Яндекс

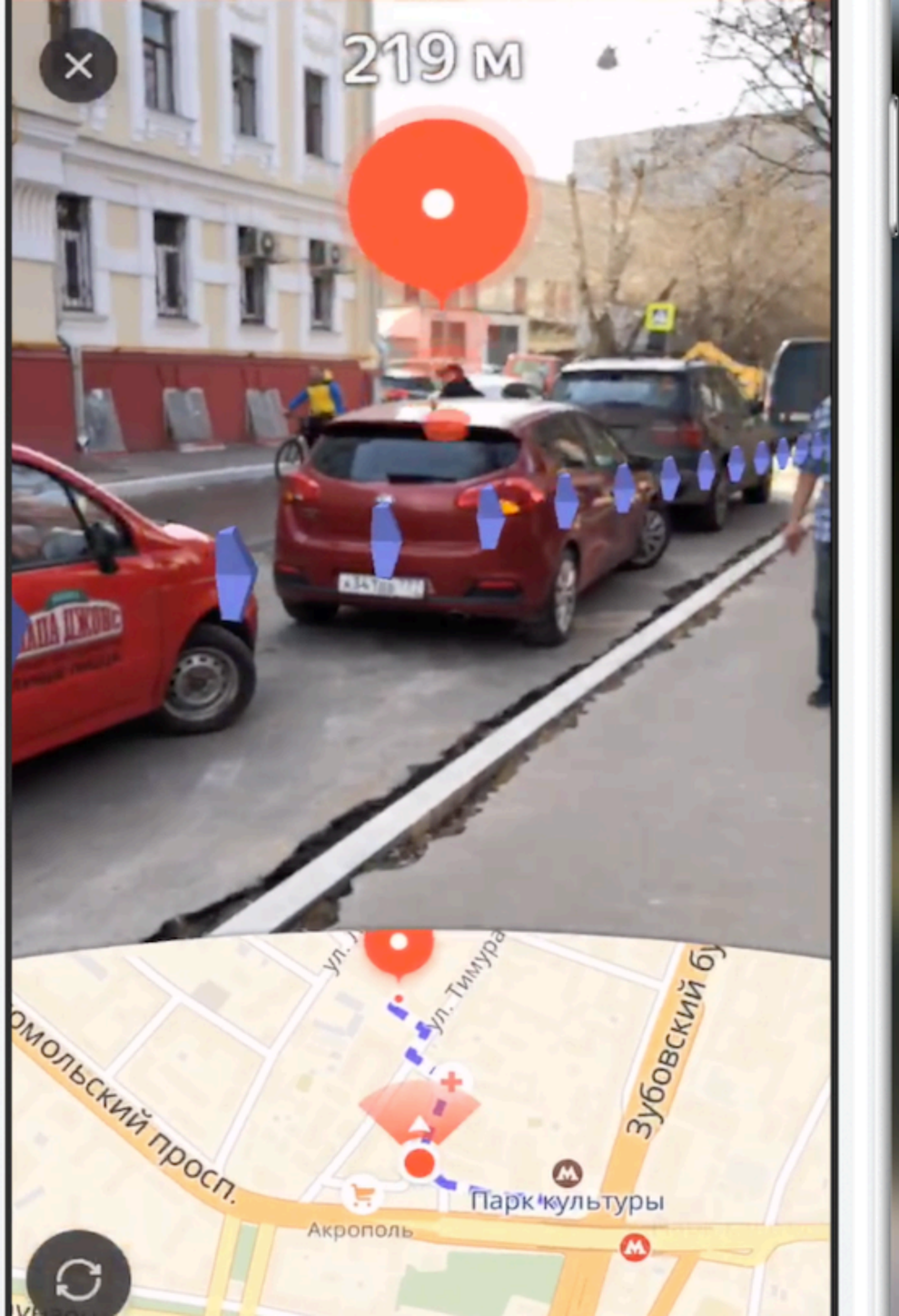
Яндекс Карты

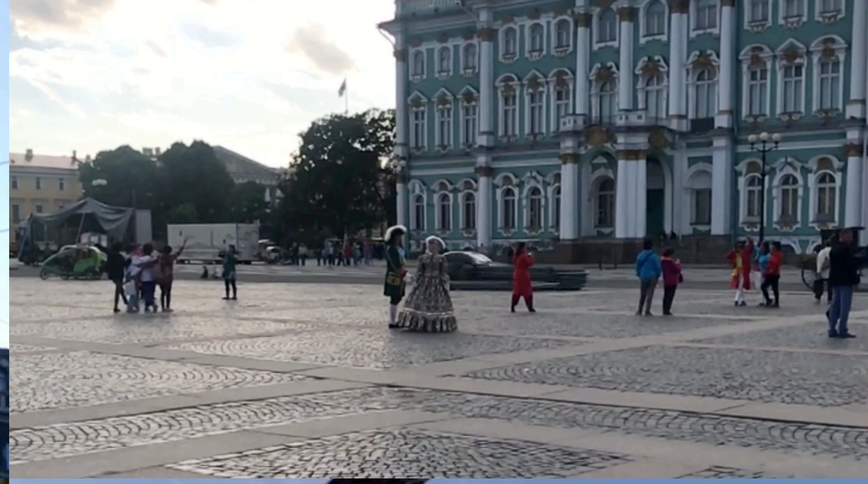
SceneKit на практике

Вячеслав Гилевич, разработчик

Третий подъезд

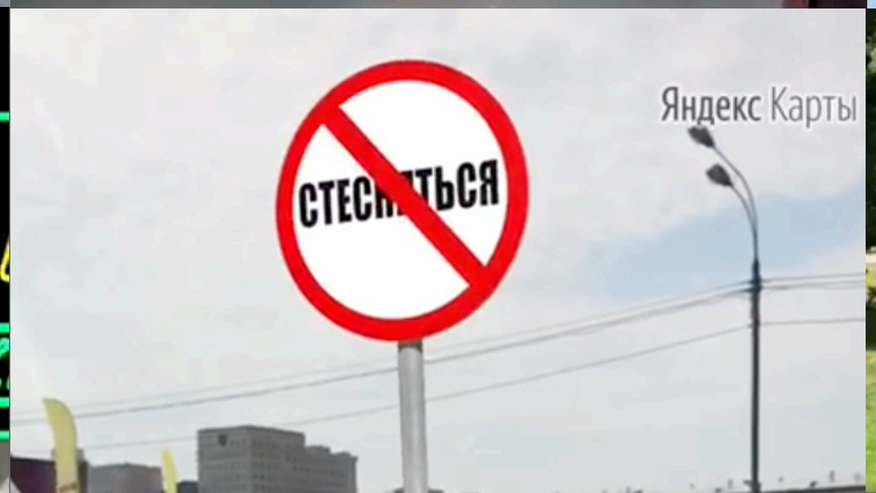
AR
ARKit


219 м
Парк культуры
Акрополь



БЕЛЫЕ
НОЧИ
В ГАЛЕРЕЕ
BIN
STEIN
CHATEAU PETERHOF
Produce of Saint-Petersburg

Яндек
PARK


СТЕСНЯТЬСЯ
Яндекс Карты

SceneKit на практике

- Загрузка моделей
- Уменьшение размера ресурсов
- Констрейнты
- Зеркальные отражения
- Видеотекстуры
- Модификаторы шейдеров
- JavaScript Action
- Запись видео

Немного про SceneKit

SceneKit — фреймворк для 3D сцен

- Высокоуровневое API
- Представлен на macOS в 2012
- Портирован на iOS в 2014
- Использован для ARKit’a в 2017

SceneKit — фреймворк для 3D сцен

Как создать сцену?

- Геометрия
- Материалы
- Освещение
- Камеры
- Анимация

SceneKit — фреймворк для 3D сцен

Кроме графики, в нём есть:

- Генератор частиц
- Физический двжиок

Загрузка моделей в SceneKit

Варианты загрузки 3D модели

- Из бандла
- По URL
- Через MDLAsset
- Через ReferenceNode

URL + настройки импорта

SCNScene.init(url:options:)

- Настраиваемый импорт
- Поддерживает .scn и .dae файлы
- Синхронная загрузка
- Необходимо копировать структуру из сцены в ноду

Используем MDLAsset

SCNScene.init(mdlAsset:)

- Загружает obj, ply, stl, usd, usda, usdc, usdz, abc, bsp, vox, md3
- Синхронная загрузка
- Необходимо копировать структуру сцены
- Оверхед на конвертацию

Используем SCNReferenceNode

SCNReferenceNode.init(url:)

- Поддерживает .scn и .dae файлы
- Ленивая загрузка
- Теряет глобальные параметры сцены

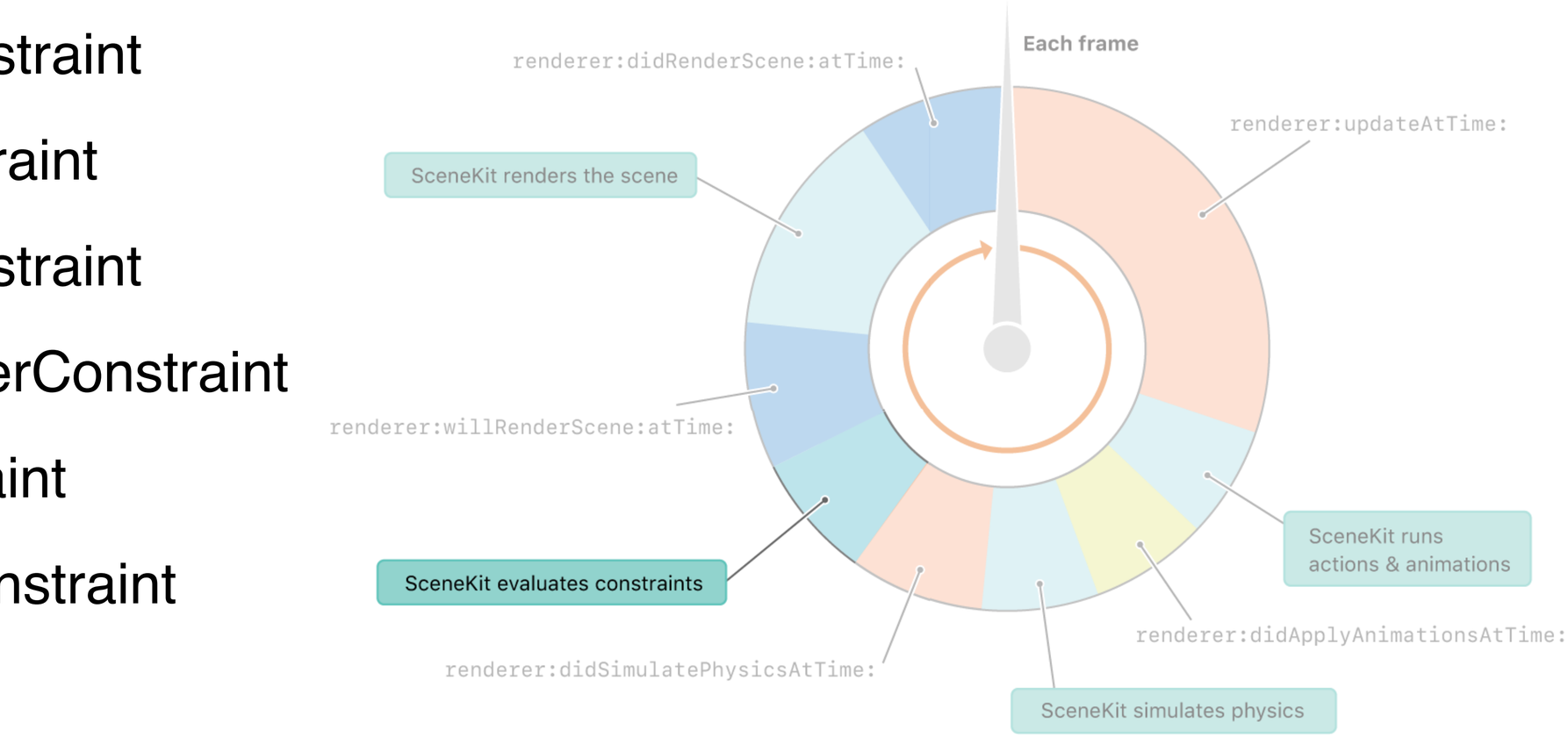
Загрузка моделей: советы

- Конвертируйте в .scn заранее
- Проверяйте сцену в Xcode
- Конвертируйте все ресурсы в .heic
- Загружайте сцену с ассетами архивом

Добавление эффектов: Констрейнты

Констрейнты

- SCNBillboardConstraint
- SCNLookAtConstraint
- SCNDistanceConstraint
- SCNAvoidOccluderConstraint
- SCNSliderConstraint
- SCNReplicatorConstraint
- SCNIKConstraint



SCNReplicatorConstraint

Просто повторяет позицию, поворот
и размер, и добавляет
дополнительное оффсеты.

