

# **PG5600**

## **IOS PROGRAMMING**

### **FORELESNING 9**

**SIST GANG**

**APP EXTENSIONS**

**AGENDA**

# **MEDIA**

» Enkle 2D spill og Swift

» Video

» Lyd

**SPRITE KIT**

# HVA ER DET?

- » Sprite Kit er et kraftfullt men enkelt rammeverk for iOS/Mac som hjelper oss å lage 2D spill
- » Ikke nødvendig å laste ned noe mer eller ha eksterne avhengigheter
- » Man kan bruke det sammen med andre iOS APIs
- » Du kan bruke dine Swift og iOS kunnskaper
- » Det er skrevet av Apple
- » Det er gratis

# HVORDAN FUNGERER DET?

- » Man animerer bilder/textures/sprites
- » Bruker en loop for rendering av innhold til hver frame
- » Ditt spill bestemmer innholde i senen og hvordan innholde endrer seg i hver frame
- » Inneholder funksjonalitet som musikkavspilling, fysikk simulering
- » Kan kombineres med Core Animation

**Each frame**

`-update:`

SKScene  
evaluates actions

`-didEvaluateActions`

SKScene  
simulates physics

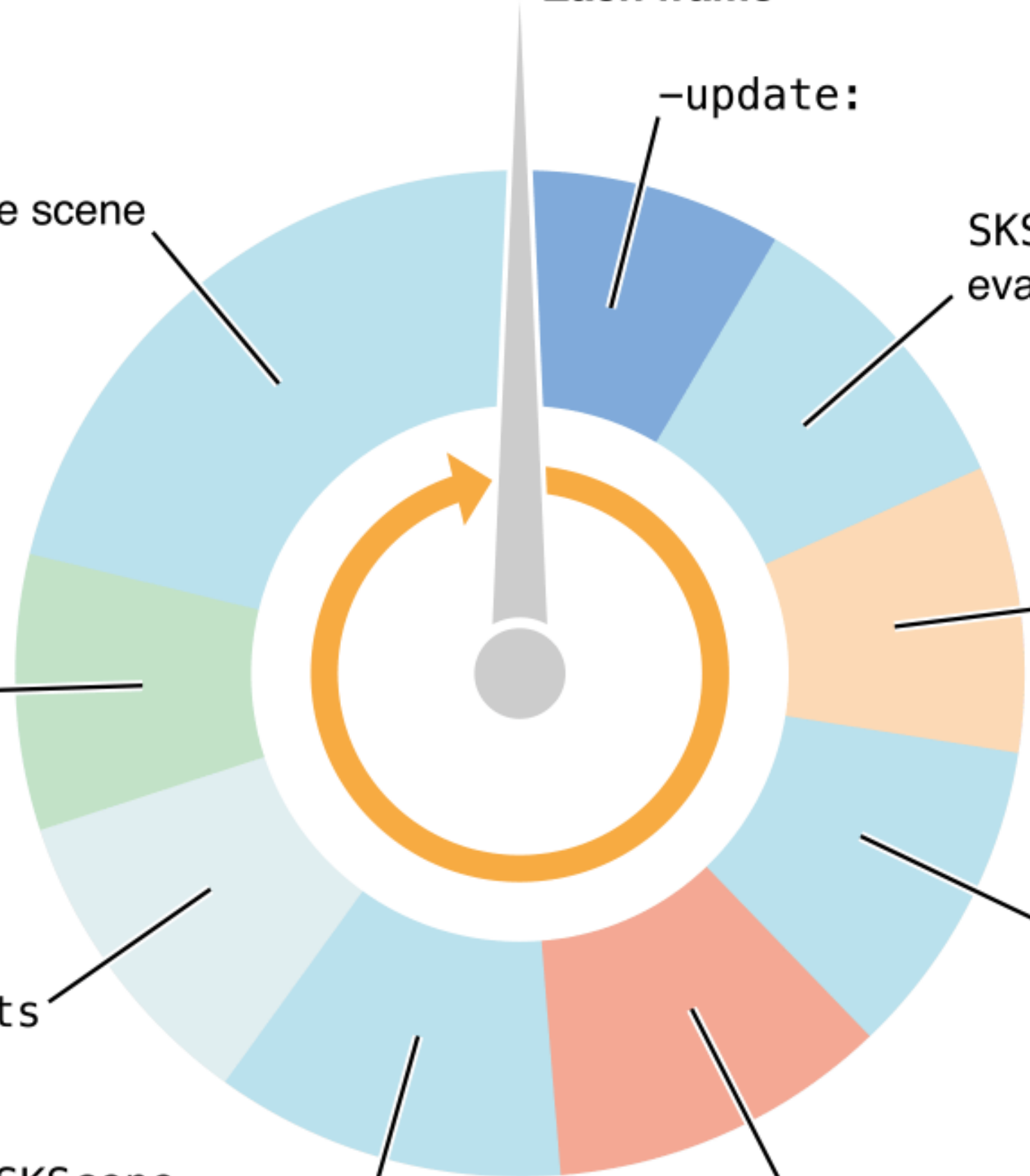
`-didSimulatePhysics`

SKScene  
applies constraints

`-didApplyConstraints`

`-didFinishUpdate`

SKView  
renders the scene



# SKVIEW

- » Står for animering og rendering
- » Plasseres inn i et window
- » Så legger man inn innholdet i SKView
- » Siden SKView er et view så kan det kombineres med flere views



Custom Class

Class

SKView

Module

None

Identity

Restoration ID

User Defined Runtime Attributes

Key Path	Type	Value

+

—

```
let skView = self.view as SKView
skView.showsFPS = true
skView.showsNodeCount = true
skView.ignoresSiblingOrder = true
```

# SKSCENE

- » Innholde i ditt spill er organisert i sener
- » En SKScene holder sprites og annet innhold som skal renderes
- » SKScene implementerer per-frame logikk og innholdsprosessering
- » Til en hver tid så presenterer SKView en SKScene
- » Så lenge SKScene vises så kjører innholdet automatisk

```
class GameScene: SKScene {  
    override func didMoveToView(view: SKView) {}  
    override func touchesBegan(touches: NSSet, withEvent event: UIEvent) {}  
    override func touchesMoved(touches: NSSet, withEvent event: UIEvent) {}  
    override func touchesEnded(touches: NSSet, withEvent event: UIEvent) {}  
    override func update(currentTime: CTimeInterval) {}  
}
```

```
// Init en GameScene klassen vi definerte i forrige slide  
let scene = GameScene(size: view.bounds.size)  
scene.scaleMode = .AspectFill  
skView.presentScene(scene)
