Projet 12

Parcours Développeur d'Application iOS

Réalisez un projet libre à impact social

Par Poudja CANESSANE
22/12/20

SOMMAIRE

PRÉSENTATION DU CONTEXTE

- Dernier projet
- Mettre nos connaissances au profit des besoins d’autrui
- Impact social

PRÉSENTATION DU PROJET

- 2 sujets possibles:
    - Créer une app iOS
    - Rédiger une présentation

- Livrables communs:
    - Note d'intention
    - Bilan

=> Choisi présentation: ***Comment développer un jeu 2D simple avec SpriteKit?***

OUTILS

Xcode

git

GitHub

Desktop

Google Slides

DÉMONSTRATION

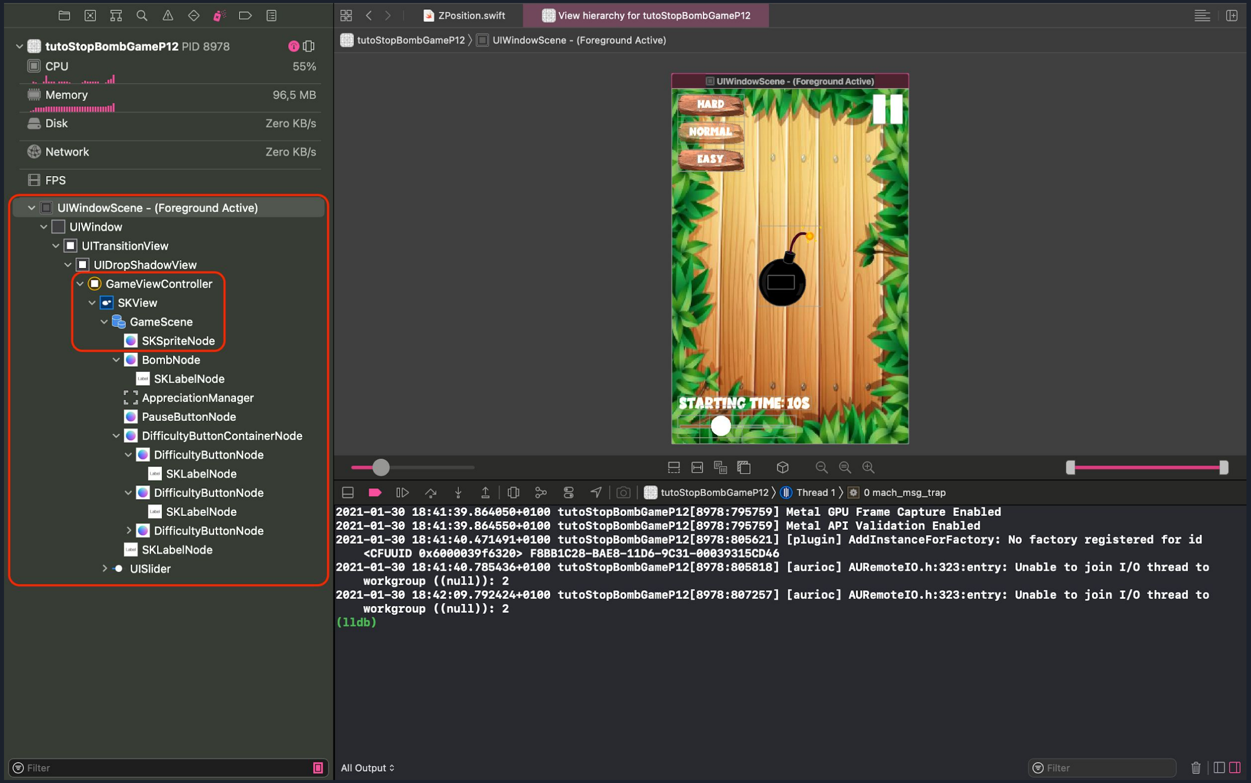
Lien repo:
https://github.com/Poudja-CANESSANE/tutoStopBombGameP12

Qu’est-ce que SpriteKit

- Framework pour coder des jeux iOS en 2D
- Hiérarchie parent/enfant

  => UIViewController/SKView/SKScene/SKNode
- Possibilité de le coupler avec UIKit

Qu'est-ce que SpriteKit


ZPosition.swift
View hierarchy for tutoStopBombGameP12
tutoStopBombGameP12 〉 UIWindowScene - (Foreground Active)
UIWindowScene - (Foreground Active)
HARD
NORMAL
EASY
STARTING TIME: 10S
tutoStopBombGameP12 〉 Thread 1 〉 0 mach_msg_trap
All Output
Filter

