

## Piscine iOS Swift - Day 06

### 모션큐브

요약: 이 문서에는 [42](#)의 iOS Swift piscine에 대한 Day 06의 주제가 포함되어 있습니다.

# 내용물

I	전문	2
II	지침	4
III	소개	5
IV 연습 00 :	TapGesture	6
V 연습 01:	동적 동작	7
VI 연습 02:	제스처	9
VII 연습 03:	CoreMotion	11

# 1장

## 전문

다음은 Science et l'hypothèse 책 4장에 나오는 Henry Poincaré의 세계입니다.

예를 들어, 큰 구체로 둘러싸여 있고 다음 법칙이 적용되는 세계를 가정해 보십시오. 온도는 균일하지 않습니다. 그것은 중심에서 최대이고, 우리가 이 세계를 둘러싸고 있는 구체에 도달할 때 절대 0으로 감소하기 위해 우리가 그것으로부터 멀어질수록 감소합니다.

나는 이 온도를 변화시키는 법칙을 더 명확히 한다.  $R$ 을 경계 구의 반경이라고 하자.  $r$ 은 이 구의 중심으로 간주되는 점으로부터의 거리입니다. 절대 온도는  $R^2 - r^2$ 에 비례합니다.

나는 더 나아가 이 세상에서 모든 물체가 같은 팽창 계수를 가지고 있다고 가정할 것입니다. 따라서 규칙의 길이는 절대 온도에 비례합니다.

나는 마침내 어떤 물체가 한 지점에서 다른 지점으로 옮겨졌다고 가정할 것이다. 온도가 다르면 즉시 새로운 환경과 균형을 이룹니다.

이러한 가정의 어떤 것도 모순되거나 상상할 수 없는 것이 아닙니다. 파 A 움직이는 물체 그런 다음 우리가 이동함에 따라 점점 작아질 것입니다. 한계 구에 접근할 것입니다.

이 세계가 우리의 일반적인 기하학의 관점에서 제한된다면 그곳의 거주자들에게 무한하게 보일 것이라는 점을 먼저 관찰합니다. 파 실제로 그들은 한계 구체에 가까워지면 냉각되어 점점 작아집니다. 따라서 그들이 취하는 단계도 점점 작아지므로 결코 한계 구에 도달할 수 없습니다. 파 우리에게 기하학이 불변의 고체가 움직이는 법칙에 대한 연구일 뿐이라면, 이 가상의 존재들에게는 방금 언급한 온도차에 의해 변형된 고체가 움직이는 법칙에 대한 연구가 될 것입니다. . par 의심할 여지 없이, 우리 세계에서 천연 고체도 가열 또는 냉각으로 인해 모양과 부피의 변화를 경험합니다. 그러나 우리는 기하학의 기초를 놓음으로써 이러한 변화를 무시합니다. 왜냐하면 그들은 매우 약한 것 외에도 불규칙하고 따라서 우리에게 우연한 것처럼 보이기 때문입니다. par 이 가상의 세계에서는 더 이상 동일하지 않으며 이러한 변형은 규칙적이고 매우 단순한 법칙을 따릅니다.

한편, 거주자의 몸을 구성하는 다양한 단단한 부분, 모양과 부피에서 동일한 변화를 겪을 것입니다.

나는 또 다른 가설을 세울 것이다. 나는 빛이 다양한 굴절 매체를 가로질러 굴절률이  $R^2 - r^2$ 에 반비례한다고 가정할 것입니다. 이러한 조건에서 광선은 직선이 아니라 원형임을 쉽게 알 수 있습니다.

위의 내용을 정당화하기 위해, 나는 여전히 외부 대상의 위치에 있는 특정 변화가 여기에 거주하는 중생의 상관적 움직임에 의해 수정될 수 있음을 보여야 합니다.

상상의 세계; 그리고 이것은 이 중생들이 경험한 원시적인 인상을 회복하기 위한 것입니다.

실제로 물체가 변형에 의해 불변의 고체가 아니라 내가 위에서 가정한 온도 법칙에 정확히 일치하는 불균등한 팽창을 경험하는 고체로 움직인다고 가정합니다. 그런 운동을 비유클리드 변위라고 부르기 위해 언어를 줄여보겠습니다.

중생이 근처에 있으면 대상의 이동에 의해 인상이 수정되지만 적절한 방법으로 움직여 대상 전체와 중생의 인상을 회복할 수 있습니다. , 단일 물체를 형성하는 것으로 간주되는 는 내가 방금 비유클리드라고 부르는 이러한 특정 변위 중 하나를 마침내 경험해야 했습니다. 이것은 이 존재의 구성원이 그들이 살고 있는 세계의 다른 신체와 동일한 법칙에 따라 확장된다고 가정하면 가능합니다.