```

# SKNODE

- » SKScene arver fra SKNode
- » SKNode er byggeblokkene for alt innhold
- » En SKScene oppfører seg som en root node i et tre av noder
- » En scene og dets barn, bestemmer hvilke innhold som blir tegnet og hvordan det blir rendert

- » Hver nodes posisjon er spesifisert i det koordinatsystemet som foreldren definerer
- » Når en node er rotert så vil alle barn rotere også
- » Alle node objekter er responder objekter og de arver fra UIResponer

# SKSPRITENODE

- » Arver også fra SKNode
- » Brukes til å tegne texture bilde
- » En farget firkant
- » Et texture bilde blandet med en farge



```
let ship = SKSpriteNode(imageNamed: "Spaceship")
ship.position = CGPointMake(self.size.width/2, self.size.height/2);
[self addChild:ship];
```

# SKACTION

- » SKNodes til en SKScene blir animert ved hjelp av SKAction
- » Noen actions bruker kun en frame for animasjon, mens andre actions går over flere frames
- » Vanlig bruk er å animere endringer på en nodes properties
- » Du kan lage grupper av actions som kjører samtidig eller i sekvens

```
let titleLabel = SKLabelNode(fontNamed: "HelveticaNeue")
titleLabel.text = "Start spillet";
titleLabel.position = CGPointMake(CGRectGetMidX(self.frame), CGRectGetMidY(self.frame));
titleLabel.fontSize = 50.0;

let angle = CGFloat(-M_PI*2)
let rotate = SKAction.rotateByAngle(angle, duration: 5.0)
let repeat = SKAction.repeatActionForever(rotate)
titleLabel.runAction(repeat)

self.addChild(titleLabel)
```

# SKPHYSICSBODY

- » Brukes når noder skal samhandle med hverandre, for eksempel:
  - » Kolidere
  - » Gravisjon
- » SKPhysicsBody legges på noder i SKScenen din
- » En sene definerer global fysikk ved hjelp av SKPhysicsWorld

```
self.physicsBody = SKPhysicsBody(circleOfRadius: frame.size.width/2)
self.physicsBody?.friction = 0.0
self.physicsBody?.restitution = 1.0
self.physicsBody?.linearDamping = 0.0
self.physicsBody?.allowsRotation = false

self.physicsBody?.applyImpulse(CGVectorMake(15, -15))
```

# SKPHYSICSBODY PROPERTIES

- » `mass` - Bestemmer hvordan utvendig kraft påvirker body, og hvor mye moment den har når den koliderer
- » `friction` - Ruhet til body, den brukes til å kalkulere friksjonskraften bodyen gir andre bodier i nærheten
- » `linearDamping`, `angularDamping` - Kalkulerer friksjons på en body når den beveger seg rundt i verden, kan brukes til å simulere luft og vann friksjon