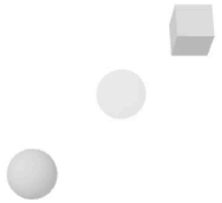
Параметры констрейнтов

Стандартные параметры

Уменьшили силу в 10 раз

Параметры констрейнтов

Стандартные параметры

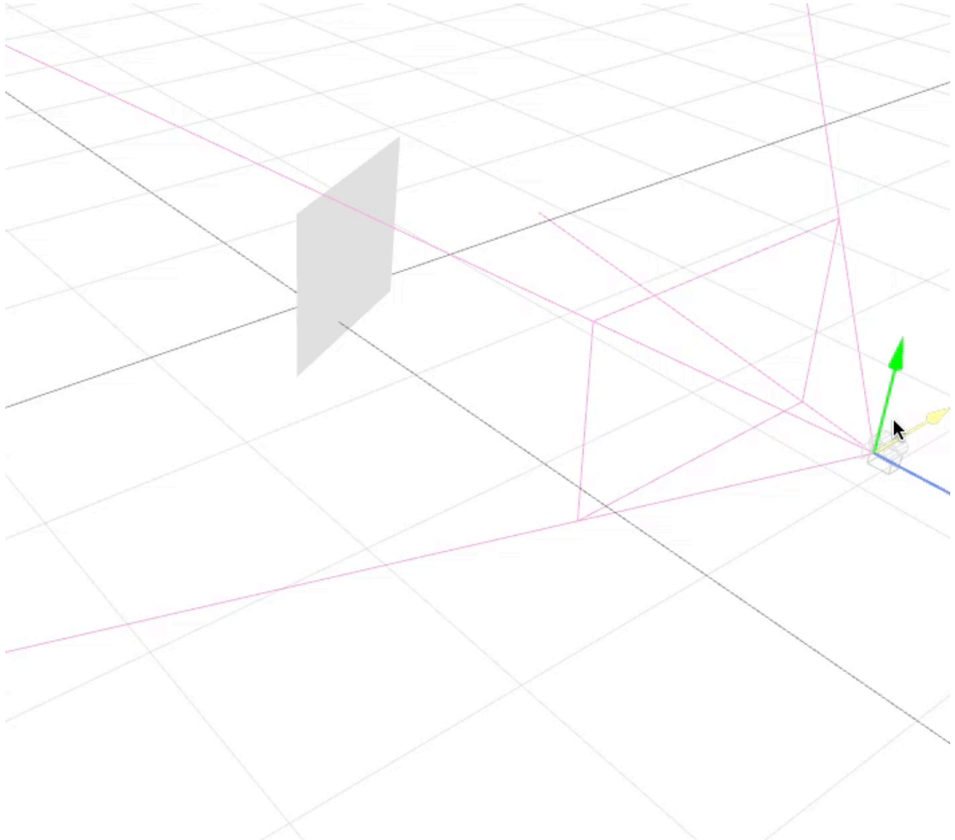


Уменьшили силу в 10 раз и
убрали инкрементальность

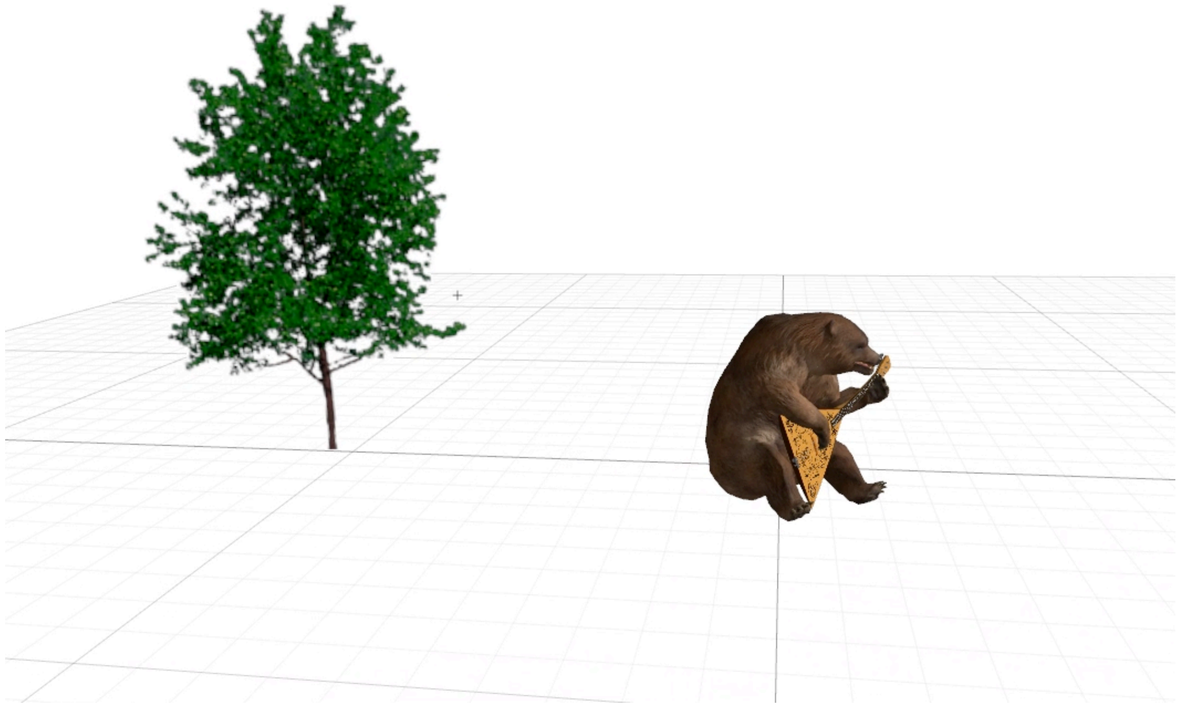
SCNBillboardConstraint I SCNLookAtConstraint

Используется для:

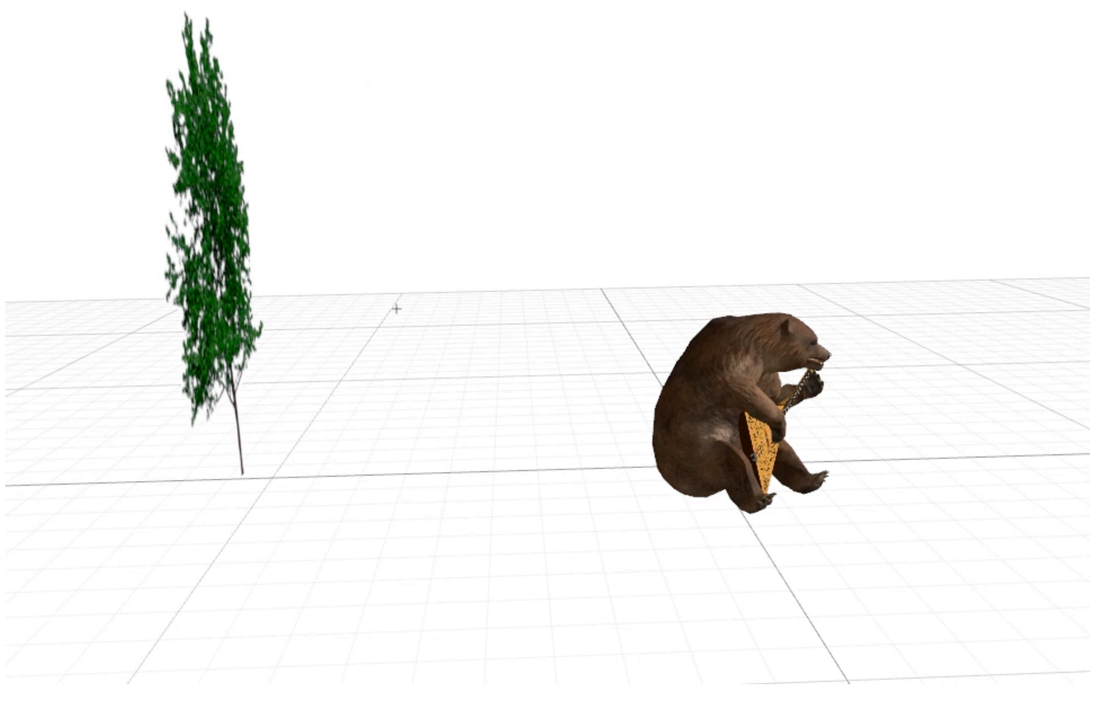
- Деревьев вдалеке
- Мелких объектов
- Спецэффектов, вроде огня или взрыва
- Привязка камеры к персонажу



SCNBillboardConstraint: зачем нужен



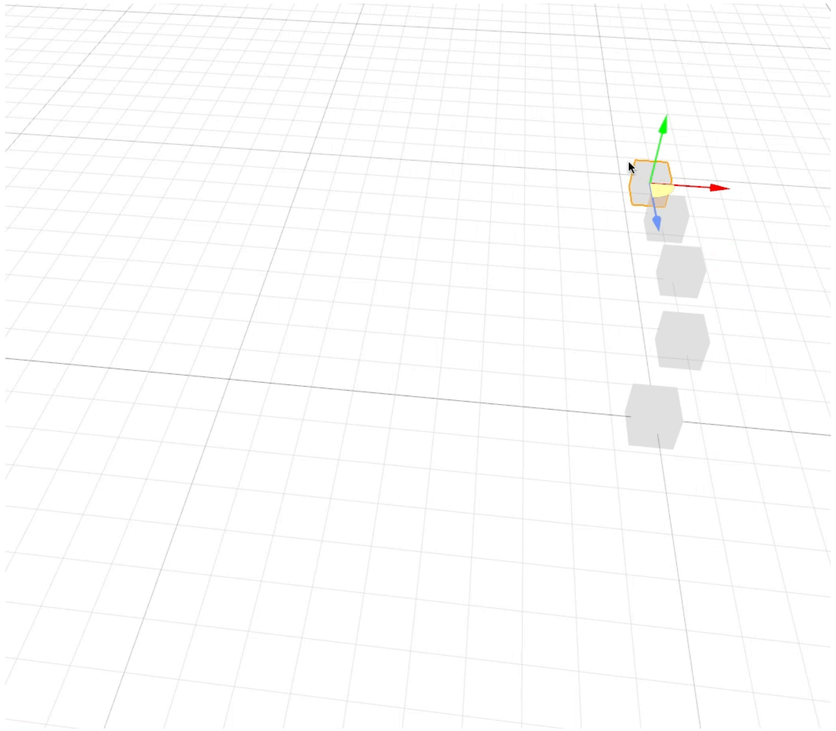
Констрейнт есть



Констрейнта нет

Яндек
PARK

SCNDistanceConstraint



Держит дистанцию между
выбранными объектами

SCNSliderConstraint

Держит дистанцию до всех физических тел

SCNIKConstraint

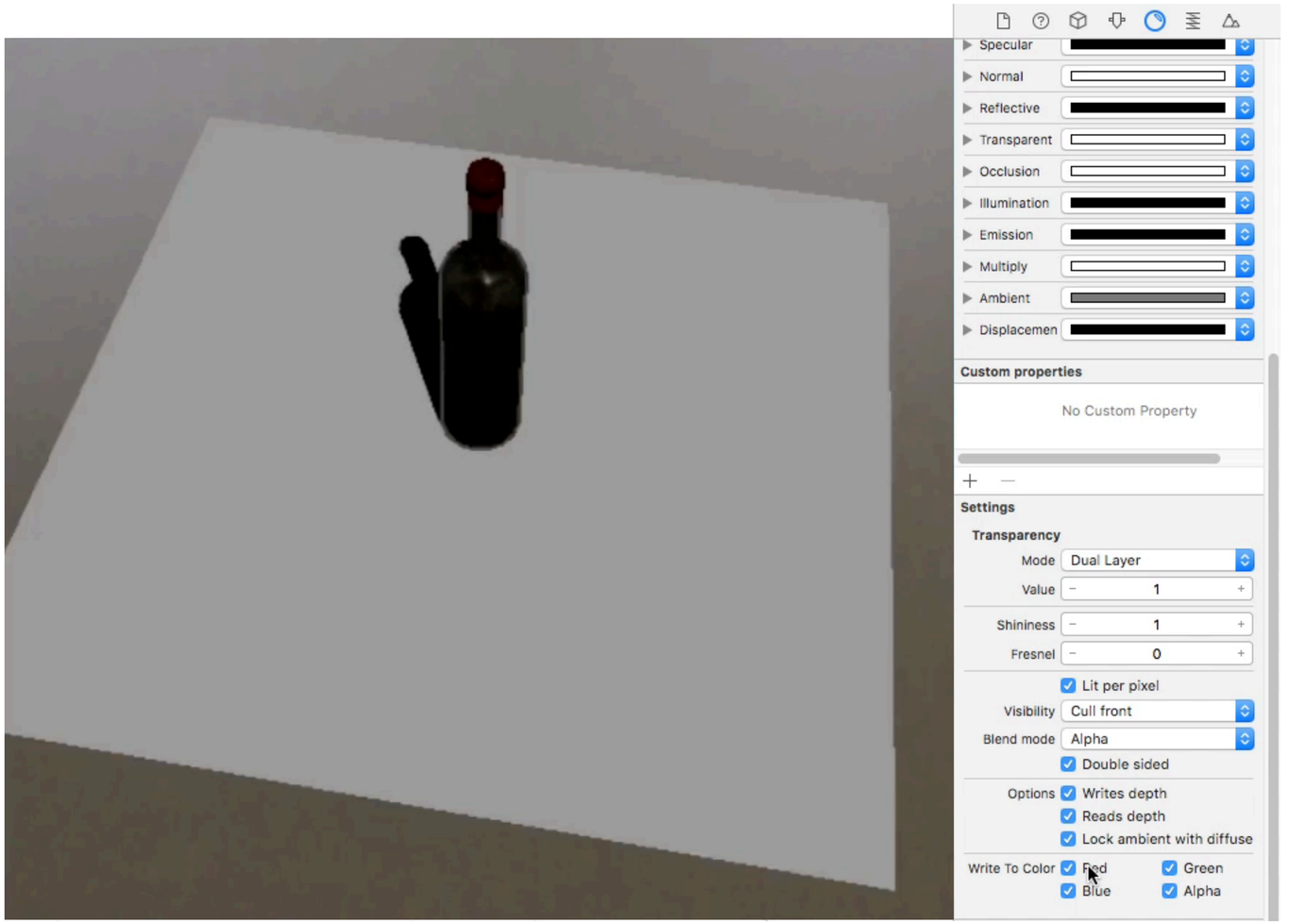
- Inverse Kinematics из коробки
- Нет возможности задать ориентацию
- Нет ограничений углов в зависимости от оси

Добавление эффектов: Тени

Тени на прозрачной поверхности

Тени для ARKit’a:

- Включи отложенные тени
- Отключи запись всех компонент



Добавление эффектов: Зеркала

Зеркальные отражения

Зеркальные отражения

Как сделать?