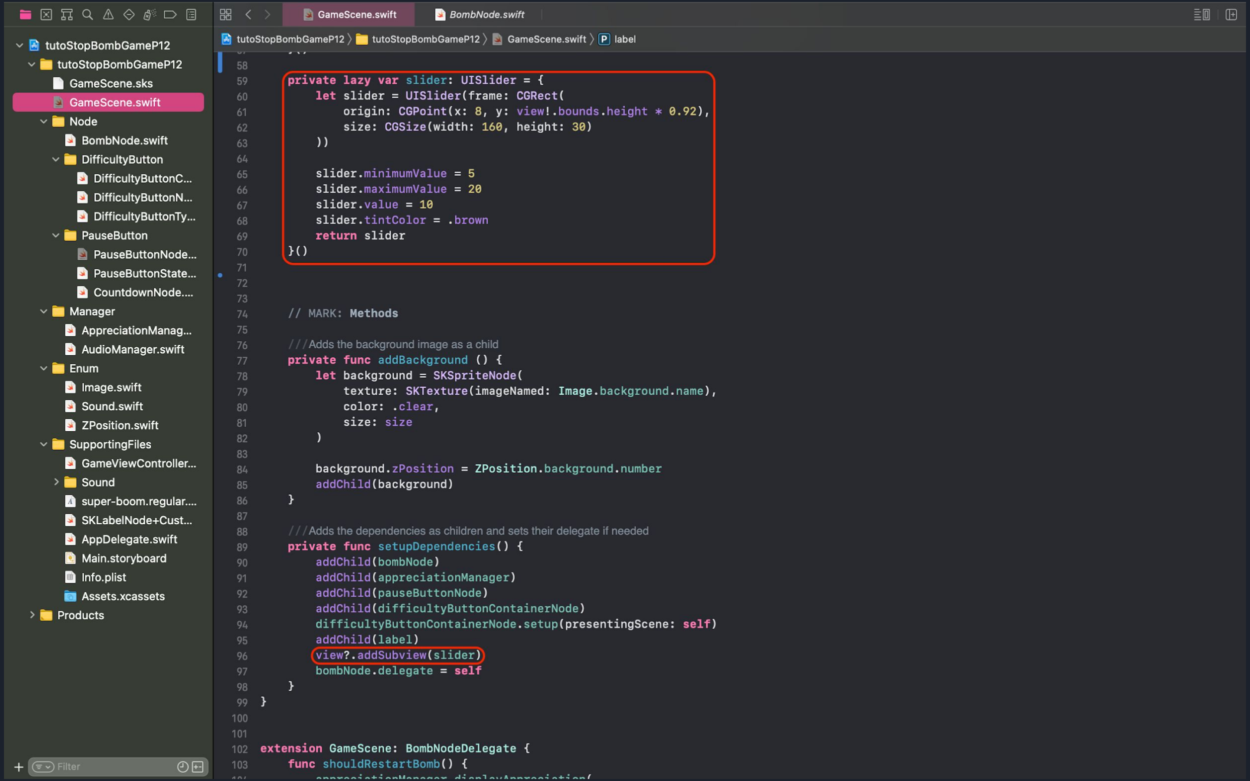
```
tutoStopBombGameP12 PID 8978
CPU          55%
Memory       96,5 MB
Disk         Zero KB/s
Network      Zero KB/s
FPS
```

```
UIWindowScene - (Foreground Active)
  UIWindow
    UITransitionView
      UIDropShadowView
        GameViewController
          SKView
            GameScene
              SKSpriteNode
              BombNode
                SKLabelNode
              AppreciationManager
              PauseButtonNode
              DifficultyButtonContainerNode
                DifficultyButtonNode
                  SKLabelNode
                DifficultyButtonNode
                  SKLabelNode
                DifficultyButtonNode
              SKLabelNode
          UISlider
```

```
2021-01-30 18:41:39.864050+0100 tutoStopBombGameP12[8978:795759] Metal GPU Frame Capture Enabled
2021-01-30 18:41:39.864550+0100 tutoStopBombGameP12[8978:795759] Metal API Validation Enabled
2021-01-30 18:41:40.471491+0100 tutoStopBombGameP12[8978:805621] [plugin] AddInstanceForFactory: No factory registered for id
    <CFUUID 0x6000039f6320> F8BB1C28-BAE8-11D6-9C31-00039315CD46
2021-01-30 18:41:40.785436+0100 tutoStopBombGameP12[8978:805818] [aurioc] AURemoteIO.h:323:entry: Unable to join I/O thread to
    workgroup ((null)): 2
2021-01-30 18:42:09.792424+0100 tutoStopBombGameP12[8978:807257] [aurioc] AURemoteIO.h:323:entry: Unable to join I/O thread to
    workgroup ((null)): 2
(lldb)
```

Exemple hiérarchie

Qu’est-ce que SpriteKit



```
    private lazy var slider: UISlider = {
        let slider = UISlider(frame: CGRect(
            origin: CGPoint(x: 8, y: view!.bounds.height * 0.92),
            size: CGSize(width: 160, height: 30)
        ))

        slider.minimumValue = 5
        slider.maximumValue = 20
        slider.value = 10
        slider.tintColor = .brown
        return slider
    }()



    // MARK: Methods

    ///Adds the background image as a child
    private func addBackground () {
        let background = SKSpriteNode(
            texture: SKTexture(imageNamed: Image.background.name),
            color: .clear,
            size: size
        )

        background.zPosition = ZPosition.background.number
        addChild(background)
    }

    ///Adds the dependencies as children and sets their delegate if needed
    private func setupDependencies() {
        addChild(bombNode)
        addChild(appreciationManager)
        addChild(pauseButtonNode)
        addChild(difficultyButtonContainerNode)
        difficultyButtonContainerNode.setup(presentingScene: self)
        addChild(label)
        view?.addSubview(slider)
        bombNode.delegate = self
    }
}


extension GameScene: BombNodeDelegate {
    func shouldRestartBomb() {
```