우리의 일반적인 기하학의 관점에서 볼 때 신체는 이러한 변위에서 변형되고 신체의 다양한 부분은 더 이상 동일한 상대적 상황에서 발견되지 않지만, 우리는 중생의 인상이 동일하게 된 것을 보게 될 것입니다.

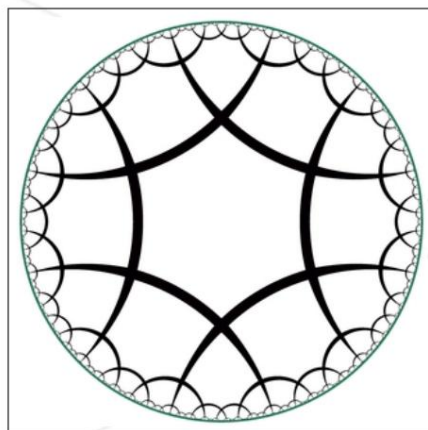
실제로, 여러 부분의 상호 거리가 다를 수 있었다면 그럼에도 불구하고 원래 접촉한 당사자는 다시 접촉했습니다. 그래서 측각적인 인상은 변하지 않았습니다.

반면에 광선의 굴절과 곡률에 대해 위에서 가정한 것을 고려하면 시각적 인상도 동일하게 유지됩니다.

그러므로 이 상상의 존재들은 우리와 마찬가지로 그들이 목격하게 될 현상을 분류하고 그들 사이에서 상관적 자발적 운동에 의해 고정될 수 있는 "위치의 변화"를 구별하도록 이끌릴 것입니다.

만일 그들이 기하학을 발견했다면 그것은 우리의 불변 입체의 움직임에 대한 연구와 같지 않을 것입니다. 그들이 그렇게 구별하게 될 위치의 변화의 변화일 것이고, 다른 아닌 "비-유클리드 변위"일 것이고, 그것은 비-유클리드 기하학이 될 것입니다.

따라서 그러한 세계에서 교육받을 우리와 같은 존재는 우리와 같은 기하학을 갖지 않을 것입니다.



# 제2장

## 지침

명시적으로 모순되지 않는 한 다음 지침은 이 폴의 모든 날짜에 유효합니다.

- 이 주제는 참고용으로만 사용하십시오. 복도의 소문에 의존하지 마십시오.
- 주제는 렌더링 1시간 전까지 변경될 수 있습니다.
- 운동은 가장 간단한 것부터 가장 복잡한 것까지 매우 정확하게 정렬되어 있습니다.  
더 간단한 운동이 완벽하게 성공하지 못한다면 어떤 경우에도 우리는 주의를 기울이거나 복잡한 운동을 고려하지 않을 것입니다.
- 파일과 디렉토리의 권한에 주의하십시오.
- 모든 연습에 대해 렌더링 절차를 따라야 합니다. 오늘의 GIT 저장소 URL은 인트라넷에서 확인할 수 있습니다.
- 당신의 운동은 폴 메이트에 의해 평가될 것입니다.
- 동료 외에도 la Moulinette라는 프로그램을 통해 평가를 받을 수 있습니다. La Moulinette는 완전히 자동화되어 있기 때문에 채점에 매우 엄격합니다. 따라서 그녀와 그녀의 등급에 대해 논의하는 것은 불가능합니다. 불쾌한 놀라움을 피하기 위해 나무랄 데 없는 엄격함을 유지하십시오.
- 셸 연습은 /bin/sh와 함께 실행해야 합니다.
- 회사에서 명시적으로 지정한 파일 이외의 파일을 남겨두어서는 안 됩니다.  
렌더링 저장소의 연습 문.
- 질문이 있습니까? 오른쪽 이웃에게 물어보세요. 그렇지 않으면 다음을 시도하십시오.  
왼쪽에 있는 당신의 이웃.
- 기술적인 질문에 대한 모든 답변은 설명서 또는 인터넷.
- Intra 및 Slack의 Pool 포럼에서 토론하는 것을 잊지 마십시오!
- 예를 식별하는 데 도움이 될 수 있으므로 주의 깊게 읽으십시오.  
첫눈에 주제에 지정되지 않은 작업.
- 생각한다. 제발, 토르로, 오딘으로!