- SCNFloor
- Своя модель поверх него

Зеркальные отражения с картой нормалей

Вода на зеркале

Кривое зеркало

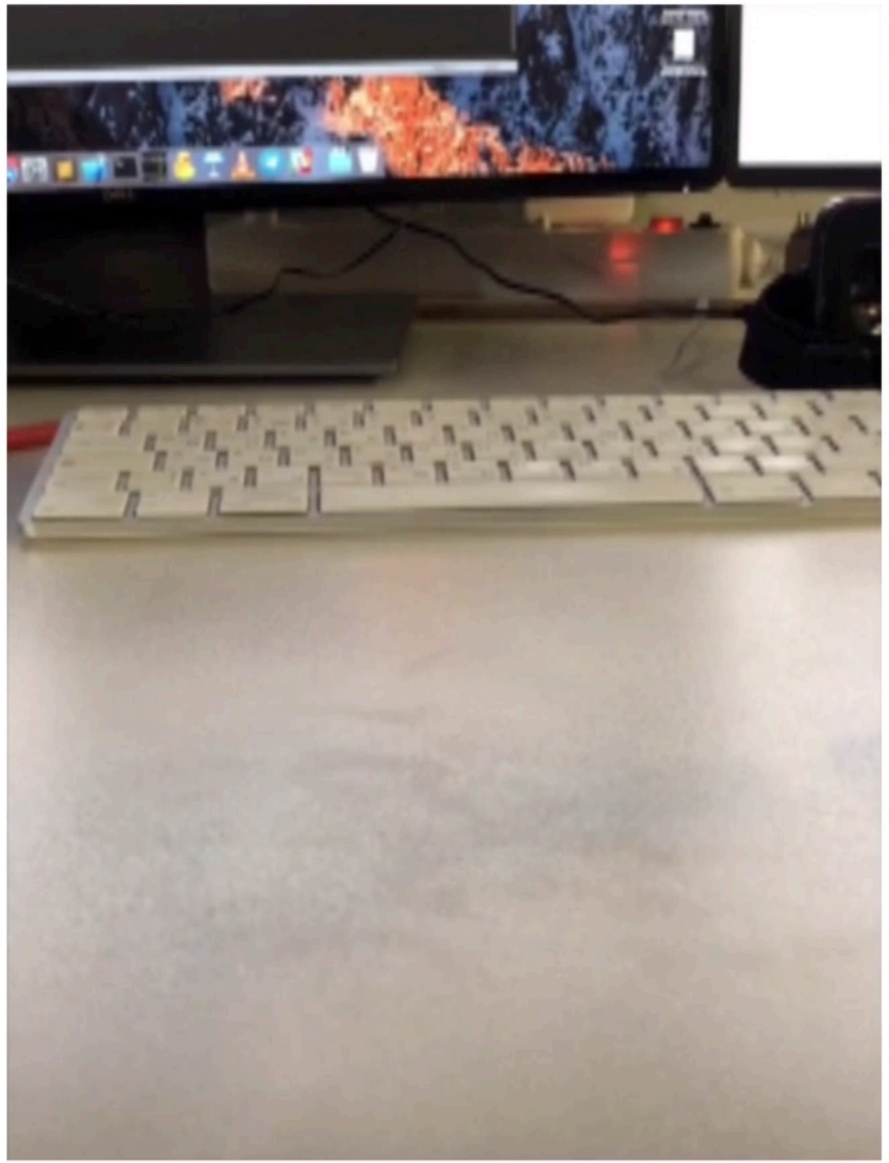
Зеркальные отражения

Добавление эффектов: Видеотекстуры

Видеотекстуры

Как сделать?

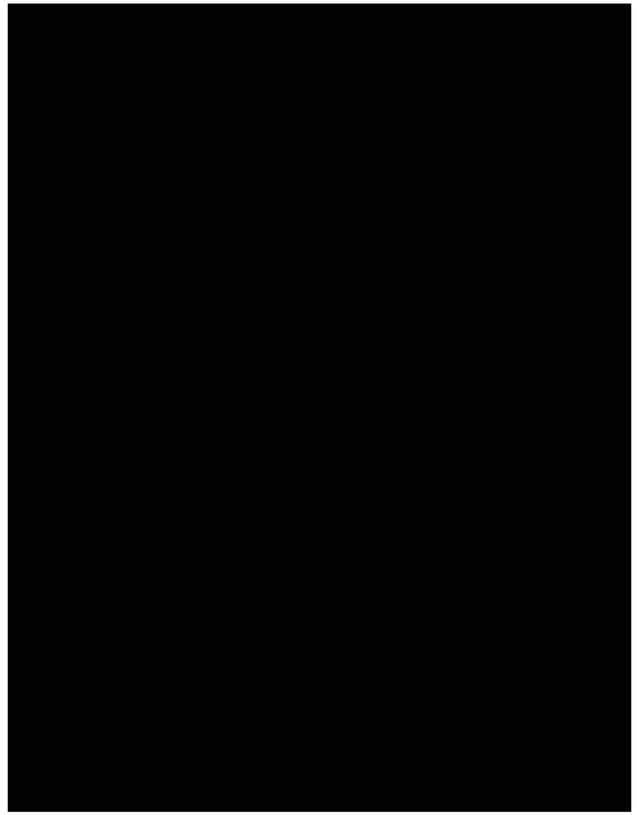
- SKScene
- SKVideoNode
- SCNMaterialProperty



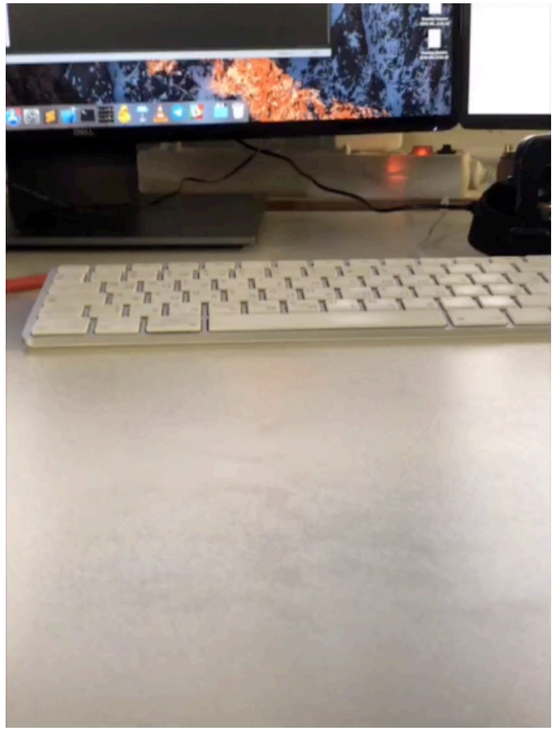
Видеотекстуры: сохранение

Как заставить работать?

- Поправить Binary Plist
- Внутри Binary Plist’a
- ~~Внутри другуго Binary Plist’a~~



Видеотекстуры: 3D видео



Видео с цветом

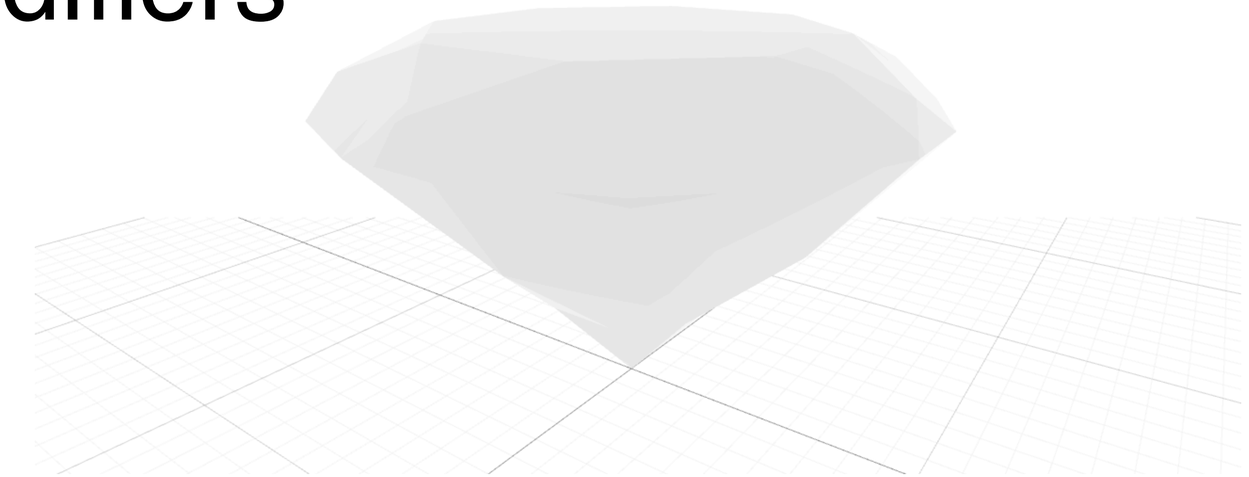
Видео с картой глубины

Добавление эффектов: Shader Modifiers

Shader Modifiers

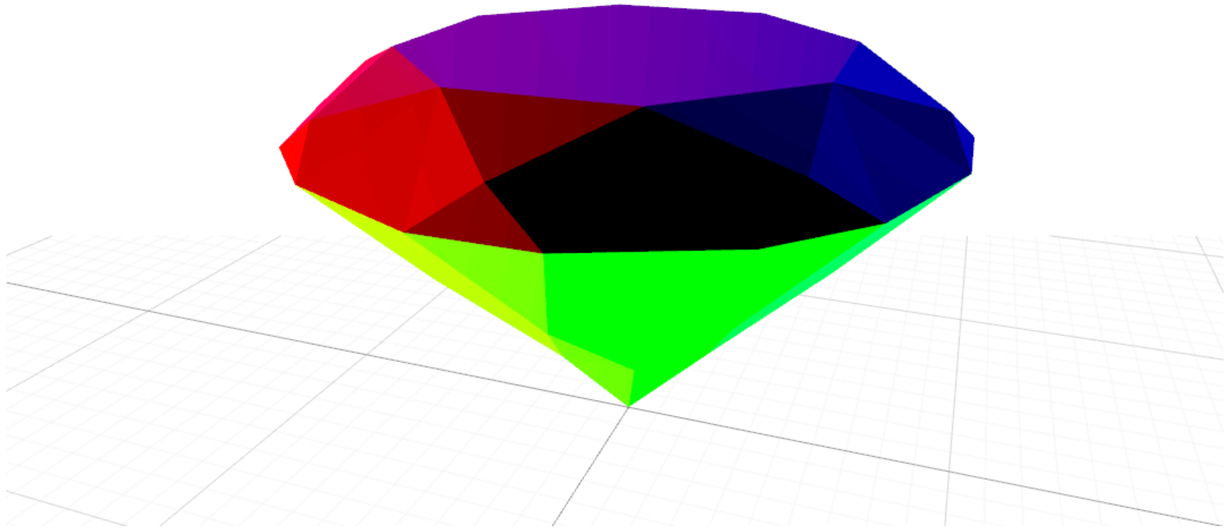
- Доступны в визуальном редакторе
- Поддерживают glsl и metal shading language
- Доступны результаты работы стандартных шейдеров

Shader Modifiers



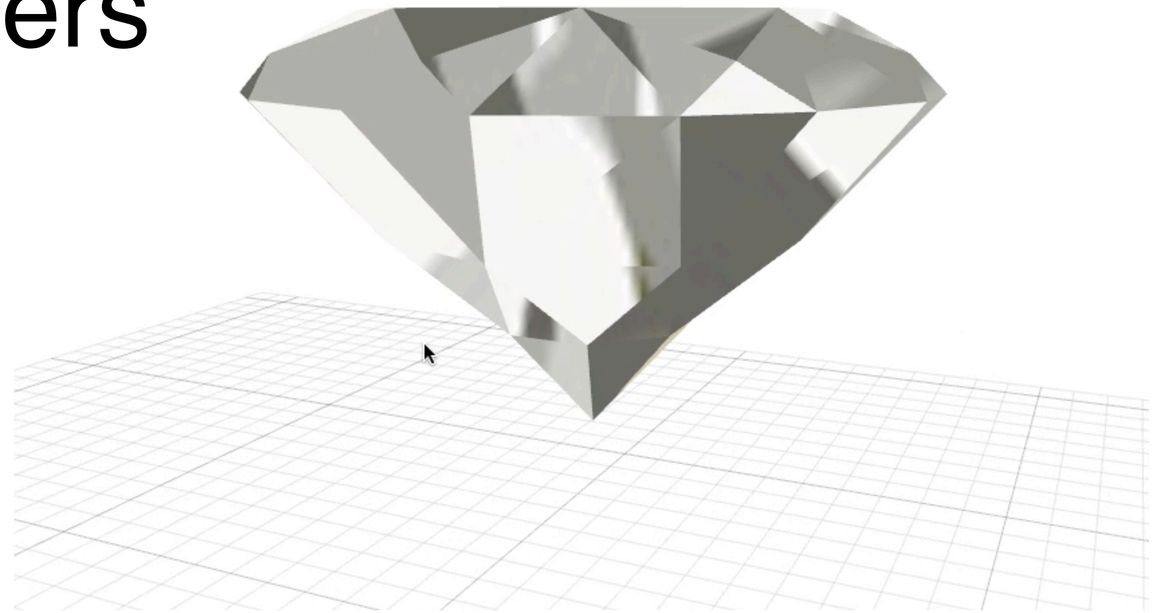
```
#pragma arguments

float refractionCoeff;
texture2d ref;

#pragma body

return _output;
```

Shader Modifiers



```
#pragma arguments

float refractionCoeff;
texture2d ref;

#pragma body

float4 cubeTexCoord = float4(refract(_surface.view, _surface.normal, refractionCoeff), 0) * scn_frame.viewTransform;

_output.color = cubeTexCoord;
return _output;
```