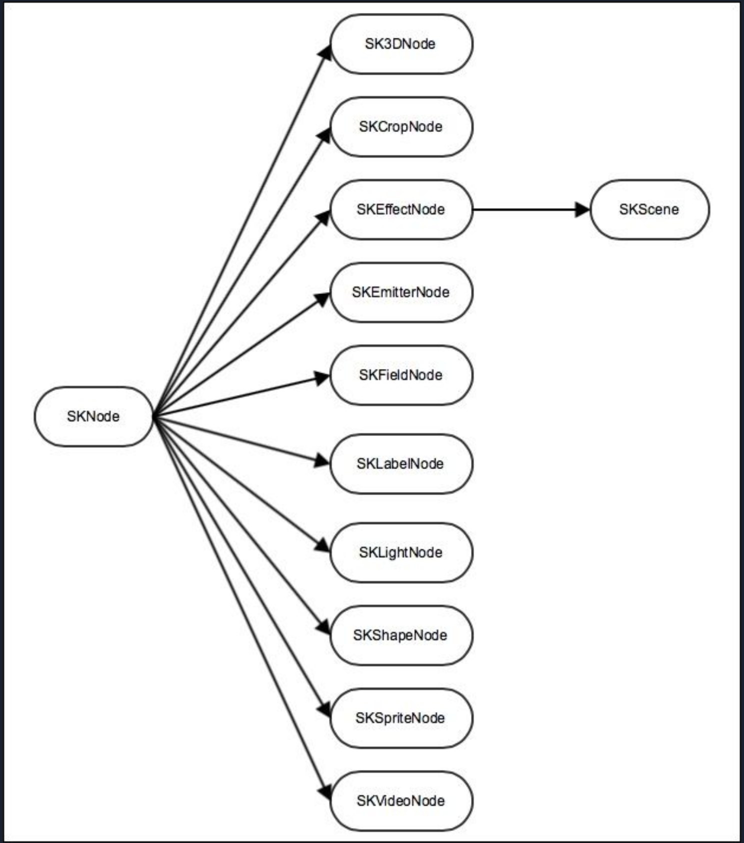
Exemple SpriteKit + UIKit

Les principales méthodes de SKScene à connaître

- Cycle de vie du jeu:
  - didMove()
  - update()

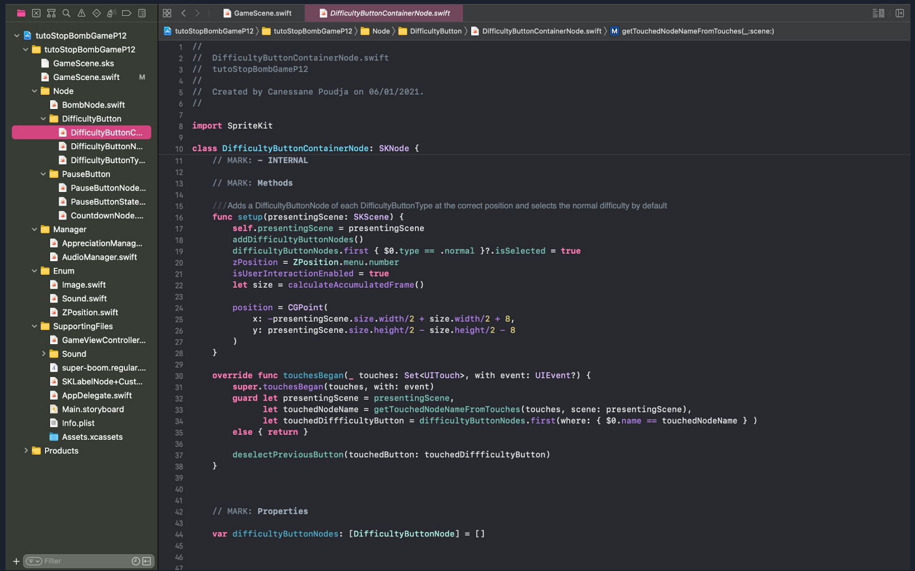
- Détecter une touche:
  - touchesBegan()
  - touchesMoved()
  - touchesEnded()
  - touchesCancelled()

Les différents types de nodes

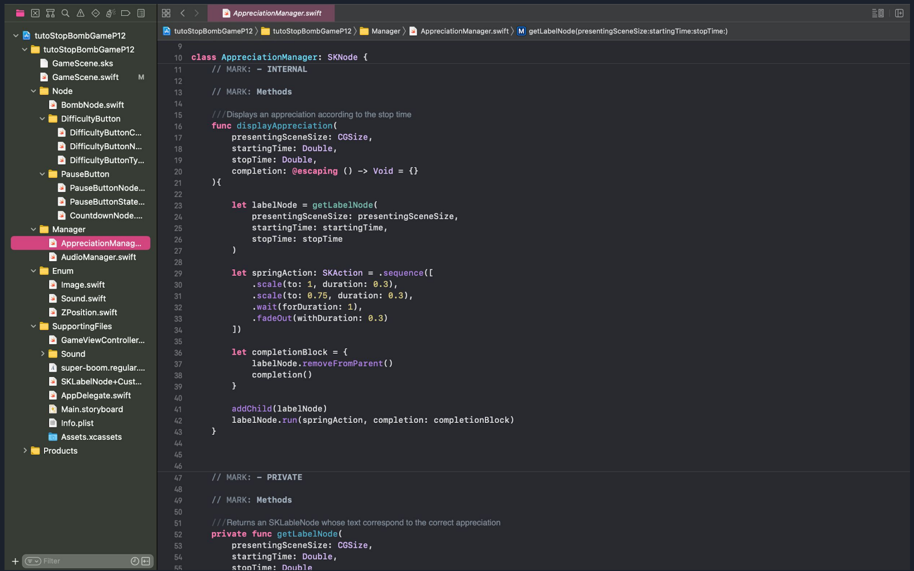


Les différents types de nodes

- SKNode:
  - ne dessine aucun contenu lui-même
  - comme UIStackView utilisé comme conteneur
  - permet d’effectuer une action sur tous les nodes enfants d’un coup (positionnement, suppression, modification alpha…)