## 제3장

### 소개

"화면을 만지지 마십시오!"라고 말한 사람이 있습니까? 물론 오늘날 이 경고는 쓸모가 없습니다. 이제 키보드는 쓸모가 없으며 키보드의 단순한 사용을 넘어 응용 프로그램이 인터페이스를 확장하는 데 도움이 되는 터치 스크린으로 대체되었습니다. 실제로 Nokia 3310에서 Snake를 플레이하려면 4개의 키만 있으면 됩니다. 나머지는 단순히 쓸모가 없습니다.


Apple은 또한 가속도계, 자이로스코프, 근접 센서와 기압계.

따라서 장치는 다음 덕분에 다른 사용자 입력 또는 주변 데이터를 받을 수 있습니다. 화면과 장치의 다양한 센서.

오늘은 UIGestureRecognizer를 사용하여 화면에서 사용자의 동작을 수집하고 사각형과 원으로 장난감을 만드는 응용 프로그램에서 장치 방향에 대한 CoreMotion을 사용하는 방법을 배웁니다. 이러한 모양은 UIDynamicAnimator를 사용하여 중력, 탄성 및 충돌과 같은 물리적 법칙의 적용을 받습니다.

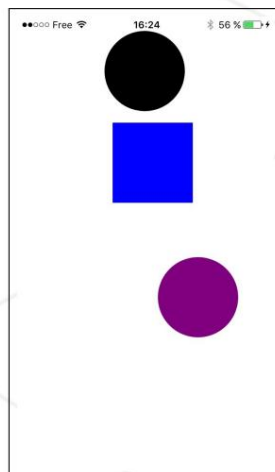
## 제4장

### 연습 00 : TapGesture

	운동: 00
탭 제스처	
제출 파일 : Swift Standard Library, UIKit	
승인된 기능: 해당 없음	
참고: 해당 사항 없음	


이 연습은 화면을 터치하는 빈 보기에 모양을 추가하는 것을 목표로 합니다. 이렇게 하려면 모양을 나타내는 UIView 클래스를 상속하는 클래스를 만듭니다.

- 정사각형 또는 원이어야 합니다.
- 크기는 100 x 100이어야 합니다.
- 모양과 크기는 임의적이어야 합니다.
- 화면을 터치할 때 모양이 손가락 끝 아래에 튀어나와야 합니다.



## 제5장

### 연습 01: 동적 동작

	운동: 01
동적 행동	
제출 파일 : Swift Standard Library, UIKit	
승인된 기능: 해당 없음	
참고: 해당 사항 없음	

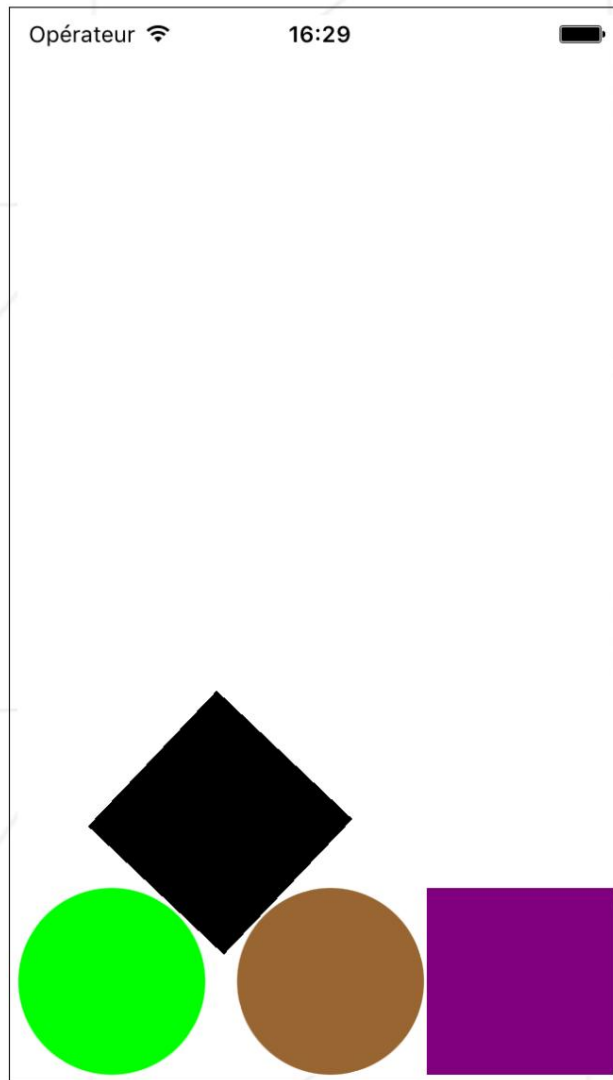
모양에 약간의 물리학을 추가하십시오. 그것들은 여러 UIDynamicBehavior를 따라야 합니다.