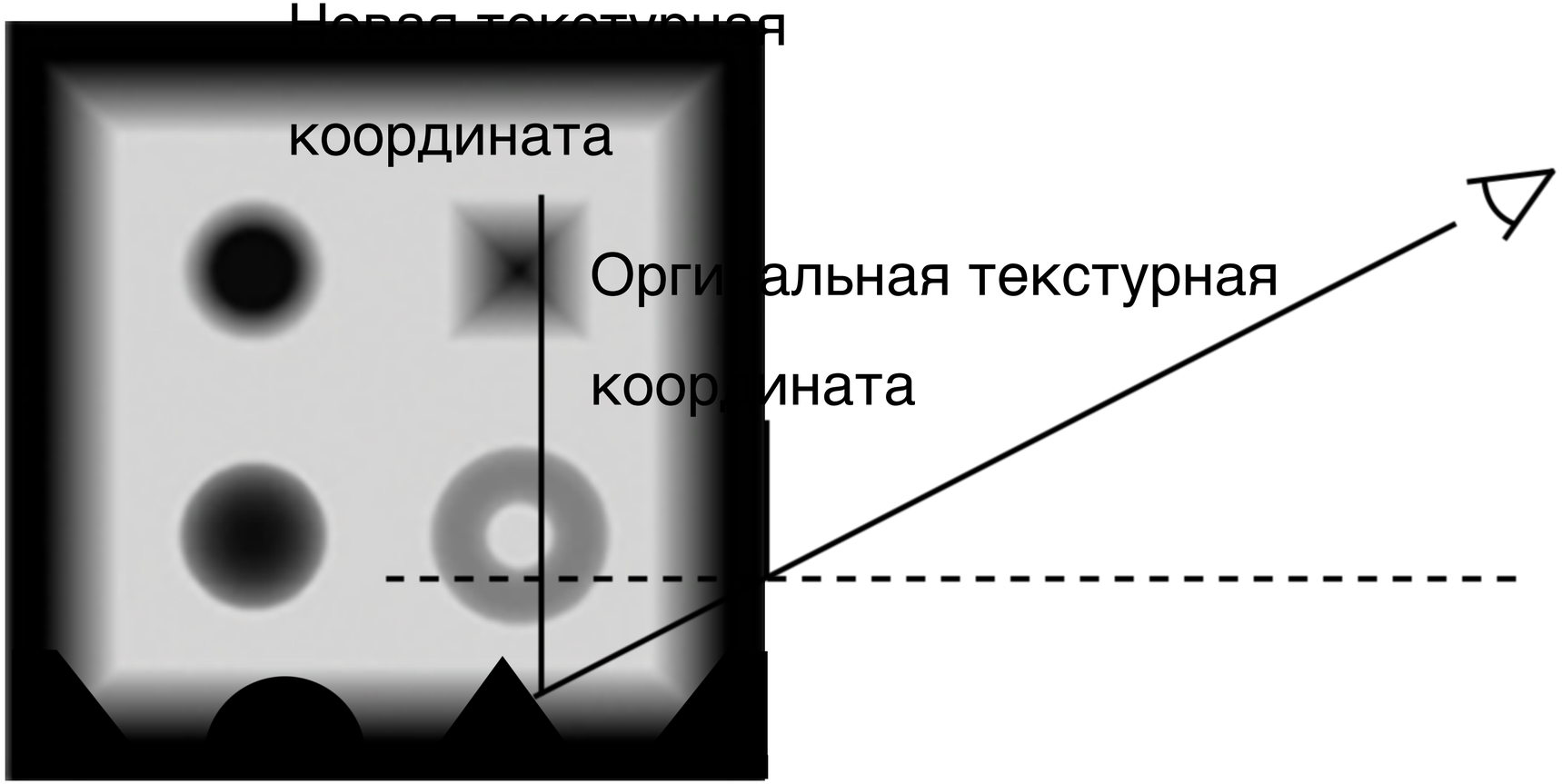
Shader Modifiers



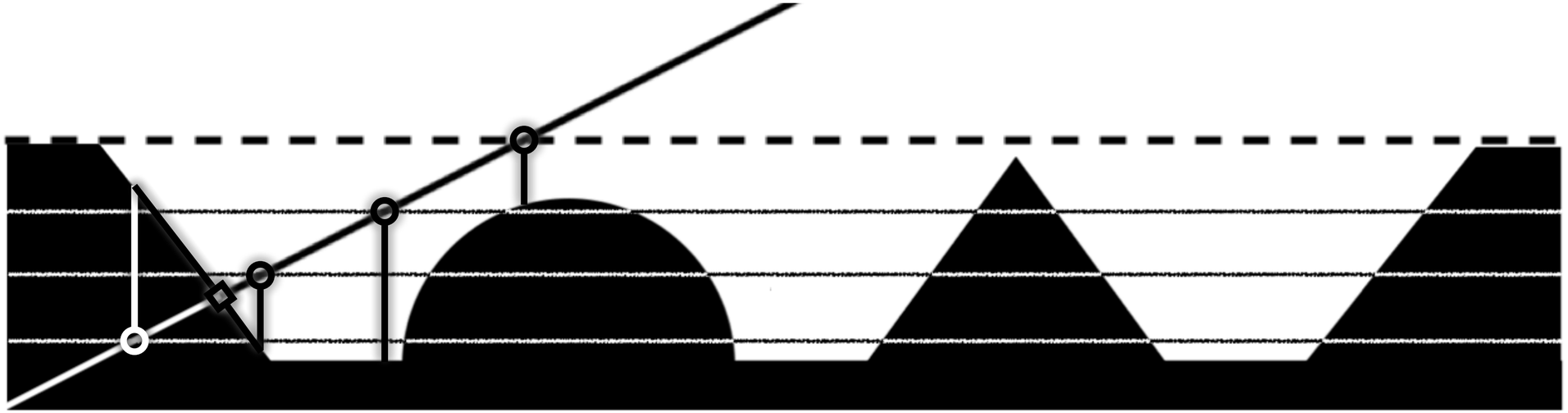
```
#pragma arguments

float refractionCoeff;
texture2d ref;

#pragma body

float4 cubeTexCoord = float4(refract(_surface.view, _surface.normal, refractionCoeff), 0) * scn_frame.viewTransform;
float maximum = max(abs(cubeTexCoord.x), max(abs(cubeTexCoord.y), abs(cubeTexCoord.z)));
cubeTexCoord = float4(cubeTexCoord.x / maximum, cubeTexCoord.y / maximum, cubeTexCoord.z / maximum, 1);

float pitch = (abs(cubeTexCoord.x) > 0.99) ? abs(cubeTexCoord.y) : abs(cubeTexCoord.x);
float yaw =  (abs(cubeTexCoord.z) > 0.99) ? abs(cubeTexCoord.y) : abs(cubeTexCoord.z);

constexpr sampler sm = sampler(filter::linear, s_address::repeat);
float4 tex = ref.sample(sm, float2(yaw, pitch));

_output.color = tex;
return _output;
```