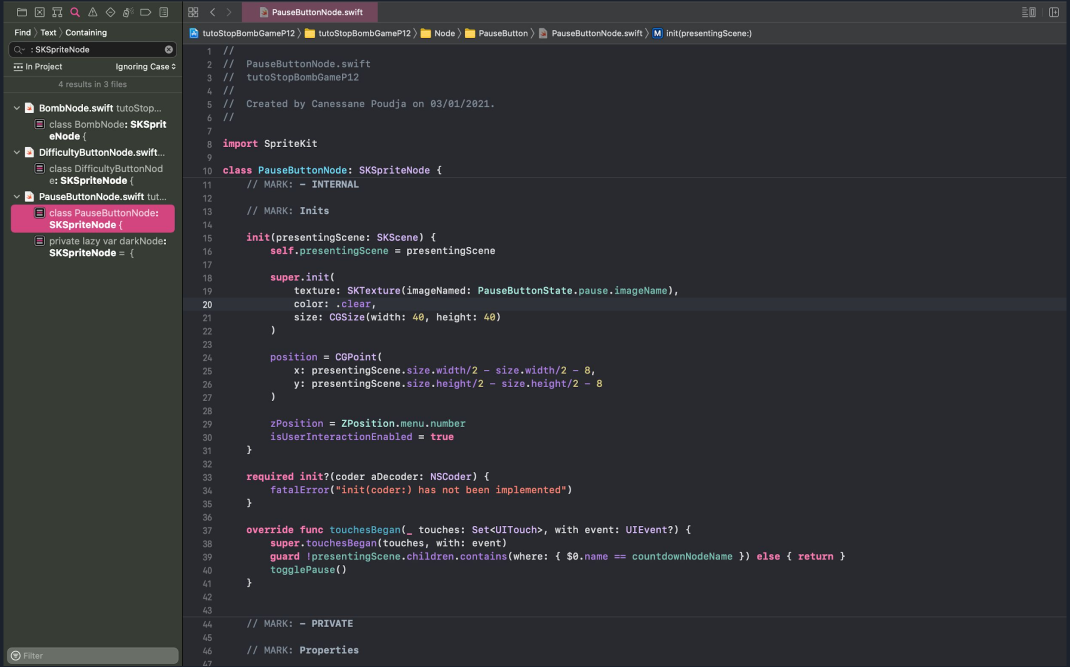
```swift
//
//  DifficultyButtonContainerNode.swift
//  tutoStopBombGameP12
//
//  Created by Canessane Poudja on 06/01/2021.
//

import SpriteKit

class DifficultyButtonContainerNode: SKNode {
    // MARK: - INTERNAL

    // MARK: Methods

    ///Adds a DifficultyButtonNode of each DifficultyButtonType at the correct position and selects the normal difficulty by default
    func setup(presentingScene: SKScene) {
        self.presentingScene = presentingScene
        addDifficultyButtonNodes()
        difficultyButtonNodes.first { $0.type == .normal }?.isSelected = true
        zPosition = ZPosition.menu.number
        isUserInteractionEnabled = true
        let size = calculateAccumulatedFrame()

        position = CGPoint(
            x: -presentingScene.size.width/2 + size.width/2 + 8,
            y: presentingScene.size.height/2 - size.height/2 - 8
        )
    }

    override func touchesBegan(_ touches: Set<UITouch>, with event: UIEvent?) {
        super.touchesBegan(touches, with: event)
        guard let presentingScene = presentingScene,
              let touchedNodeName = getTouchedNodeNameFromTouches(touches, scene: presentingScene),
              let touchedDiffficultyButton = difficultyButtonNodes.first(where: { $0.name == touchedNodeName } )
        else { return }

        deselectPreviousButton(touchedButton: touchedDiffficultyButton)
    }



    // MARK: Properties

    var difficultyButtonNodes: [DifficultyButtonNode] = []


```



```swift

class AppreciationManager: SKNode {
    // MARK: - INTERNAL

    // MARK: Methods

    ///Displays an appreciation according to the stop time
    func displayAppreciation(
        presentingSceneSize: CGSize,
        startingTime: Double,
        stopTime: Double,
        completion: @escaping () -> Void = {}
    ){

        let labelNode = getLabelNode(
            presentingSceneSize: presentingSceneSize,
            startingTime: startingTime,
            stopTime: stopTime
        )

        let springAction: SKAction = .sequence([
            .scale(to: 1, duration: 0.3),
            .scale(to: 0.75, duration: 0.3),
            .wait(forDuration: 1),
            .fadeOut(withDuration: 0.3)
        ])

        let completionBlock = {
            labelNode.removeFromParent()
            completion()
        }

        addChild(labelNode)
        labelNode.run(springAction, completion: completionBlock)
    }



    // MARK: - PRIVATE

    // MARK: Methods

    ///Returns an SKLableNode whose text correspond to the correct appreciation
    private func getLabelNode(
        presentingSceneSize: CGSize,
        startingTime: Double,
        stopTime: Double
```

Les différents types de nodes

- SKSpriteNode:
  - dessine du contenu graphique (texture ou couleur)
  - utilisé le plus souvent



Find 〉 Text 〉 Containing

: SKSpriteNode

In Project — Ignoring Case

4 results in 3 files

- BombNode.swift tutoStop...
  - class BombNode: **SKSpriteNode** {
- DifficultyButtonNode.swift...
  - class DifficultyButtonNode: **SKSpriteNode** {
- PauseButtonNode.swift tut...
  - class PauseButtonNode: **SKSpriteNode** {
  - private lazy var darkNode: **SKSpriteNode** = {

