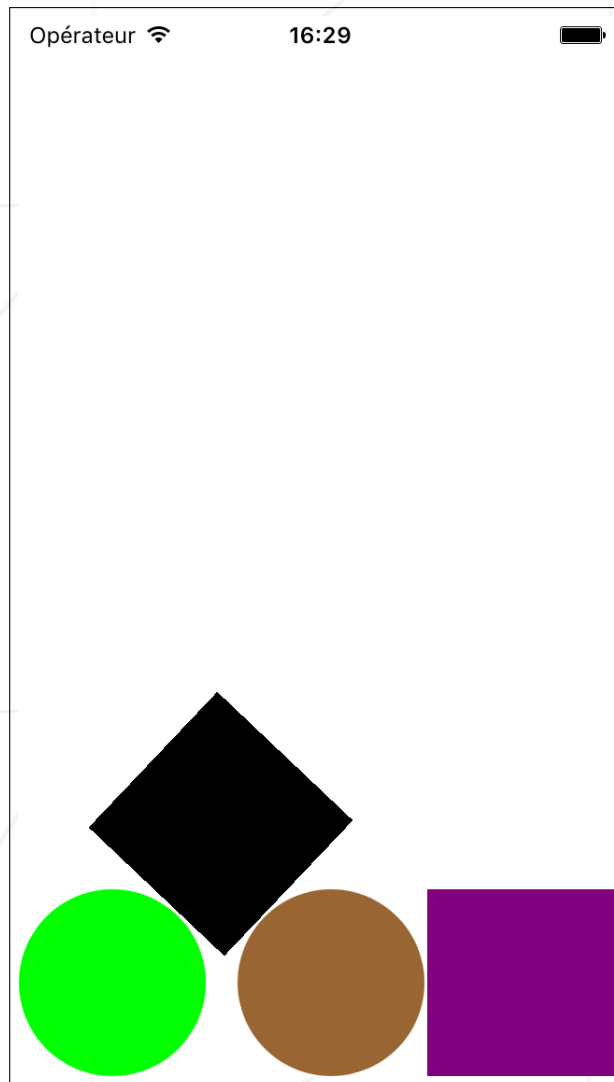
	Exercice : 01
Dynamic Behavior	
Fichiers à rendre : .xcodeproj et tous les fichiers nécessaires	
Fonctions Autorisées : Swift Standard Library, UIKit	
Remarques : n/a	

Ajoutez un peu de physique à vos formes. Elles doivent être soumises à plusieurs **UIDynamicBehavior** :

- Tomber sous l'effet de la gravité.
- Se percuter et ne doivent pas sortir de leur "superview".
- Avoir un effet d'élasticité, elles doivent rebondir.




Les formes doivent rebondir sur les bords de la vue. Elles ne doivent pas disparaître.



Chapitre VI

Exercice 02 : Gestures

	Exercice : 02
Gestures	
Fichiers à rendre : .xcodeproj et tous les fichiers nécessaires	
Fonctions Autorisées : Swift Standard Library, UIKit	
Remarques : n/a	

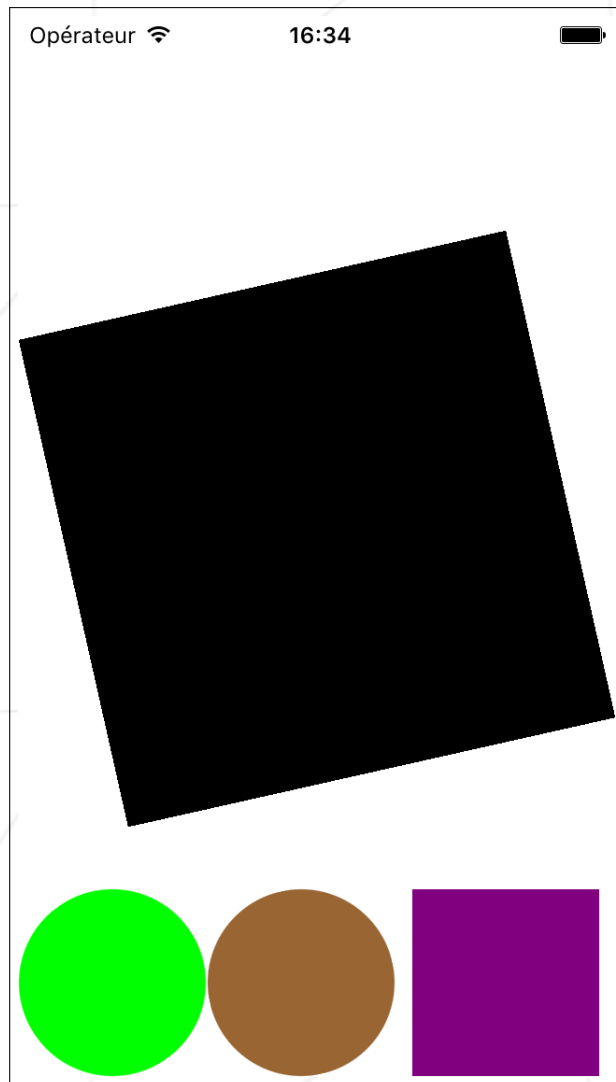
Interagir avec ces formes serait super fun !

Ajoutez maintenant des **Gesture** à vos formes :

- **UIPanGestureRecognizer** pour les déplacer.
- **UIPinchGestureRecognizer** pour les agrandir et les réduire.
- **UIRotationGestureRecognizer** pour les tourner.




Lorsque vous interagissez avec une forme, vous pouvez supprimer son effet de gravité mais pas de collision et d'élasticité. Quand vous relâchez, l'objet doit être à nouveau soumis à la gravité.



Chapitre VII

Exercice 03 : CoreMotion

	Exercice : 03
CoreMotion	
Fichiers à rendre : .xcodeproj et tous les fichiers necessaires	
Fonctions Autorisées : Swift Standard Library, UIKit, CoreMotion	
Remarques : n/a	

Dans les réglages du projet, n'autorisez que le mode Portrait.

Vous allez maintenant modifier le vecteur gravité en fonction des données de l'accéléromètre en utilisant **CoreMotion**. Les formes doivent tomber dans différentes direction en fonction de l'orientation dans l'espace du device.



Le simulateur iOS ne permet pas encore de simuler les données des capteurs donc il est conseillé de tester cette application sur un vrai device apple. Si vous n'en possédez pas, ne paniquez pas, demandez à vos voisins de vous en prêter un. Si malgré tout vous n'arrivez pas à trouver de device apple, ne vous inquiétez pas, cet exercice est assez simple et la correction n'en tiendra pas rigueur.