//: [Previous](@previous)

import UIKit

import PlaygroundSupport

// Calls the views from View Class

let color = ViewBuilder().color

let baseView = ViewBuilder().baseView

let topViewScore = ViewBuilder().topViewScore

let topViewText = ViewBuilder().topViewText

// Create view for Animal Picture

let tipView = ViewBuilder().tipView

let beeBug = ViewBuilder().beeBug

let wordArray = ["ABC","Cat","Bee"]

var wordLetters = wordArray[0].characters.count //Gets number of letters depending on Word selected

var wordLetter = 0

var letterArray= ViewBuilder().letterArray:[UIImageView] = []

var time=1

var letterCreatedArray:[Character] = []

// Cria as views das imagens das LETRAS

func createImageView(x: Int, y: Int, width: Int, height: Int, image: String) -> UIImageView{

let imgName = UIImage(named: image)

let viewName = UIImageView (frame: CGRect(x: x, y: y, width: width, height: height))

viewName.image=imgName

baseView.addSubview(viewName)

UIView.animate(withDuration: 0.1, animations: {

viewName.center = CGPoint(x: x+45, y: y+45)

})

letterArray.append(viewName)

return viewName

}

//Create Player Images

let beeView = ViewBuilder().beeView

let downArrowView = ViewBuilder().downArrowView

let upArrowView = ViewBuilder().upArrowView

PAREI AQUI!!!

//Funcao remove as Flechas de dica

//func removeArrows() -> Void{

// moveView(viewName: downArrowView, x:-55, y: 435)

// moveView(viewName: upArrowView, x:-55, y: 255)

//}

//// Funcao base de mover as imagens

//func moveView(viewName: UIImageView, x: Int, y: Int) -> Void{

// UIView.animate(withDuration: 2.0, animations: { viewName.center = CGPoint(x: x, y: y)})

//}

//// Funcao base de mover as imagens

func moveView(viewName: UIImageView, x: Int, y: Int) -> Void{

// UIView.animate(withDuration: 2.0, options: [], animations: { viewName.center = CGPoint(x: x, y: y)}, completion:nil)

UIView.animate(withDuration: 5.0, delay: 0, usingSpringWithDamping: 0, initialSpringVelocity: 0, options: [], animations: { viewName.center = CGPoint(x: x, y: y)}, completion: { (finished: Bool) -> Void in

///// Tentar com callback inves de loop

let numberRepete = letterArray.count

for eachImageView in 1..<numberRepete{

moveView(viewName: letterArray[eachImageView], x: 100, y: 500)

}

})

}

// Funcao move as Letras que batem na abelha!

func specialMoveView(viewName: UIImageView) -> Void{

UIView.animate(withDuration: 5.0, animations: {

var oldX = viewName.center.x

var oldY = viewName.center.y

let newImage = UIImageView (frame: CGRect(x: oldX, y: oldY, width: 80, height: 80))

newImage.image = viewName.image

baseView.addSubview(newImage)

UIView.animate(withDuration: 0.1, animations: {

newImage.center = CGPoint(x: oldX+45, y: oldY+45)

})

newImage.transform = newImage.transform.scaledBy(x: 0.5, y: 0.5)

let rotateAnimation = CABasicAnimation(keyPath: "transform.rotation")

rotateAnimation.fromValue = 0.0

rotateAnimation.toValue = CGFloat(M\_PI \* 2.0)

rotateAnimation.duration = 5.0

newImage.layer.add(rotateAnimation, forKey: nil)

let newX = CGFloat(300+(wordLetter\*(70)))

let newY = CGFloat(50)

newImage.center = CGPoint(x: newX, y: newY)

viewName.center = CGPoint(x: -50, y: oldY)

wordLetter += 1

}) { (result) in viewName.layer.removeAllAnimations()}

}

//Funcao move as Letras Horizontalmente

func moveLetters(time: Int) -> Void{

for eachLetter in 0..<letterArray.count {

let letterY = Int(letterArray[eachLetter].frame.origin.y+40.0)

let letterX = Int(letterArray[eachLetter].frame.origin.x+40.0)-(time\*105)

moveView(viewName: letterArray[eachLetter], x: letterX, y: letterY)

}

}

//Funcao escolhe uma Letra Aleatoria para ser criada

func randomLetter() -> String{

let random = arc4random\_uniform(3) + 1;

var newletter = ""

switch random {

case 1:

newletter = "letterA.png"

case 2:

newletter = "letterB.png"

case 3:

newletter = "letterC.png"

default:

print("Error")

}

return newletter

}

// Funcao cria uma letra a direita da tela em posicao aletoria

func spawnLetter() -> Void{

let random = arc4random\_uniform(5) + 1;

let newLetter = randomLetter()

let index = newLetter.characters.index(newLetter.startIndex, offsetBy: 6)

let letter = newLetter[index]

letterCreatedArray.append(letter)

switch random {

case 1:

\_ = createImageView(x: 525, y: 110, width: 80, height: 80, image: newLetter)

case 2:

\_ = createImageView(x: 525, y: 215, width: 80, height: 80, image: newLetter)

case 3:

\_ = createImageView(x: 525, y: 320, width: 80, height: 80, image: newLetter)

case 4:

\_ = createImageView(x: 525, y: 425, width: 80, height: 80, image: newLetter)

case 5:

\_ = createImageView(x: 525, y: 530, width: 80, height: 80, image: newLetter)

default:

print("Error")

}

}

var commandNumber = 0

// Funcao move Abelha pra Cima/Baixo

func moveBee(comand: String) -> Void{

spawnLetter() // Chama funcao de Criar letras a Direta

moveLetters(time: time) // Chama funcao que mexe letras quando jogador se mexe

let arrowX = Int(downArrowView.frame.origin.x+25)

let letterX = Int(beeView.frame.origin.x+40.0)

commandNumber+=1

if comand == "up"{

// time+=1

if (beeView.frame.origin.y > 110){

let letterY = Int(beeView.frame.origin.y+40.0)-105

let arrowUpY = Int(upArrowView.frame.origin.y+40.0)-115

let arrowDownY = Int(downArrowView.frame.origin.y+40.0)-115

moveView(viewName: beeView, x: letterX, y: letterY)

if(upArrowView.frame.origin.y < 110){

moveView(viewName: upArrowView, x: arrowX, y: arrowUpY)

} else {

moveView(viewName: upArrowView, x: arrowX-100, y: arrowUpY)

}

moveView(viewName: downArrowView, x: arrowX, y: arrowDownY)

}

// removeArrows()

} else if comand == "down" {

// time+=1

if (beeView.frame.origin.y < 530){

let letterY = Int(beeView.frame.origin.y+40.0)+105

let arrowUpY = Int(upArrowView.frame.origin.y+40.0)+115

let arrowDownY = Int(downArrowView.frame.origin.y+40.0)+115

moveView(viewName: beeView, x: letterX, y: letterY)

moveView(viewName: downArrowView, x: arrowX, y: arrowDownY)

moveView(viewName: upArrowView, x: arrowX, y: arrowUpY)

if(upArrowView.frame.origin.y > 530){

moveView(viewName: downArrowView, x: arrowX, y: arrowDownY)

} else {

moveView(viewName: downArrowView, x: arrowX-100, y: arrowDownY)

}

}

// removeArrows()

}

checkBeeLetterContact()

}

//Funcao coonfere se Abelha e Letra se encontram

func checkBeeLetterContact() -> Void{

let numberOfLetterImages = letterArray.count

for eachLetter in 0 ..< numberOfLetterImages {

let letterCoordinates = (Int(letterArray[eachLetter].frame.origin.x / 100) , Int(letterArray[eachLetter].frame.origin.y / 100))

let beeCoordinates = (Int(beeView.frame.origin.x / 100) , Int(beeView.frame.origin.y / 100))

var selectedWordArray = Array(wordArray[0].characters)

// print("\(eachLetter) \(letterCoordinates) \(beeCoordinates)")

if ((letterCoordinates.0 == beeCoordinates.0) && (letterCoordinates.1 == beeCoordinates.1)){

// if(letterCreatedArray[eachLetter]==selectedWordArray[eachLetter]){

print(letterCreatedArray[eachLetter])

print(selectedWordArray[eachLetter])

print("ENTROUUUU")

specialMoveView(viewName: letterArray[eachLetter])

letterCreatedArray.remove(at: eachLetter)

// letterArray.remove(at: eachLetter)

// selectedWordArray.remove(at: eachLetter)

// if(letterCreatedArray.count == selectedWordArray.count){

// //Animacao de winner!

// }

// }

}

if (wordLetter == wordLetters){

break

}

}

}

//func gameLevel( lvl: Int) {

// switch lvl {

// case 1:

// //Sequencia de letras que surgem!

// default:

// print("Error")

// }

//}

//moveBee(comand: "down")

// gameLevel(1)

//moveBee(comand: "up")

//checkBeeLetterContact()

//moveBee(comand: "up")

//checkBeeLetterContact()

//moveBee(comand: "up")

//checkBeeLetterContact()

//moveBee(comand: "up")

//checkBeeLetterContact()

//moveBee(comand: "up")

////checkBeeLetterContact()

//moveBee(comand: "up")

////checkBeeLetterContact()

//moveBee(comand: "up")

//checkBeeLetterContact()

//moveBee(comand: "up")

//checkBeeLetterContact()

//moveBee(comand: "up")

//checkBeeLetterContact()

moveBee(comand: "down")

moveBee(comand: "down")

moveBee(comand: "down")

moveBee(comand: "down")

//checkBeeLetterContact()

//class myself {

//

//

// func buttonAction(sender:UIButton!)

// {

// print("Button tapped")

// }

//

//}

//

//var s = myself()

//class Responder : NSObject {

//

//}

//

//// Button tem que estar em uma classe para o selector funcionar

//var arrowButton = UIButton(frame: CGRect(x: 300, y: 300, width: 100, height: 50))

//let downArrowImg = UIImage(named: "downArrow.png")

//arrowButton.setImage( downArrowImg , for: .normal)

//arrowButton.setTitle("TestButton", for: .normal)

//baseView.addSubview(arrowButton)

////arrowButton.addTarget(Responder, action: #selector(), for: .touchUpInside)

//

//if( commandNumber == 0){

//// arrowButton

//}