Parallax Mapping

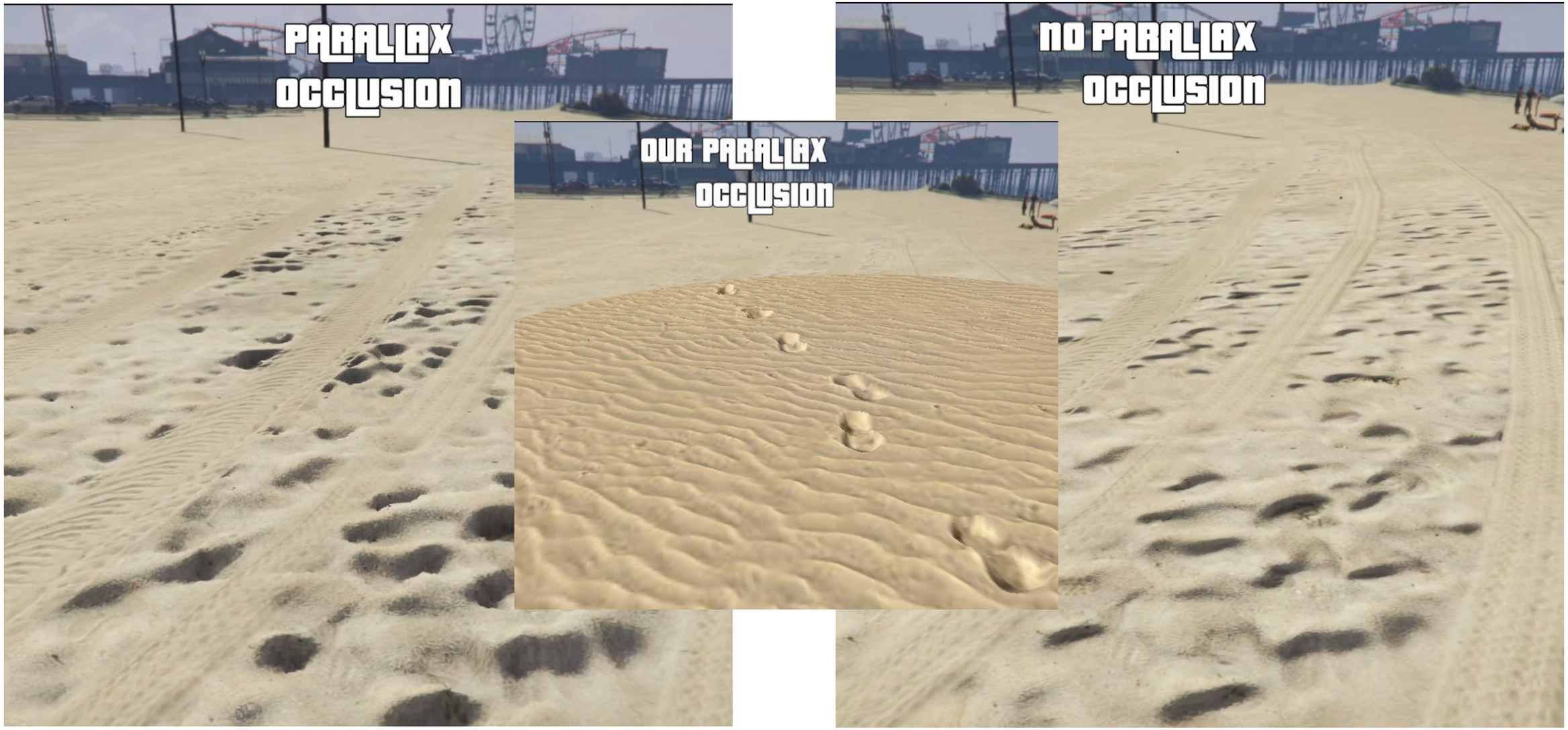
Parallax Mapping



Parallax Mapping

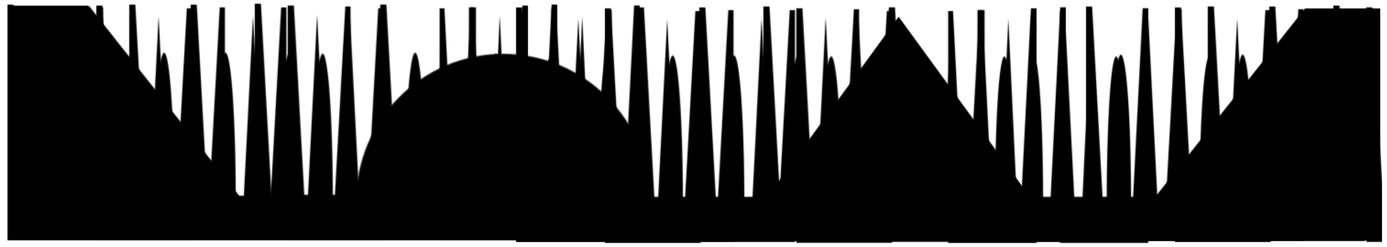


Parallax Mapping



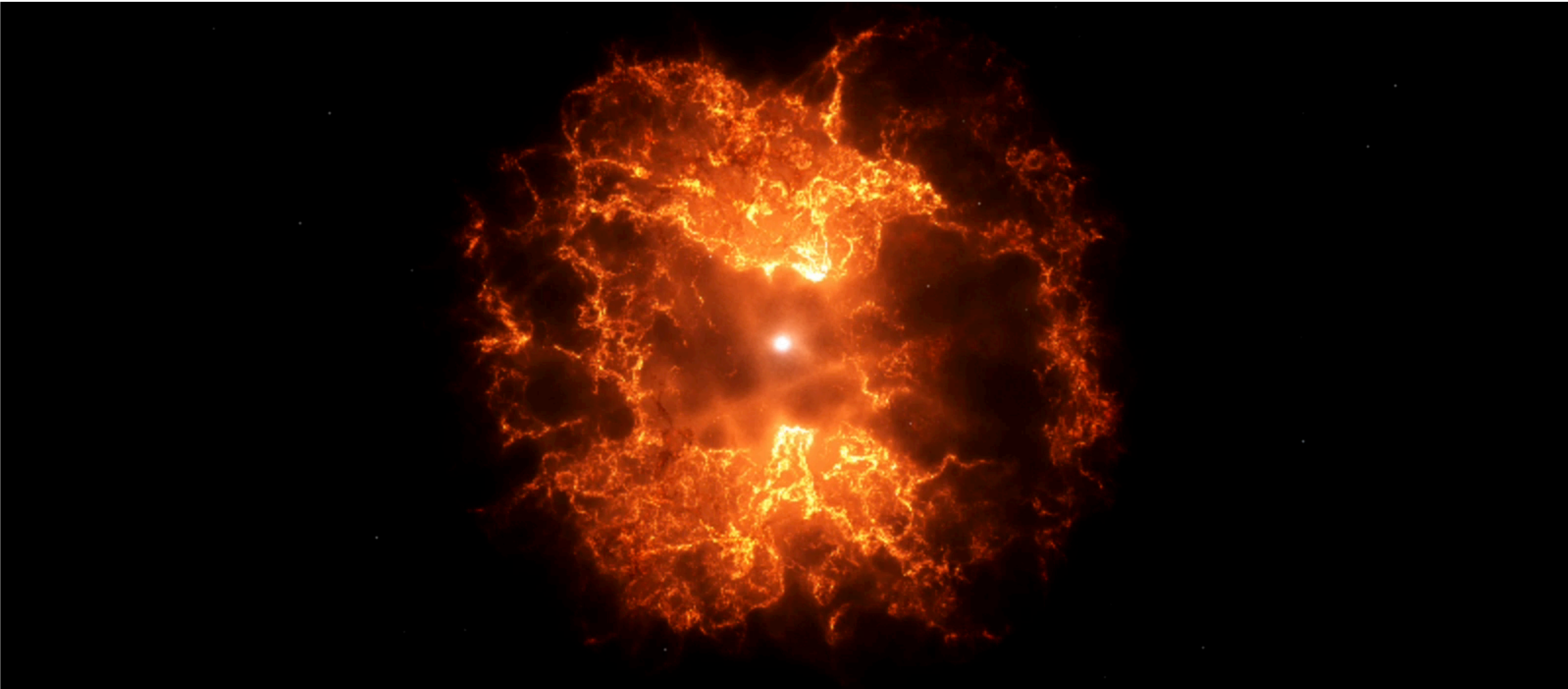
Мех в SceneKit’е

Мех в SceneKit’е

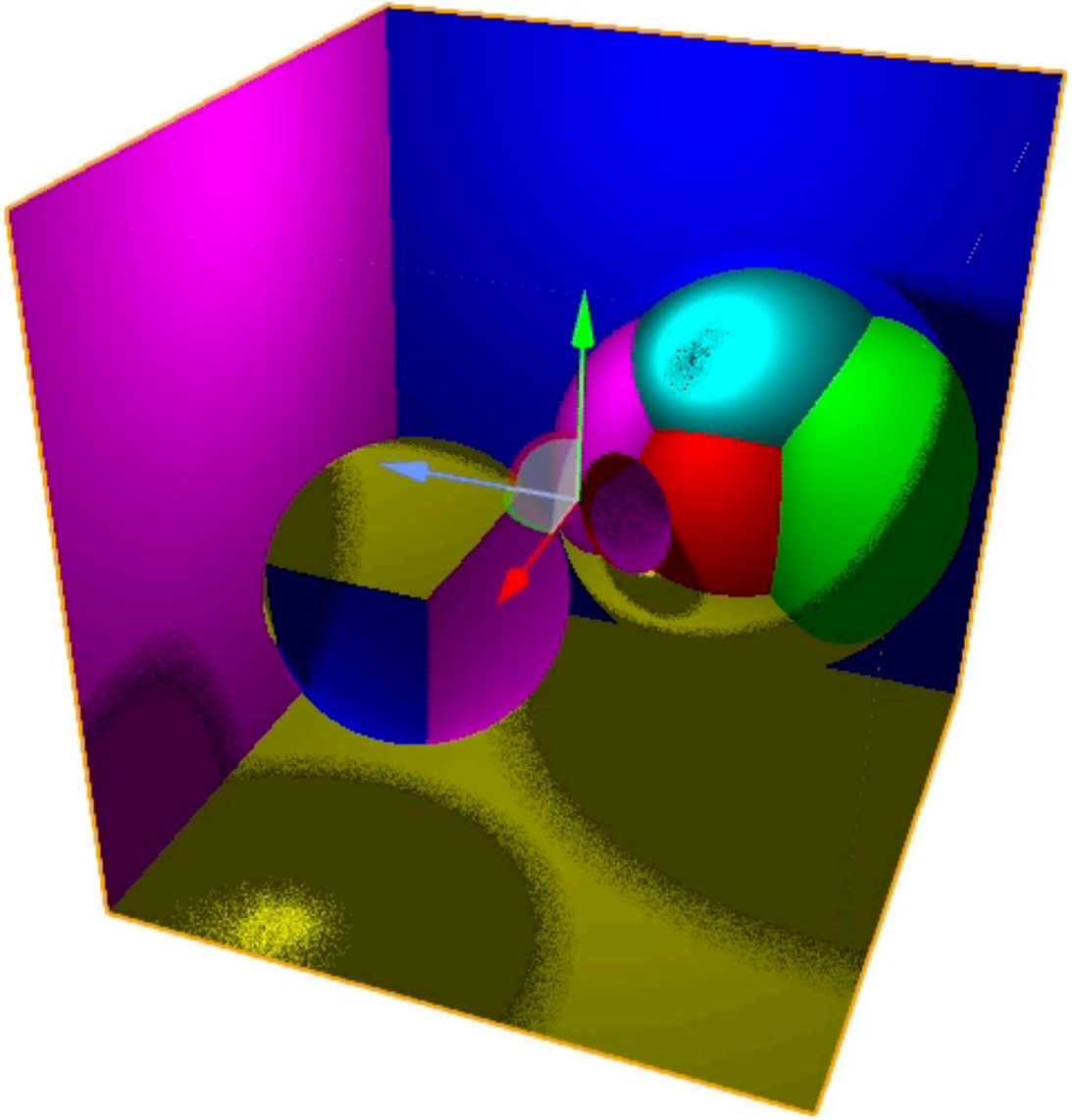


Мех в SceneKit’е

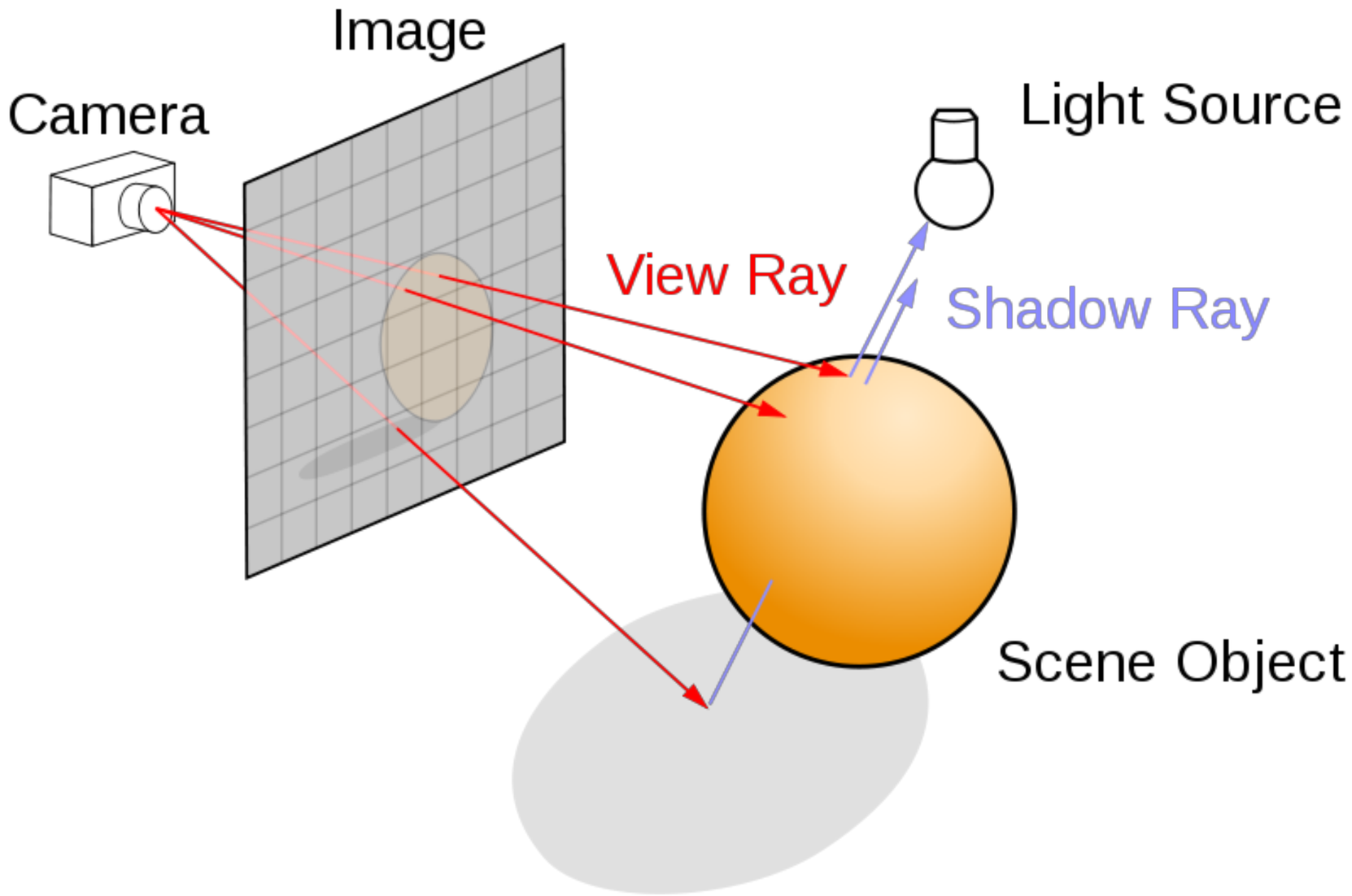
Ray Marching в SceneKit’е



Ray Casting

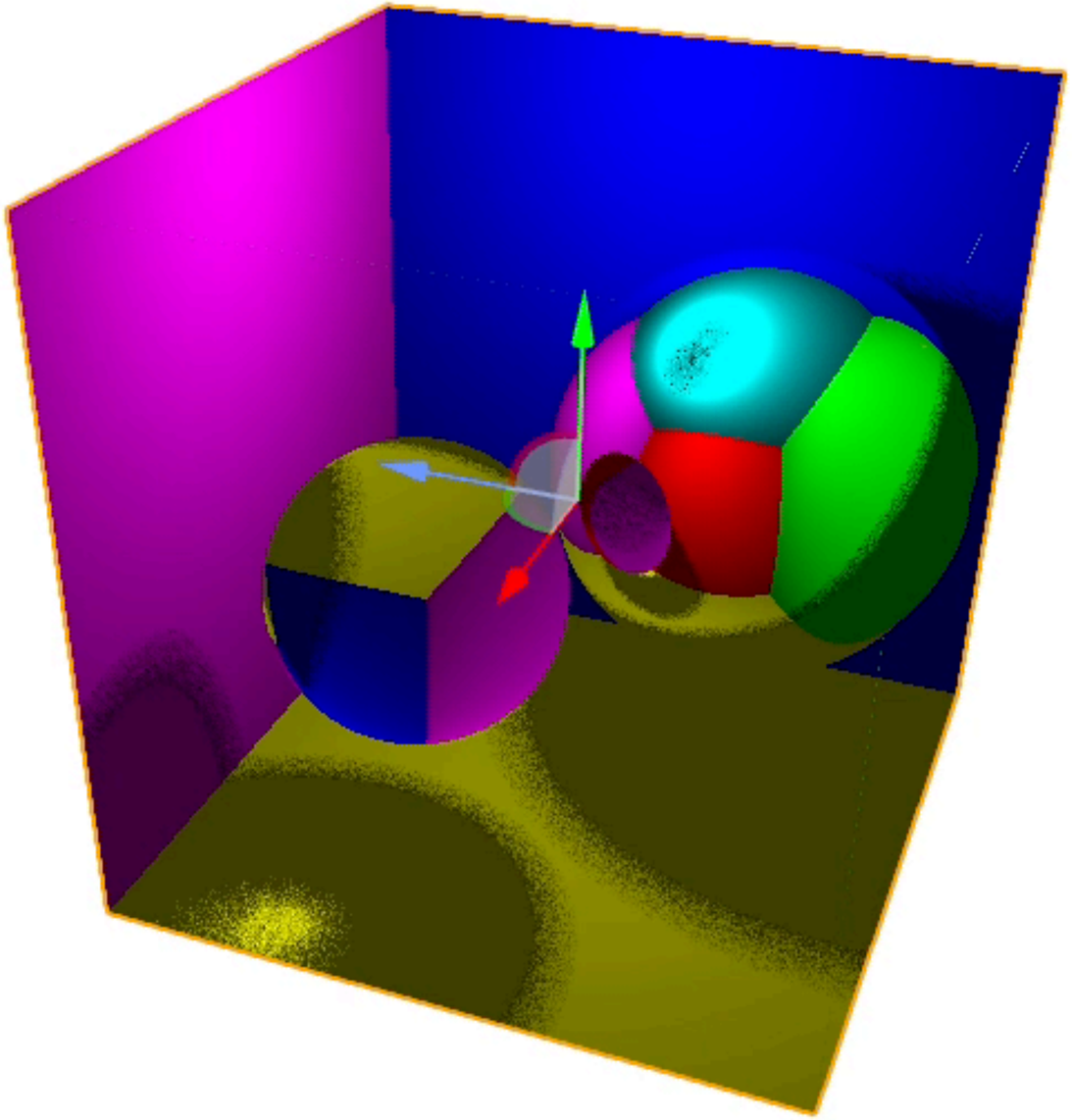


Ray Casting

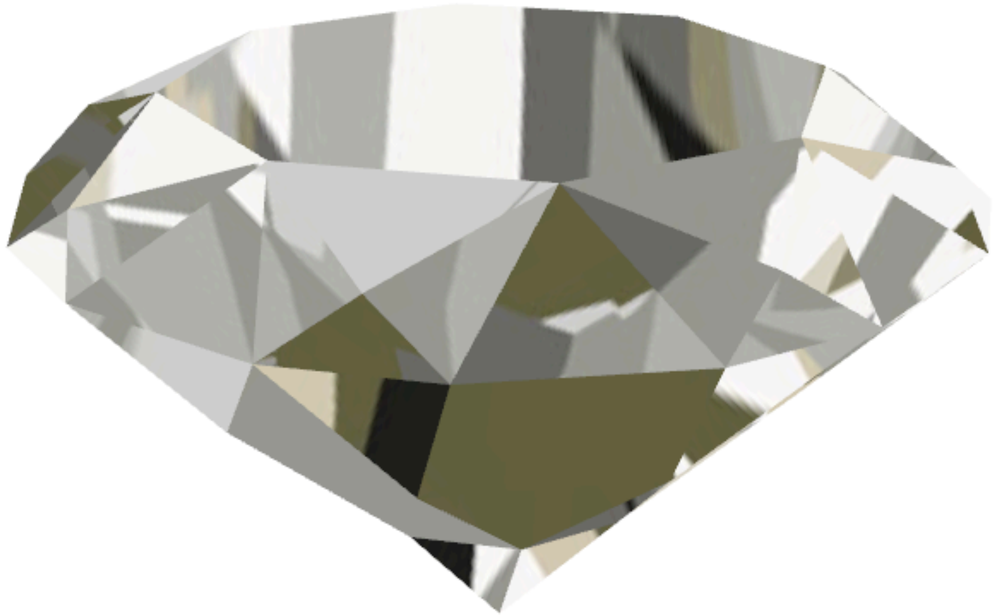


Ray Casting

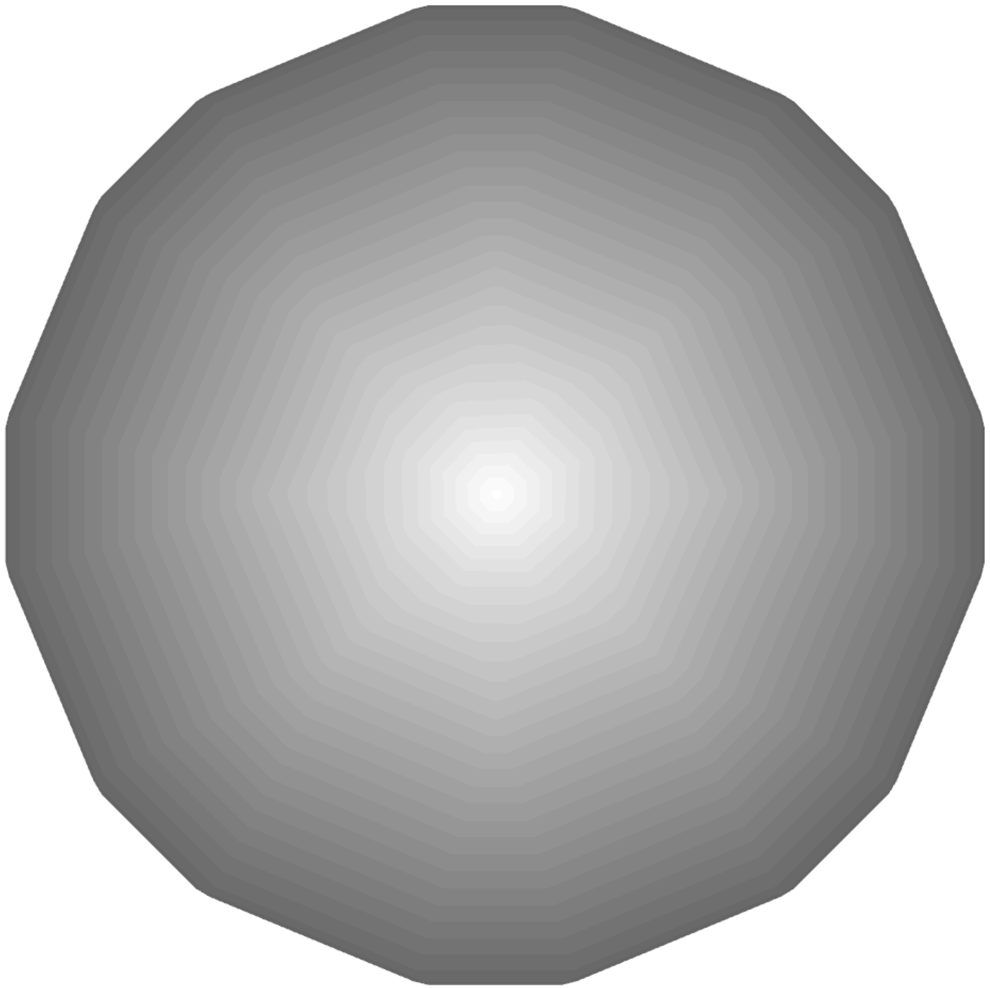
Ray Casting

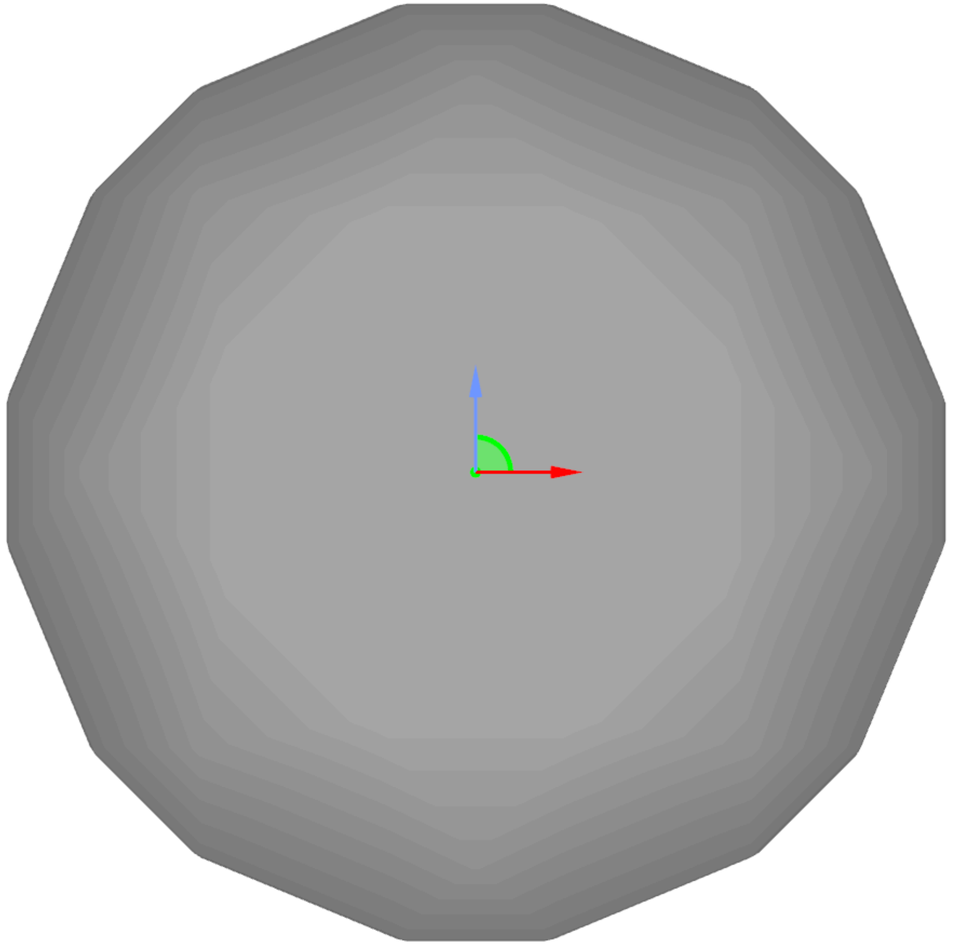


Diamond v2



Diamond v2

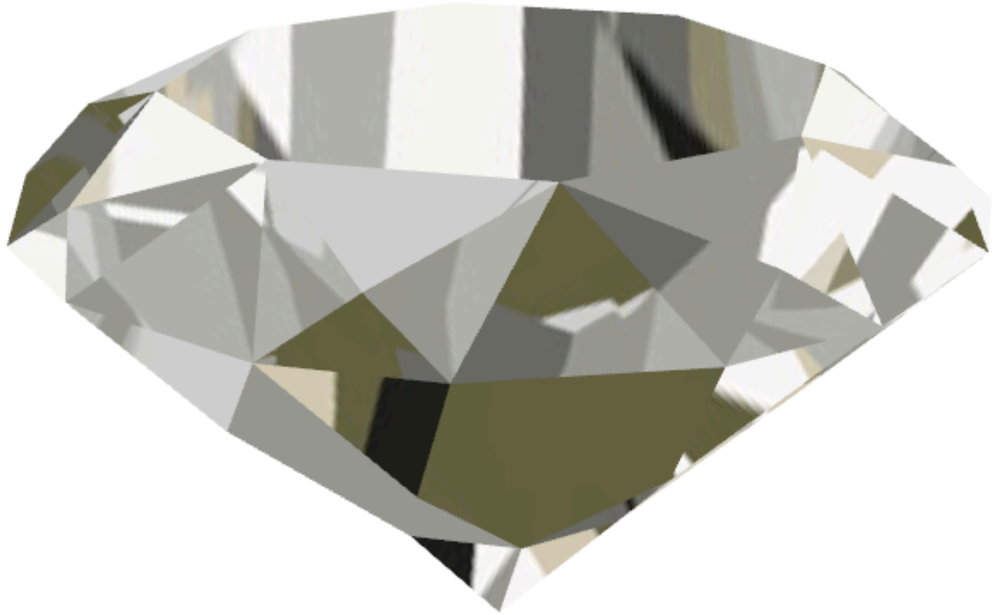




[!!!] Поменять рисунок: добавить анимацию распилки

Diamond v2

FPS: 6



FPS: 60

JavaScript в SceneKit’e

JavaScript?

Доступ к:

- Нодам
- Анимациям
- Модификаторам шейдеров
- JavaScript Action’ам
- … и всему остальному SceneKit’у(Кроме вьюх) и URL’ов)

JavaScript Action

Созданная анимация

Загруженная анимация

JavaScript Action в scn файле

JavaScript Action в scn файле

```
519             <string>node.opacity = Math.sin(elapsedTime);</string>
520             <dict>
521                 <key>$classes</key>
522                 <array>
523                     <string>SCNActionJavaScript</string>
524                     <string>SCNAction</string>
525                     <string>NSObject</string>
526                 </array>
527                 <key>$classname</key>
528                 <string>SCNActionJavaScript</string>
529             </dict>
530             <true/>
```

Погружаемся в дебаггер

```
SceneKit`-[SCNActionJavaScript initWithCoder:]:
->  0x1063c99d4 <+0>:   pushq  %rbp
    0x1063c99d5 <+1>:   movq   %rsp, %rbp
    0x1063c99d8 <+4>:   pushq  %r15
    0x1063c99da <+6>:   pushq  %r14
    0x1063c99dc <+8>:   pushq  %rbx
    0x1063c99dd <+9>:   subq   $0x38, %rsp
    0x1063c99e1 <+13>:  movq   %rdx, %rbx
    0x1063c99e4 <+16>:  leaq   -0x48(%rbp), %rax
    0x1063c99e8 <+20>:  movq   %rdi, (%rax)
    0x1063c99eb <+23>:  movq   0x329f2e(%rip), %rcx      ; (void *)0x00000001066f7200: SCNActionJavaScript
    0x1063c99f2 <+30>:  movq   %rcx, 0x8(%rax)
    0x1063c99f6 <+34>:  movq   0x3237db(%rip), %rsi      ; "initWithCoder:"
    0x1063c99fd <+41>:  movq   %rax, %rdi