- 중력을 받으면 넘어집니다.
- 그들의 "감속관"을 벗어나지 않고 충돌합니다.
- 탄성이 있으므로 바운스해야 합니다.




모양은 보기의 가장자리에서 바운스되어야 합니다. 그들은 사라지지 않아야 합니다.





## 제6장

### 연습 02: 제스처

	운동: 02
제스처	
제출 파일 : Swift Standard Library, UIKit	
승인된 기능: 해당 없음	
참고: 해당 사항 없음	

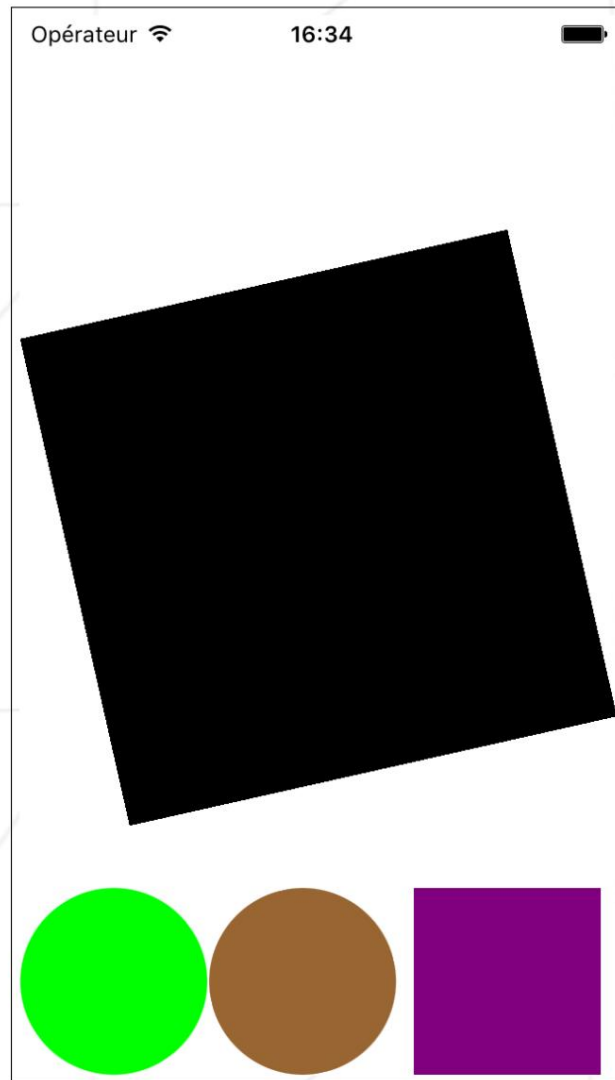
이러한 견해와 상호 작용하는 것이 얼마나 재미있겠습니까!

이제 모양 에 제스처 를 추가합니다.

- UIPanGestureRecognizer 를 사용하여 이동합니다.
- UIPinchGestureRecognizer 를 사용하여 확장하거나 축소합니다.
- UIRotationGesture 를 사용하여 회전합니다.




모양과 상호 작용할 때 중력 효과를 취소할 수 있지만 충돌이나 탄성은 취소할 수 없습니다. 물체를 떨어뜨리면 다시 중력을 받아야 합니다.



## 제7장

### 연습 03: CoreMotion

	운동: 03
코어모션	
제출할 파일: Swift Standard Library, UIKit, CoreMotion	
승인된 기능: 해당 없음	
참고: 해당 사항 없음	

프로젝트 설정에서 세로 모드는 승인되지 않아야 합니다.

이제 CoreMotion을 사용하여 가속도계 데이터에 따라 중력 설정을 수정합니다. 모양은 공간에서 장치의 방향에 따라 다른 방향으로 떨어져야 합니다.



iOS 시뮬레이터는 아직 센서 데이터 시뮬레이션을 허용하지 않으므로 실제 Apple 기기에서 이 애플리케이션을 테스트해야 합니다. 만약에 소유한 것이 없으면 이웃 중 한 사람에게 빌려달라고 요청하십시오. Apple 장치를 찾을 수 없는 경우에도 걱정하지 마십시오. 이 연습은 매우 간단하며 평가에서는 이를 고려하지 않습니다.