PauseButtonNode.swift

tutoStopBombGameP12 〉 tutoStopBombGameP12 〉 Node 〉 PauseButton 〉 PauseButtonNode.swift 〉 init(presentingScene:)

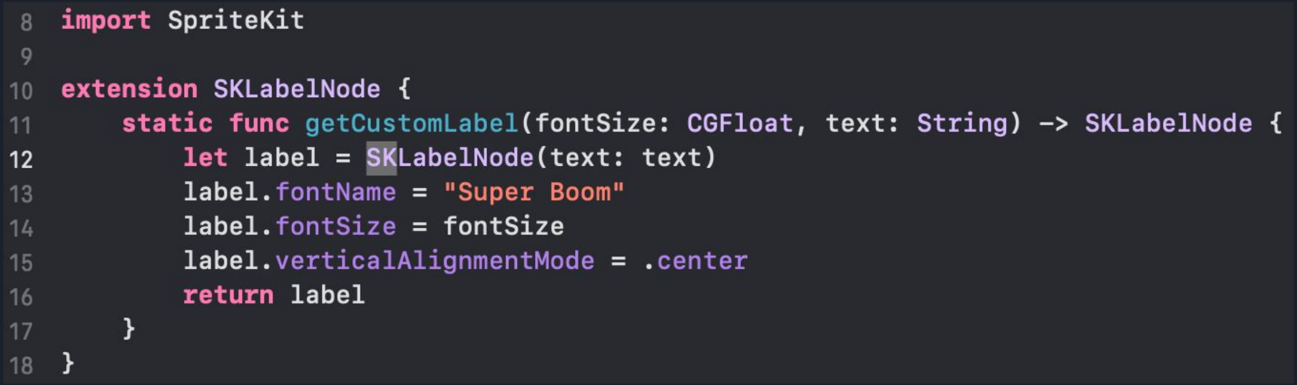
```swift
//
//  PauseButtonNode.swift
//  tutoStopBombGameP12
//
//  Created by Canessane Poudja on 03/01/2021.
//

import SpriteKit

class PauseButtonNode: SKSpriteNode {
    // MARK: - INTERNAL

    // MARK: Inits

    init(presentingScene: SKScene) {
        self.presentingScene = presentingScene

        super.init(
            texture: SKTexture(imageNamed: PauseButtonState.pause.imageName),
            color: .clear,
            size: CGSize(width: 40, height: 40)
        )

        position = CGPoint(
            x: presentingScene.size.width/2 - size.width/2 - 8,
            y: presentingScene.size.height/2 - size.height/2 - 8
        )

        zPosition = ZPosition.menu.number
        isUserInteractionEnabled = true
    }

    required init?(coder aDecoder: NSCoder) {
        fatalError("init(coder:) has not been implemented")
    }

    override func touchesBegan(_ touches: Set<UITouch>, with event: UIEvent?) {
        super.touchesBegan(touches, with: event)
        guard !presentingScene.children.contains(where: { $0.name == countdownNodeName }) else { return }
        togglePause()
    }


    // MARK: - PRIVATE

    // MARK: Properties
```

Filter

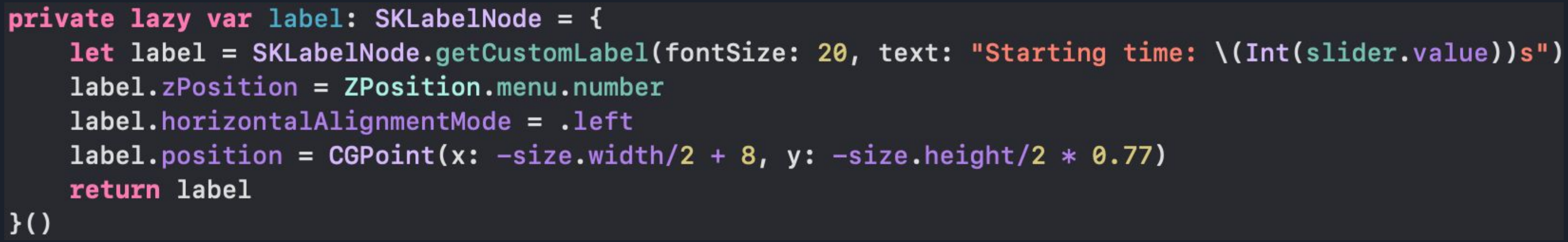
```swift
private lazy var darkNode: SKSpriteNode =  {
    let darkNode = SKSpriteNode(
        color: .black,
        size: presentingScene.size
    )

    darkNode.alpha = 0.7
    return darkNode
}()
```