- » `restitution` - Hvor mye energi bodyen beholder under kollisjon
- » `affectedByGravity` - Avgjør om simulering utøver en gravitasjonskraft på body
- » `allowsRotation` - Avgjør om krafter kan formidle vinkelhastighet på body

# SKPHYSICSWORLD PROPERTIES

- » gravity - Gjør at bodies i siumuleringen får akselasjon
- » speed - Sier i hvilke fart siumuleringen kjører



# SKTRANSITION

- » Brukes til å animere vekslingen mellom to SKScene som ligger på SKView
- » Si for eksempel at du skal veksle mellom:
  - » Laster sene
  - » Deretter en meny
  - » Så spill senen

```
let scene = GameScene(size: self.size)
let transition = SKTransition.pushWithDirection(.Up, duration: 1)
self.view?.presentScene(scene, transition: transition)
```

# OPPSUMMERT

1. SKView - Er subklasse av UIView.
2. SKScene - En spill sene som en tittel, instilling
3. SKTransition - Animerer mellom sener
4. SKSpriteNode - Textur bilde/farget rektangle
5. SKLabelNode - En node med en linje tekst
6. SKAction - Animering av noder

**DEMO**

**DET FINNES MANGE  
FLERE KLASSER I**

**SPRITE KIT**

**SCENE KIT**

**3D RAMMEVERK FRA  
APPLE**

**FUNGERE SAMMEN MED  
SPRITE KIT**

**FOR ALTERNATIVER SE  
FOR EKSEMPEL**

**UNITY**

**BUILDER**



# CORE IMAGE

- » Core Image er et kraftfullt rammeverk som er en del av Apple sin SDK
- » Du kan enkelt bruke det til å legge på filter og effekter
- » Du kan velge om man skal bruke CPU eller GPU til å prosessere bildene
- » Eksempel på filter er farge invertering og cropping

# VIKTIGE KLASSER

`CImageContext` - Prosessering av et core image objekt  
gjøres i en `CImageContext`

`CUIImage` - Holder på bilddedata og kan lages av et  
`UIImage`, en bilde file eller av pixel data

`CIFilter` - Inneholder et sett med attributer som  
definerer hva slags filter som er representert

```
import AssetsLibrary

func applyFilter(image: CIImage) {
    let filter = CIFilter(name: "CIHueAdjust")
    filter.setValue(image, forKey: kCIInputImageKey)
    filter.setValue(100, forKey: kCIInputAngleKey)
    let context = CIContext(options:nil)

    let newImage = UIImage(CIImage: filter.outputImage)
    self.imageView.image = newImage
}
```

Filter oversikt

# HASTIGHET

- » I eksempelet på sist slide så brukte vi ikke `CImageContext` direkte
- » `UIImage(CIImage:)` lager automatisk et `CImageContext` og legger på filterne
- » Dette er ikke lurt å gjøre om man skal legge på filter realtime
- » Da må man bruke en `CImageContext`
- » `CImageContext` tar inne options som kan bestemme om det kjører på CPU eller GPU

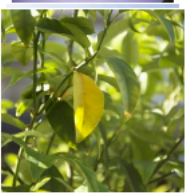
```
let context = CIContext(options:nil)

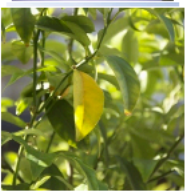
func applyFilter(image: CIImage) {
    let filter = CIFilter(name: "CIHueAdjust")
    filter.setValue(image, forKey: kCIInputImageKey)
    filter.setValue(100, forKey: kCIInputAngleKey)

    let cgimg = context.createCGImage(filter.outputImage, fromRect: filter.outputImage.extent())

    let newImage = UIImage(CGImage: cgimg)
    self.imageView.image = newImage
}
```

**HENTE OPP BILDER**

- 

Moments
- 

Camera Roll

5

```
@IBAction func loadPhoto(sender : AnyObject) {  
    let picker = UIImagePickerController()  
    picker.delegate = self  
    self.presentViewController(picker, animated: true, completion: nil)  
}  
  
// UIImagePickerControllerDelegate  
  
func imagePickerController(picker: UIImagePickerController!,  
    didFinishPickingMediaWithInfo info: NSDictionary!) {  
  
    self.dismissViewControllerAnimated(true, completion: nil);  
    let gotImage = info[UIImagePickerControllerOriginalImage] as UIImage  
    applyFilter(CIImage(image: gotImage))  
  
}
```