    0x1063c9a00 <+44>:  callq  0x1065c5ac0               ; symbol stub for: objc_msgSendSuper2
    0x1063c9a05 <+49>:  movq   %rax, %r15
    0x1063c9a08 <+52>:  testq  %r15, %r15
    0x1063c9a0b <+55>:  je     0x1063c9a9d               ; <+201>
    0x1063c9a11 <+61>:  movq   0x329650(%rip), %rdi      ; (void *)0x00000001045e85d8: NSString
    0x1063c9a18 <+68>:  movq   0x323851(%rip), %rsi      ; "class"
    0x1063c9a1f <+75>:  movq   0x2b39f2(%rip), %r14      ; (void *)0x000000010484cd80: objc_msgSend
    0x1063c9a26 <+82>:  callq  *%r14
    0x1063c9a29 <+85>:  movq   0x3245c8(%rip), %rsi      ; "decodeObjectOfClass:forKey:"
    0x1063c9a30 <+92>:  leaq   0x2d1d71(%rip), %rcx      ; @"script"
```

Давайте посвизлим

```
27      static let swizzle: Void = {
28          guard let classType = NSClassFromString("SCNActionJavaScript") else { return assert(false) }
29
30          let originalMethod = class_getInstanceMethod(classType, Selector("initWithCoder:"))!
31          let swizzledMethod = class_getInstanceMethod(SCNAction.self, Selector("swizzledinitWithCoder:"))!
32
33          let added = class_addMethod(classType, Selector("initWithCoder:"), method_getImplementation(swizzledMethod),
34                                      method_getTypeEncoding(swizzledMethod))
35
36          if added {
37              class_replaceMethod(classType, Selector("swizzledinitWithCoder:"), method_getImplementation(originalMethod),
38                                  method_getTypeEncoding(originalMethod))
39          } else {
40              method_exchangeImplementations(originalMethod, swizzledMethod);
41          }
42      }()
```

Давайте посвизлим

```
15  fileprivate extension SCNAction {
16
17      @objc func swizzledinit(coder: NSCoder) -> Any? {
18          self.perform(Selector("swizzledinitWithCoder:"), with: coder)
19          guard let script = coder.decodeObject(forKey: "script") as? String else { return self }
20          self.setValue(script, forKey: "_script")
21          return self
22      }
23
24  }
```

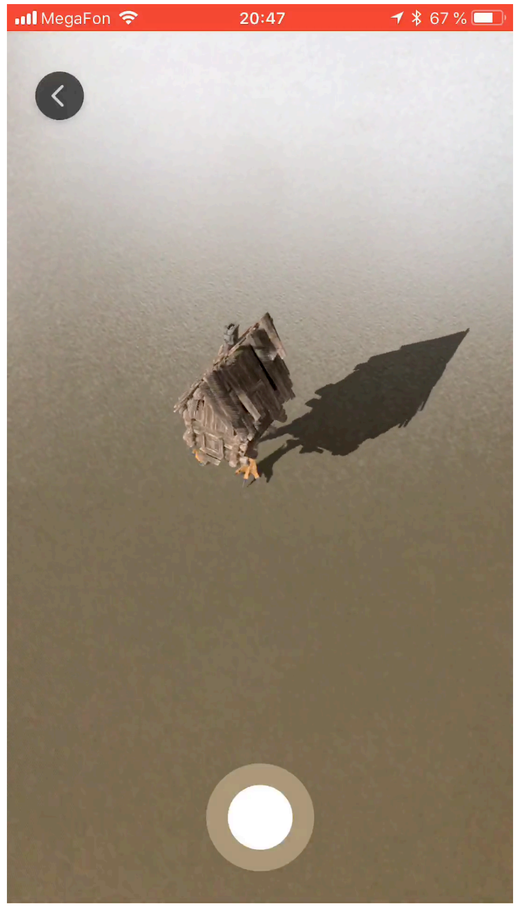
JavaScript Action: результат

Созданная анимация

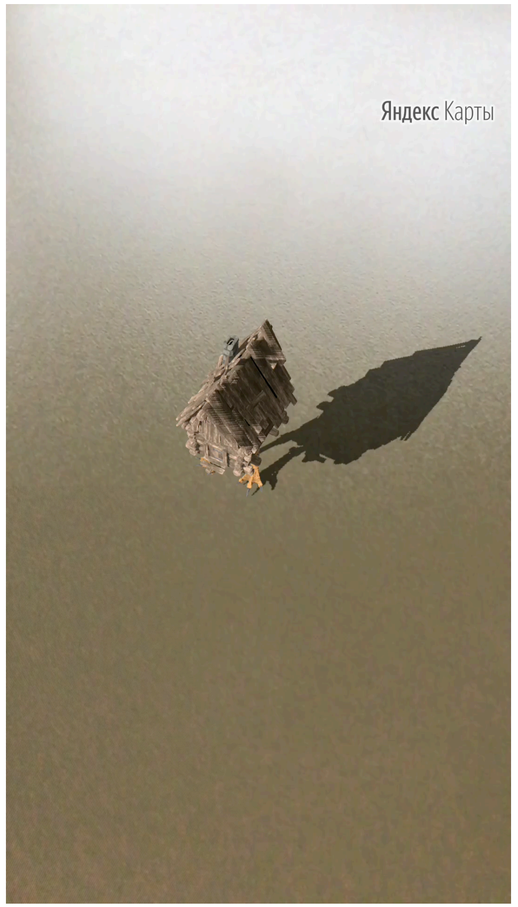
Загруженная анимация

Запись видео

Запись видео



На девайсе



Записывается

Запись видео: ReplayKit

Начать запись:

RPScreenRecorder.shared()

func startRecording()

Закончить запись:

RPScreenRecorder.shared()

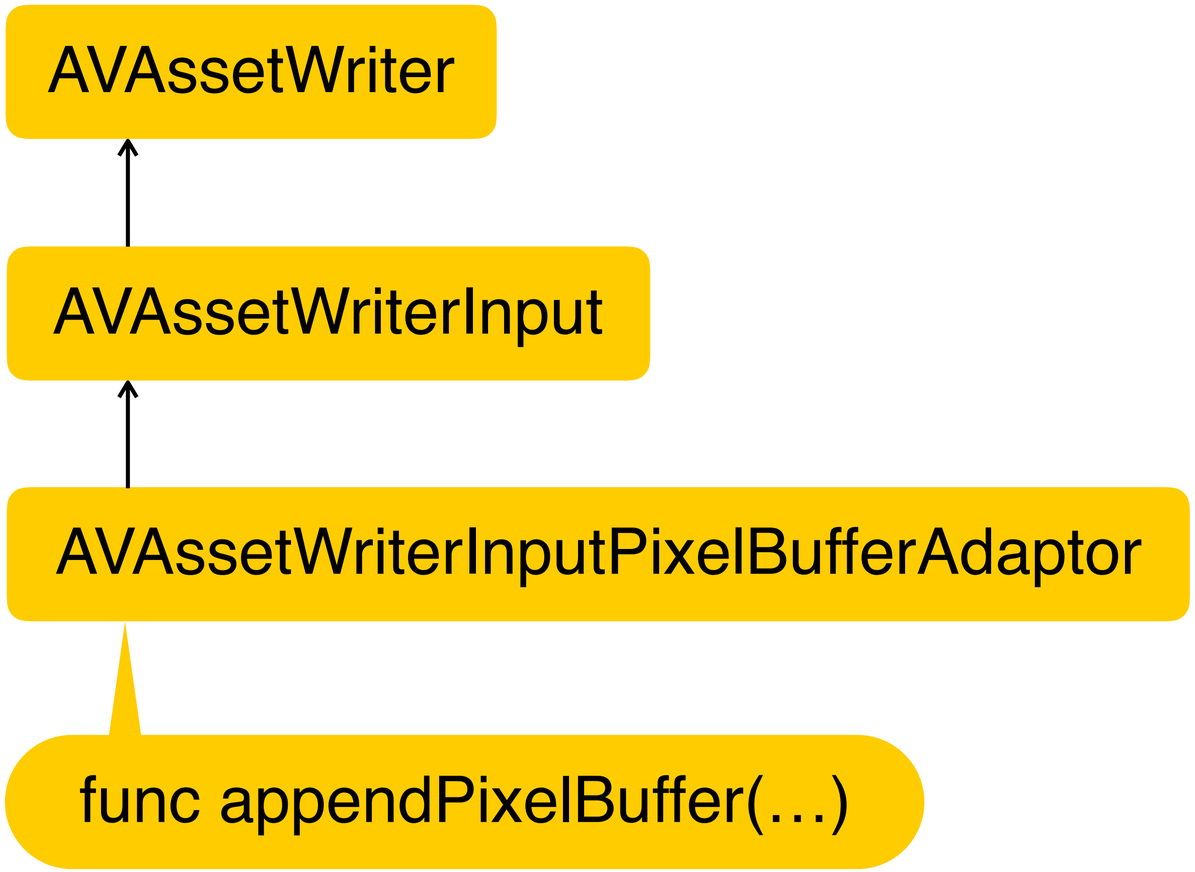
func stopRecording(callback:)

Запись видео: ReplayKit

- Работает в пару строк кода
- Нельзя записывать конкретную вьюху
- Неудобный стандартный превью

Запись видео: основы

Процесс записи:



Закончить запись:

UIImage → CVPixelBuffer

```
var pixelBuffer: CVPixelBuffer? = nil
guard CVPixelBufferCreate(kCFAllocatorDefault, width, height, kCVPixelFormatType_32BGRA , nil,
    &pixelBuffer) == kCVReturnSuccess else { return }
CVPixelBufferLockBaseAddress(pixelBuffer!, CVPixelBufferLockFlags.init(rawValue: 0))
let data = CVPixelBufferGetBaseAddress(pixelBuffer!)
let rgbColorSpace = CGColorSpaceCreateDeviceRGB()
let bitmapInfo = CGBitmapInfo(rawValue: CGBitmapInfo.byteOrder32Little.rawValue |
    CGImageAlphaInfo.premultipliedFirst.rawValue)
let context = CGContext(data: data, width: width, height: height, bitsPerComponent: 8, bytesPerRow:
    CVPixelBufferGetBytesPerRow(pixelBuffer!), space: rgbColorSpace, bitmapInfo: bitmapInfo.rawValue)
context?.draw(image, in: CGRect(x: 0, y: 0, width: image.width, height: image.height))
CVPixelBufferUnlockBaseAddress(pixelBuffer!, CVPixelBufferLockFlags(rawValue: 0))
```

UIImage → CVPixelBuffer

CVPixelBufferCreate(… width:height:pixelFormat:

CVPixelBufferLockBaseAddress(_

CVPixelBufferGetBaseAddress(_

CGContext(data: ... bitsPerComponent: bytesPerRow:…

func draw(image: UIImage)

CVPixelBufferUnlockBaseAddress(_

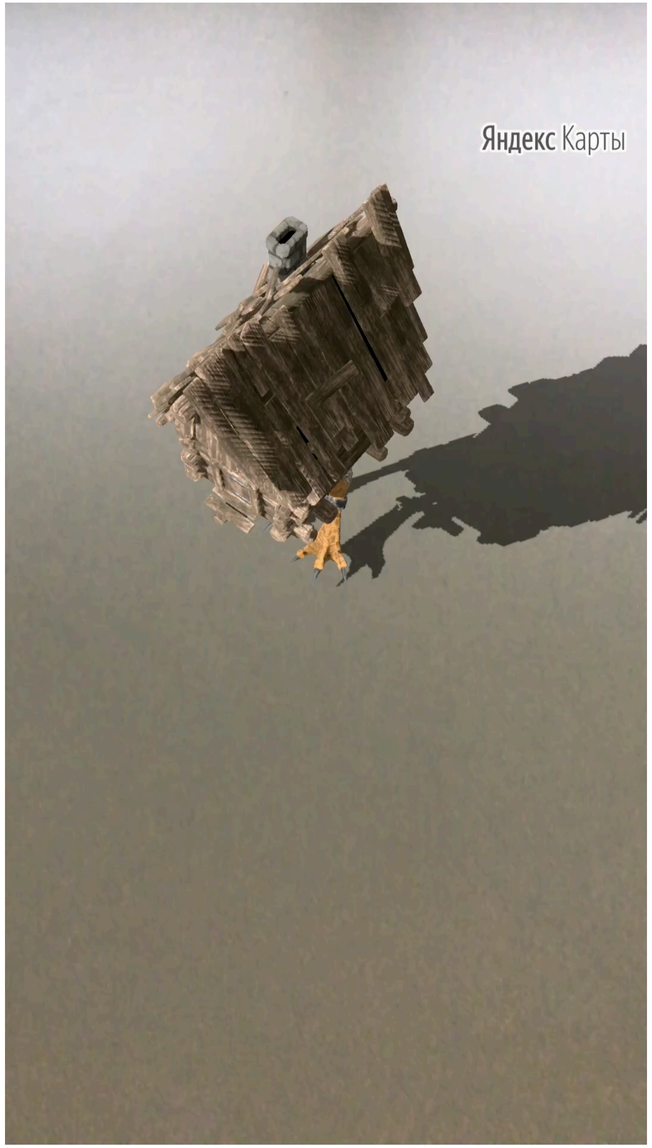
Теперь нужно это делать каждый кадр

Начать запись: CADisplayLink(target:selector)

func add(to:forModes:)

Закончить запись: func invalidate()

Производительность

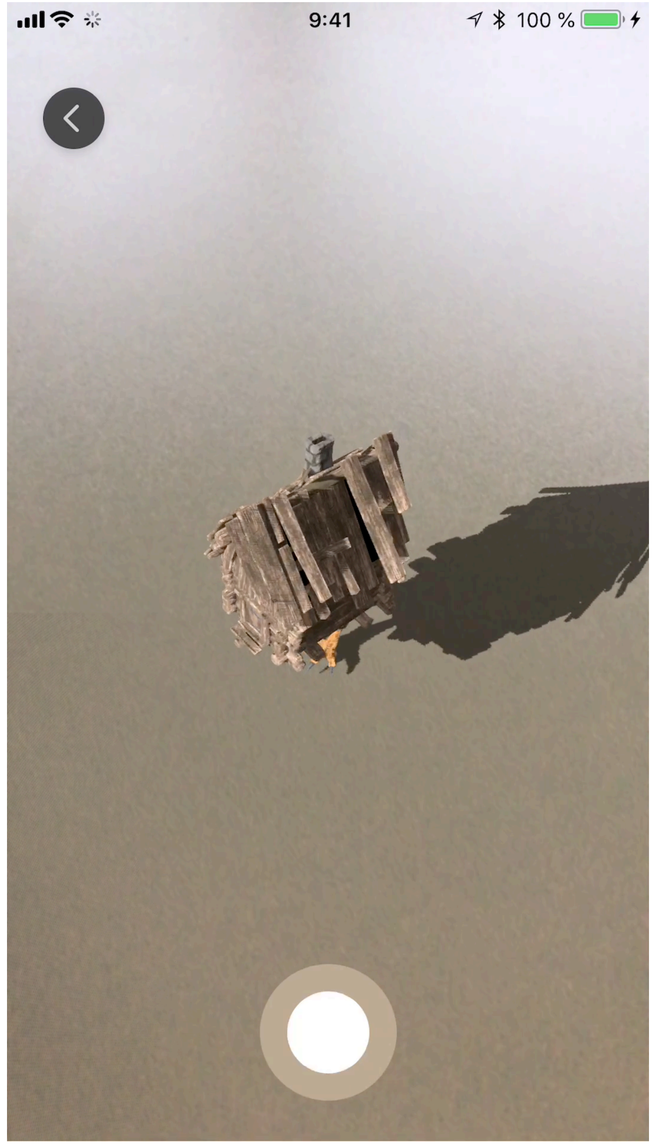


Оптимизации: раунд 1!

DispatchQueue.global()

func async(execute:)

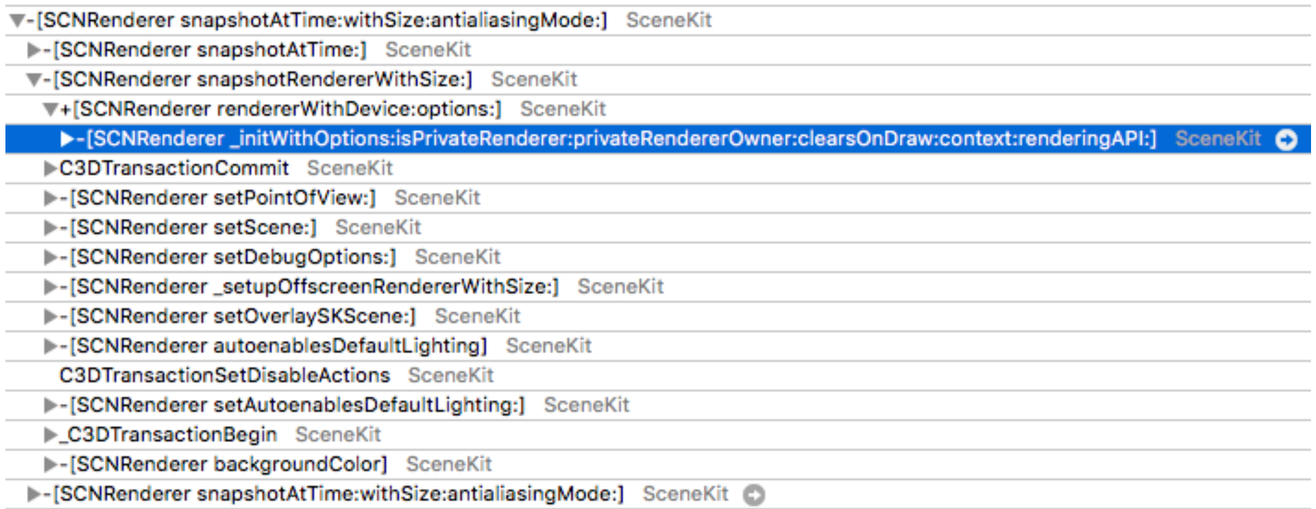
Оптимизации: раунд 1!