Les différents types de nodes

- SKLabelNode: similaire à UILabel



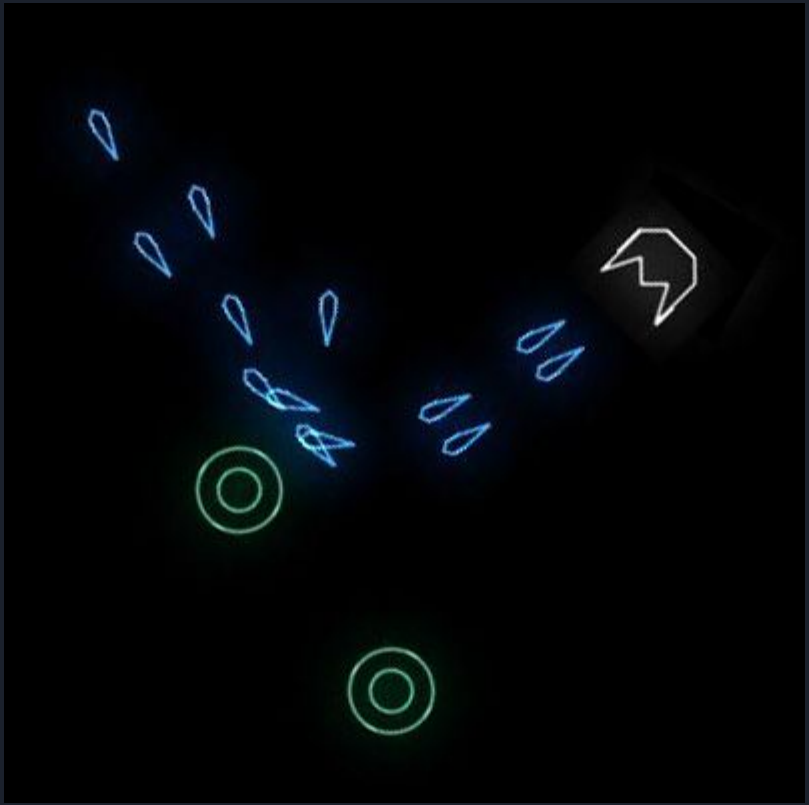
```swift
import SpriteKit

extension SKLabelNode {
    static func getCustomLabel(fontSize: CGFloat, text: String) -> SKLabelNode {
        let label = SKLabelNode(text: text)
        label.fontName = "Super Boom"
        label.fontSize = fontSize
        label.verticalAlignmentMode = .center
        return label
    }
}
```



```swift
private lazy var label: SKLabelNode = {
    let label = SKLabelNode.getCustomLabel(fontSize: 20, text: "Starting time: \(Int(slider.value))s")
    label.zPosition = ZPosition.menu.number
    label.horizontalAlignmentMode = .left
    label.position = CGPoint(x: -size.width/2 + 8, y: -size.height/2 * 0.77)
    return label
}()
```

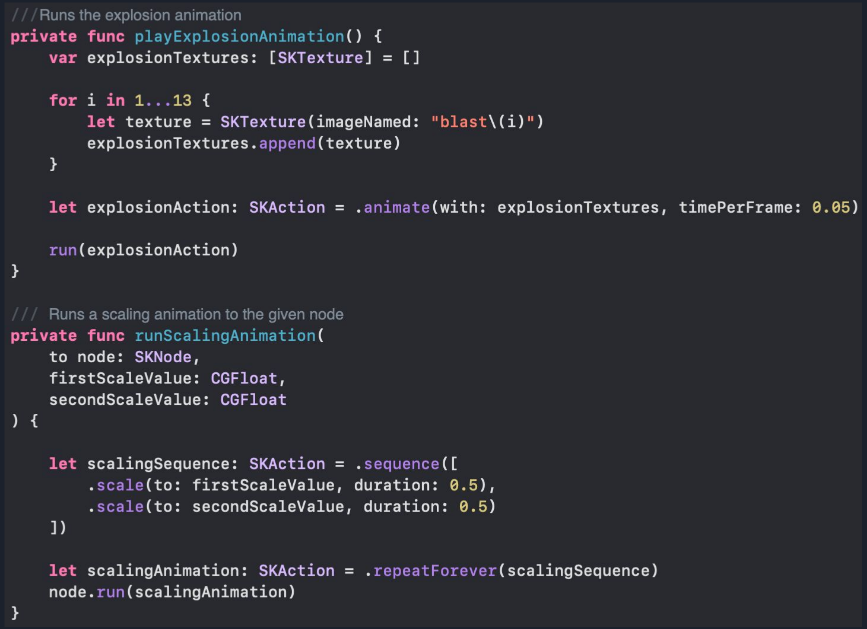
Les différents types de nodes

- SKShapeNode:
  - SKNode de la forme souhaitée
  - Avantage: forme non pixelisée contrairement à SKTexture si l’image est trop petite ou de mauvaise qualité

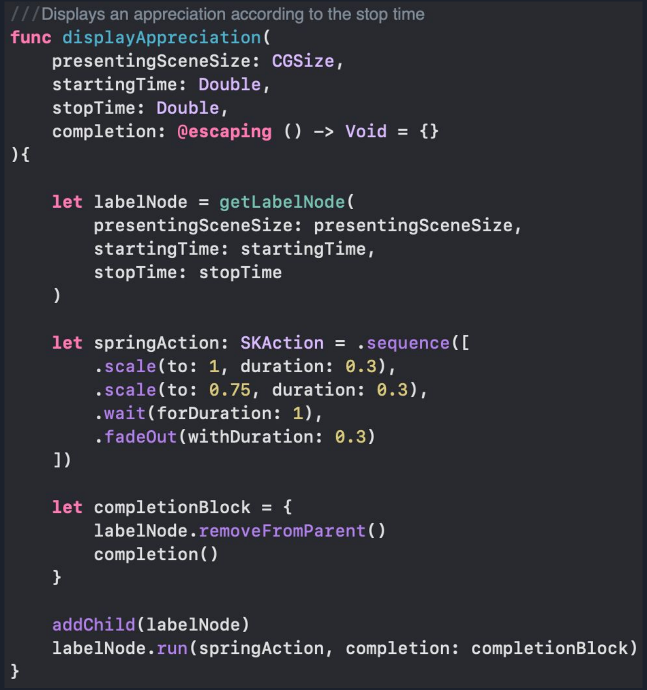


Les animations

- Existe une 50aine de SKAction
- Pour annuler l'effet d'une animation (comme *node.fadeOut()*): appliquer l'animation inverse (c-à-d *node.fadeIn()* et non *node.alpha = 1*)