# LAGRE BILDER

```
@IBAction func savePhoto(sender: AnyObject) {  
    let imageToSave = UIImage(image: self.imageView.image)  
    let softwareContext = CIContext(options:[kCIContextUseSoftwareRenderer: true])  
    let cgimg = softwareContext.createCGImage(imageToSave, fromRect:imageToSave.extent())  
    let library = ALAssetsLibrary()  
    library.writeImageToSavedPhotosAlbum(cgimg,  
        metadata:imageToSave.properties(),  
        completionBlock:nil)  
}
```

**VIDEO**

# VIDEO FRA BUNDLE FIL

```
import MediaPlayer
```

```
class VideoViewController: UIViewController {
```

```
    var moviePlayerController:MPMoviePlayerController!
```

```
    override func viewDidLoad() {  
        super.viewDidLoad()
```

```
        if let fileURL = NSBundle.mainBundle().pathForResource("hunger", ofType: "mp4") {  
            self.moviePlayerController = MPMoviePlayerController(contentURL: NSURL.fileURLWithPath(fileURL))  
            self.moviePlayerController.prepareToPlay()  
            self.moviePlayerController.view.frame = self.view.frame  
            self.moviePlayerController.repeatMode = .One  
            self.moviePlayerController.controlStyle = .Embedded  
            self.view.addSubview(self.moviePlayerController.view)  
            self.moviePlayerController.play()  
        } else {  
            println("Failed")  
        }  
    }
```

```
}
```

```
}
```

# VIDEO FRA BILDESTRØM

```
import MediaPlayer
```

```
...
```

```
func imagePickerController(picker: UIImagePickerController!,
    didFinishPickingMediaWithInfo info: NSDictionary!) {
    self.dismissViewControllerAnimated(true, completion: nil);
    let videoURL = info[UIImagePickerControllerMediaURL] as NSURL

    let videoController = MPMoviePlayerController(contentURL: videoURL)
    videoController.prepareToPlay()
    videoController.view.frame = self.view.frame
    videoController.repeatMode = .One
    videoController.controlStyle = .Embedded
    self.view.addSubview(videoController.view)
    videoController.play()
}
```

# **MPMOVIEPLAYERVIEWCONTROLLER**

**INNEBYGD VIEW CONTROLLER SOM OGSÅ KAN BRUKES TIL Å SPILLE AV VIDEO**

# TA OPP VIDEO

```
import MobileCoreServices
```

```
...
```

```
@IBAction func captureVideo(sender: AnyObject) {  
    if UIImagePickerController.isSourceTypeAvailable(UINavigationControllerSourceType.Camera) {  
        let picker = UIImagePickerController()  
        picker.delegate = self  
        picker.allowsEditing = true  
        picker.sourceType = UINavigationControllerSourceType.Camera;  
        picker.mediaTypes = [kUTTypeMovie]  
        self.presentViewController(picker, animated: true, completion: nil)  
    }  
}
```

**LYD**

```
import AVFoundation
```



```
var player:AVAudioPlayer!

...

func startPlaying() {

    var error: NSError?
    if let fileURL = NSBundle.mainBundle().
        URLForResource("Canton_Snow-Break-320",
            withExtension: "mp3") {
        self.player = AVAudioPlayer(contentsOfURL: fileURL, error: &error)

        if let e = error {
            println(e.localizedDescription)
            return
        }

        player.delegate = self
        player.volume = 1.0
        player.prepareToPlay()
        player.play()
    } else {
        println("could not load file")
    }
}
```

```
var player:AVAudioPlayer!
```

```
...
```

```
func stopPlaying() {  
    if player.playing {  
        player.stop()  
    }  
}
```

# EN ANNEN MÅTE Å STØTTE DELEGATES

```
extension Sound : AVAudioPlayerDelegate {  
    func audioPlayerDidFinishPlaying(player: AVAudioPlayer!, successfully flag: Bool) { }  
    func audioPlayerDecodeErrorDidOccur(player: AVAudioPlayer!, error: NSError!) {}  
}
```

# AVAUDIOSESSION

```
func setSessionPlayer() {  
    let session = AVAudioSession.sharedInstance()  
    var error: NSError?  
  
    if !session.setCategory(AVAudioSessionCategorySoloAmbient, error:&error) {  
        if let e = error {  
            println(e.localizedDescription)  
        }  
    }  
  
    if !session.setActive(true, error: &error) {  
        if let e = error {  
            println(e.localizedDescription)  
        }  
    }  
}
```

# **OPPGAVER**

## **SE ØVINGSOPPGAVENE**

**[HTTPS://GITHUB.COM/HINDERBERG/IOS-SWIFT-KURS](https://github.com/hinderberg/ios-swift-kurs)**