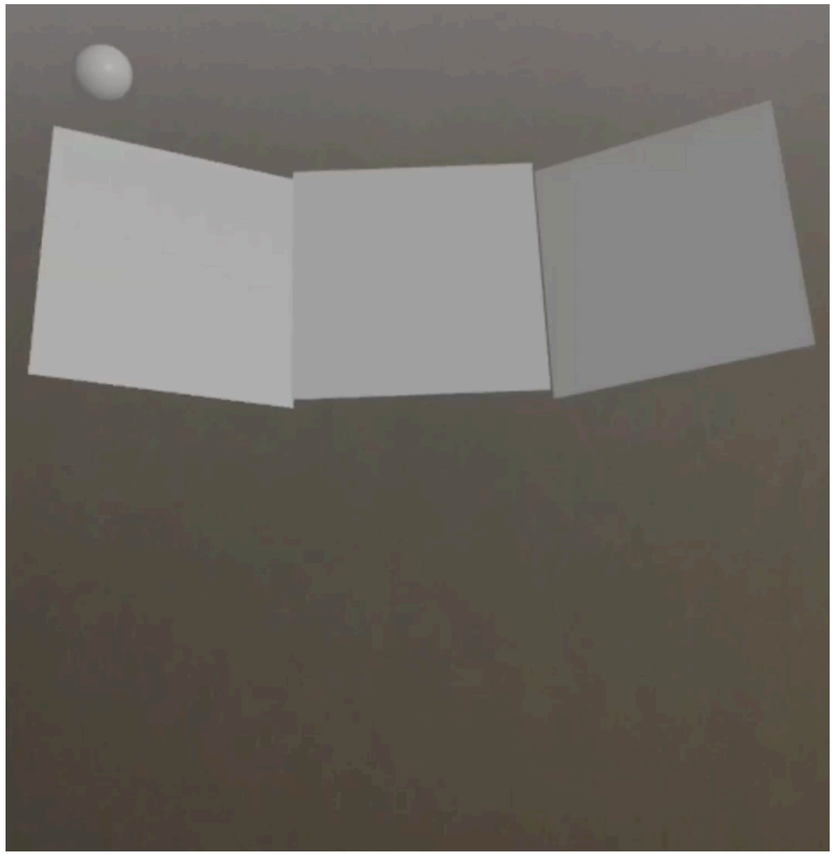


Оптимизации: раунд 1.5!

Оптимизации: раунд 1.5!

Оптимизации: раунд 2!

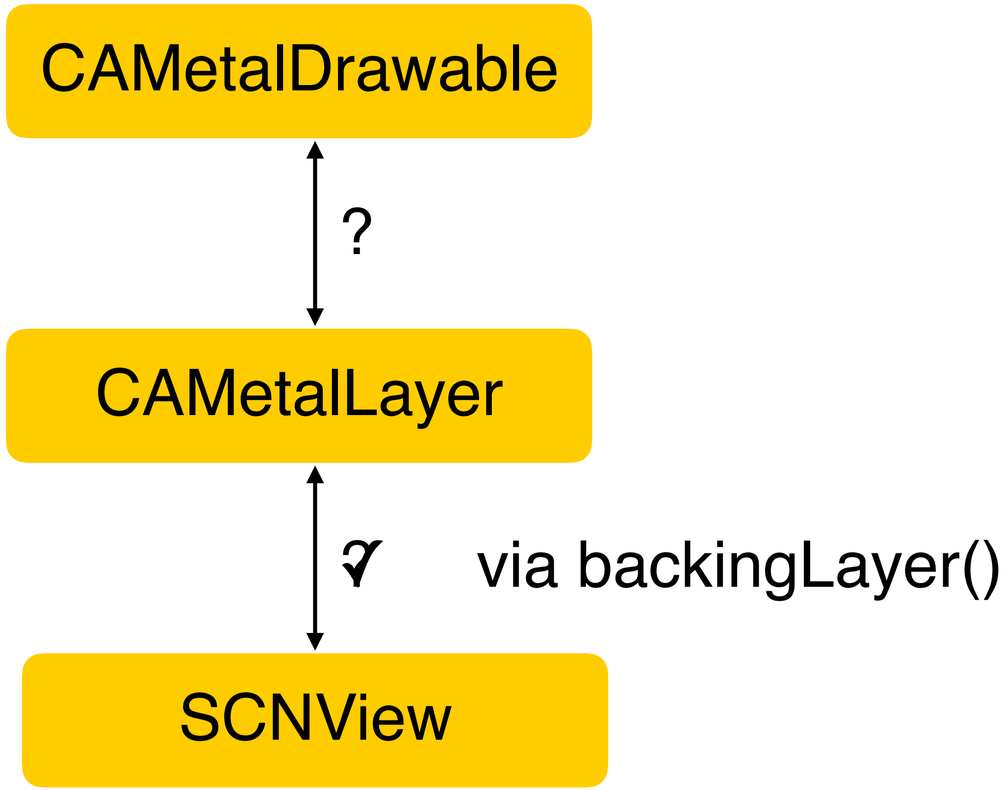




Причина

Следствие

Оптимизации: раунд 2!



Пул CAMetalLayer’a

Drawable 0

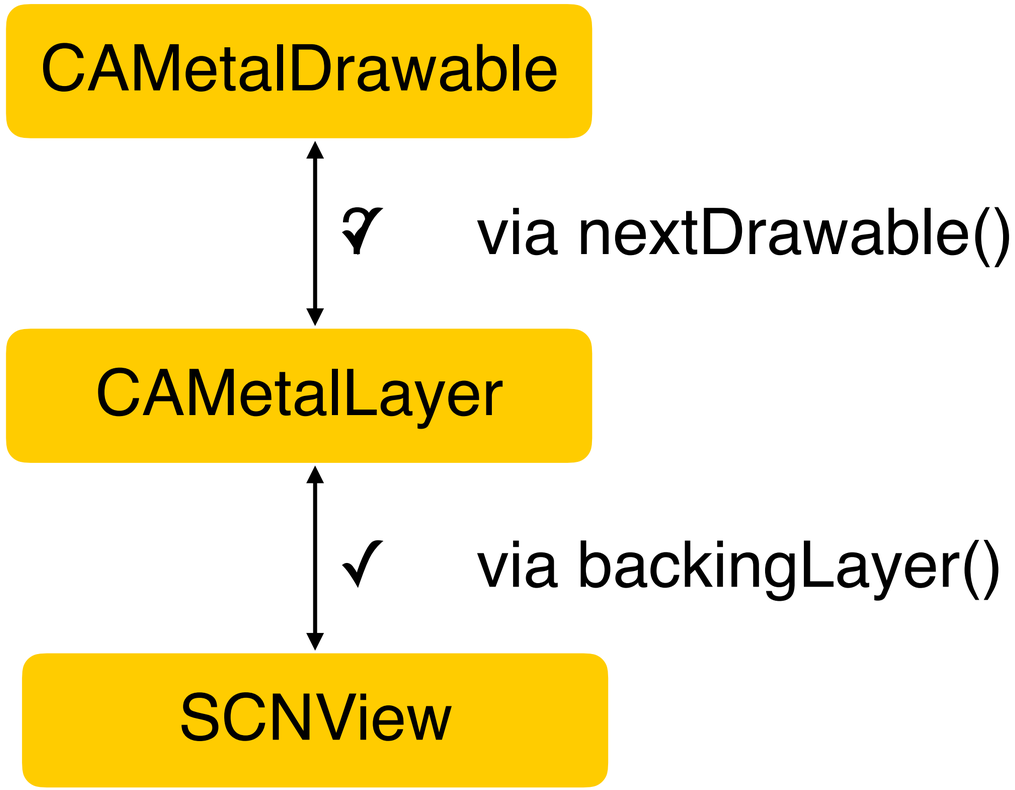
Drawable 1

Drawable 2

Drawable 3

func present()

Оптимизации: раунд 2!



Оптимизации: раунд 2!

CVPixelBufferCreate(… width:height:pixelFormat:

CVPixelBufferLockBaseAddress(_

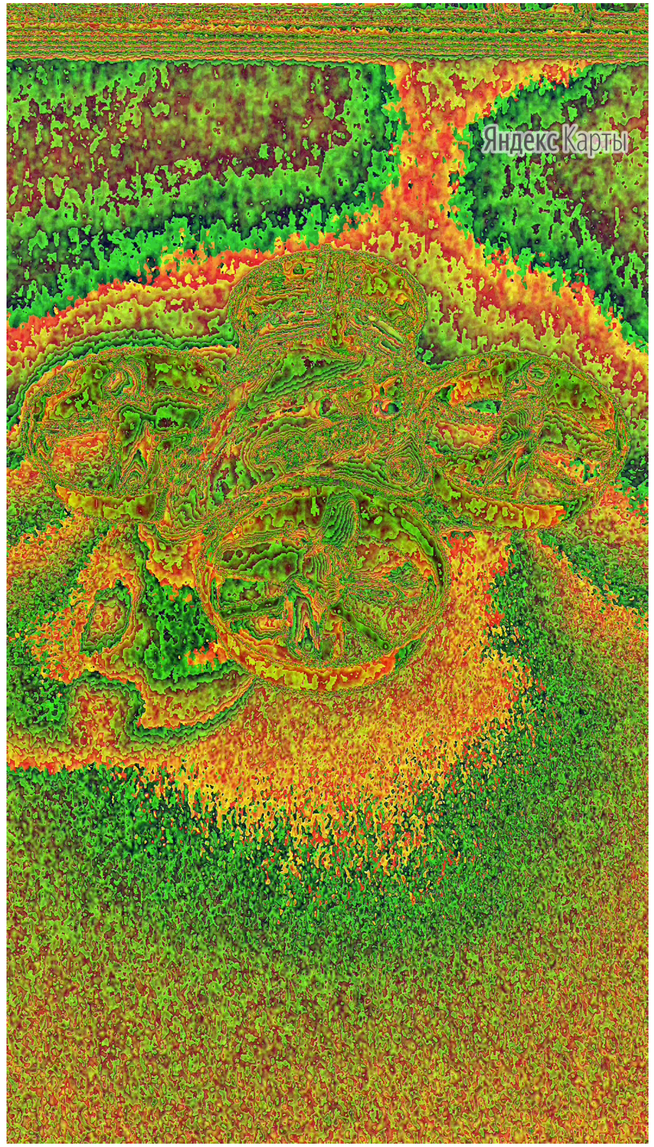
CVPixelBufferGetBaseAddress(_

CAMetalDrawable da var texture: tesPerRow:…

func draw(image: l func getBytes(data: …)

CVPixelBufferUnlockBaseAddress(_

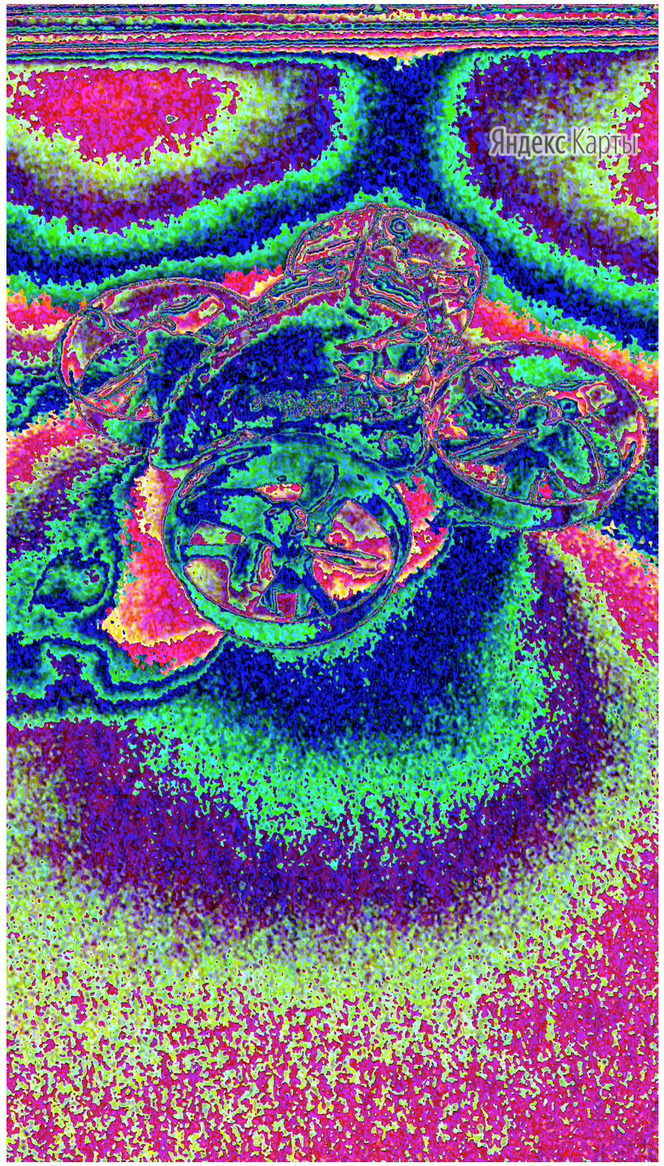
Оптимизации, раунд 3?



Оптимизации, раунд 3!

Оптимизации, раунд 3!

Оптимизации, раунд 3!



Оптимизации, раунд 3!

Оптимизации, раунд 3!

Retrieving Display-Related Traits

```
var displayScale: CGFloat
```

The display scale of the trait collection.

```
var displayGamut: UIDisplayGamut
```

The gamut of the current display.

```
enum UIDisplayGamut
```

Constants indicating the gamut of the current display.

Третий подъезд

SceneKit на практике

- Загрузка моделей
- Уменьшение размера ресурсов
- Констрейнты
- Зеркальные отражения
- Видеотекстуры
- Модификаторы шейдеров
- JavaScript Action
- Запись видео

Немного статистики

За последние выходные:

- Пройдено > 2000 км
- Показано > 3000 объектов
- Записано > 1000 видео

Yandex

Подписывайтесь на наш блог!

https://medium.com/yandex-maps-ios/