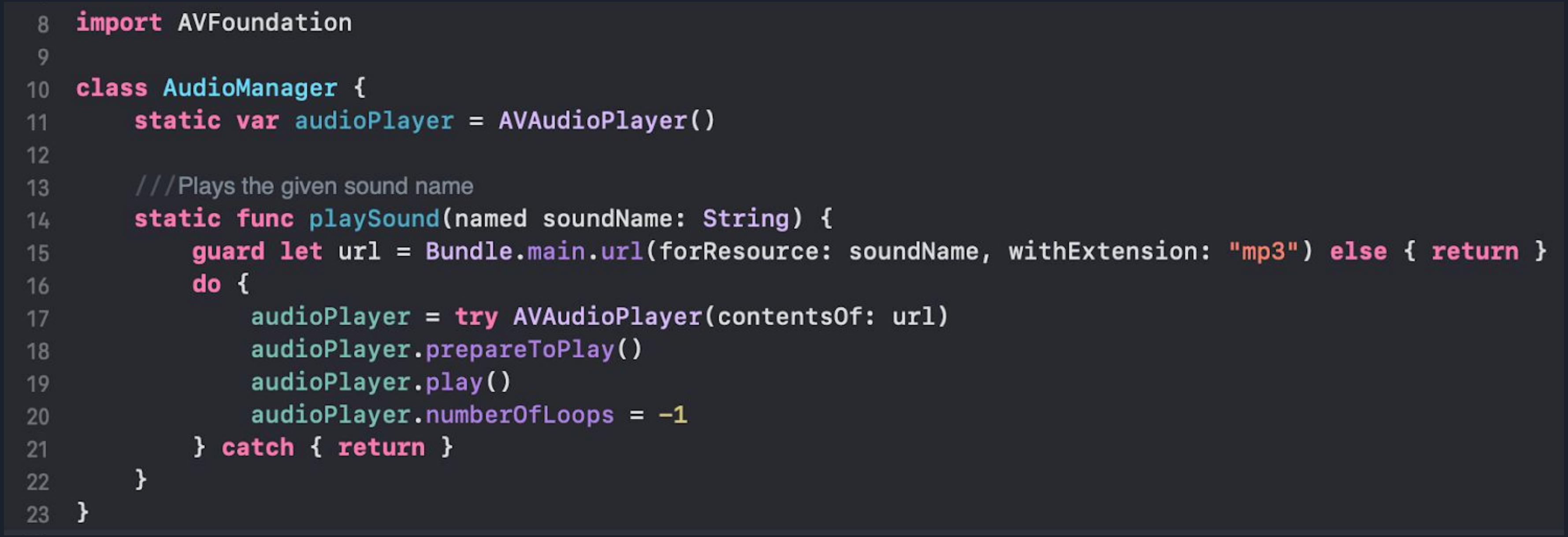
```swift
///Runs the explosion animation
private func playExplosionAnimation() {
    var explosionTextures: [SKTexture] = []

    for i in 1...13 {
        let texture = SKTexture(imageNamed: "blast\(i)")
        explosionTextures.append(texture)
    }

    let explosionAction: SKAction = .animate(with: explosionTextures, timePerFrame: 0.05)

    run(explosionAction)
}

///  Runs a scaling animation to the given node
private func runScalingAnimation(
    to node: SKNode,
    firstScaleValue: CGFloat,
    secondScaleValue: CGFloat
) {

    let scalingSequence: SKAction = .sequence([
        .scale(to: firstScaleValue, duration: 0.5),
        .scale(to: secondScaleValue, duration: 0.5)
    ])

    let scalingAnimation: SKAction = .repeatForever(scalingSequence)
    node.run(scalingAnimation)
}
```



```swift
///Displays an appreciation according to the stop time
func displayAppreciation(
    presentingSceneSize: CGSize,
    startingTime: Double,
    stopTime: Double,
    completion: @escaping () -> Void = {}
){

    let labelNode = getLabelNode(
        presentingSceneSize: presentingSceneSize,
        startingTime: startingTime,
        stopTime: stopTime
    )

    let springAction: SKAction = .sequence([
        .scale(to: 1, duration: 0.3),
        .scale(to: 0.75, duration: 0.3),
        .wait(forDuration: 1),
        .fadeOut(withDuration: 0.3)
    ])

    let completionBlock = {
        labelNode.removeFromParent()
        completion()
    }

    addChild(labelNode)
    labelNode.run(springAction, completion: completionBlock)
}
```

Les sons et musiques

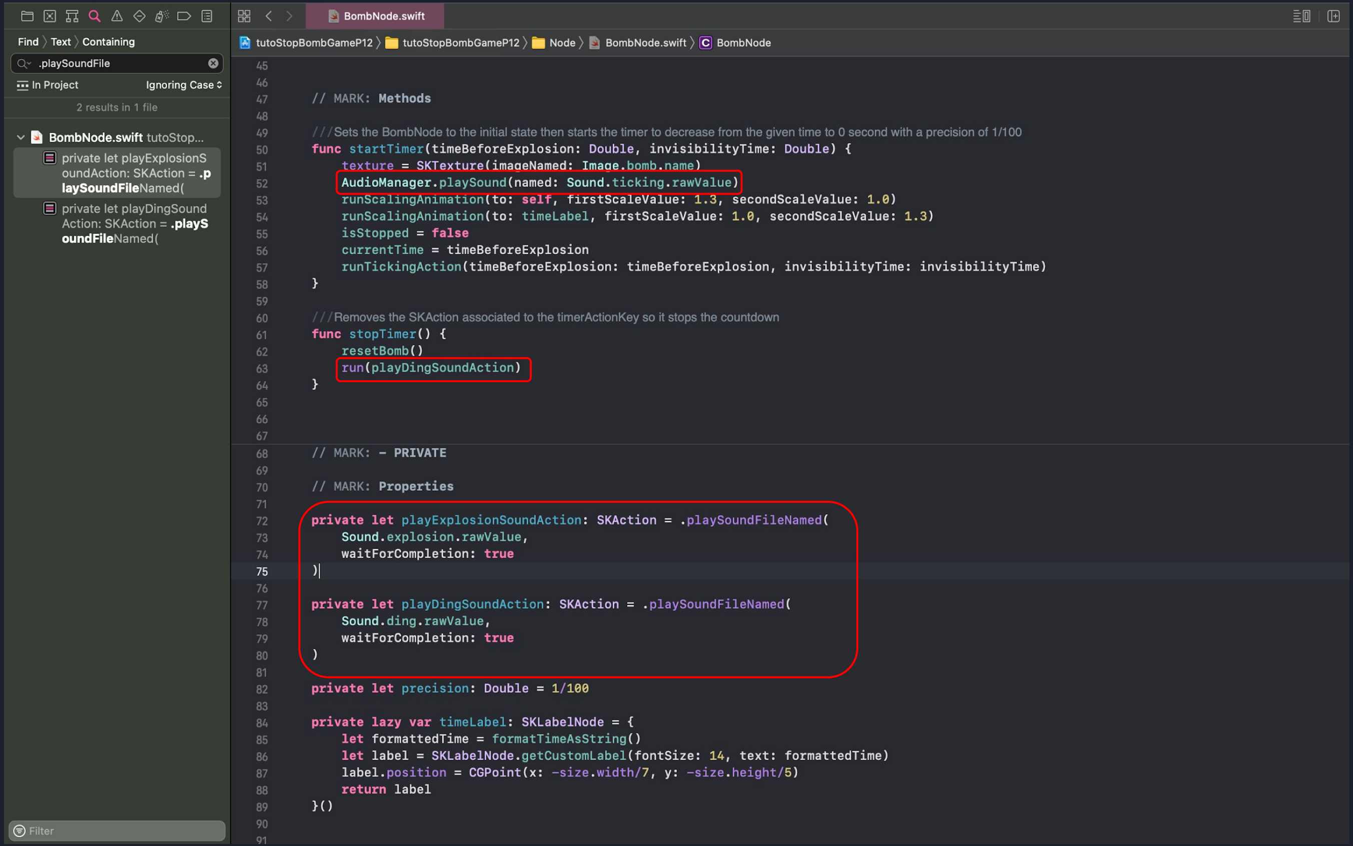
Pour les musiques ou sons en boucle



```swift
import AVFoundation

class AudioManager {
    static var audioPlayer = AVAudioPlayer()

    ///Plays the given sound name
    static func playSound(named soundName: String) {
        guard let url = Bundle.main.url(forResource: soundName, withExtension: "mp3") else { return }
        do {
            audioPlayer = try AVAudioPlayer(contentsOf: url)
            audioPlayer.prepareToPlay()
            audioPlayer.play()
            audioPlayer.numberOfLoops = -1
        } catch { return }
    }
}
```

Les sons et musiques

Pour les sons ponctuels



Find 〉 Text 〉 Containing

.playSoundFile

In Project   Ignoring Case

2 results in 1 file

BombNode.swift tutoStop...

private let playExplosionSoundAction: SKAction = .playSoundFileNamed(

private let playDingSoundAction: SKAction = .playSoundFileNamed(

BombNode.swift

tutoStopBombGameP12 〉 tutoStopBombGameP12 〉 Node 〉 BombNode.swift 〉 BombNode

```swift


    // MARK: Methods

    ///Sets the BombNode to the initial state then starts the timer to decrease from the given time to 0 second with a precision of 1/100
    func startTimer(timeBeforeExplosion: Double, invisibilityTime: Double) {
        texture = SKTexture(imageNamed: Image.bomb.name)
        AudioManager.playSound(named: Sound.ticking.rawValue)
        runScalingAnimation(to: self, firstScaleValue: 1.3, secondScaleValue: 1.0)
        runScalingAnimation(to: timeLabel, firstScaleValue: 1.0, secondScaleValue: 1.3)
        isStopped = false
        currentTime = timeBeforeExplosion
        runTickingAction(timeBeforeExplosion: timeBeforeExplosion, invisibilityTime: invisibilityTime)
    }

    ///Removes the SKAction associated to the timerActionKey so it stops the countdown
    func stopTimer() {
        resetBomb()
        run(playDingSoundAction)
    }



    // MARK: - PRIVATE

    // MARK: Properties

    private let playExplosionSoundAction: SKAction = .playSoundFileNamed(
        Sound.explosion.rawValue,
        waitForCompletion: true
    )

    private let playDingSoundAction: SKAction = .playSoundFileNamed(
        Sound.ding.rawValue,
        waitForCompletion: true
    )

    private let precision: Double = 1/100

    private lazy var timeLabel: SKLabelNode = {
        let formattedTime = formatTimeAsString()
        let label = SKLabelNode.getCustomLabel(fontSize: 14, text: formattedTime)
        label.position = CGPoint(x: -size.width/7, y: -size.height/5)
        return label
    }()
```

Filter

CONCLUSION

- Connaître SpriteKit est un atout, pas une perte de temps
- Jeux mobile à succès: Flappy Bird, Candy Crush, Doodle Jump, Angry Birds
- Objectif: gameplay addictif