



#digitalReskilling Software Developer

Desenvolvimento de Aplicações Mobile - iOS

Bruno Santos
bruno.santos@cesae.pt

Las Vegas Dicee

- Com o Projeto “Las Vegas Dicee” vamos criar uma APP de lançamento de dados

Las Vegas Dicee

Choose options for your new project:

Product Name:

Team:

Organization Identifier:

Bundle Identifier:

Interface:

Language:

☐ Use Core Data

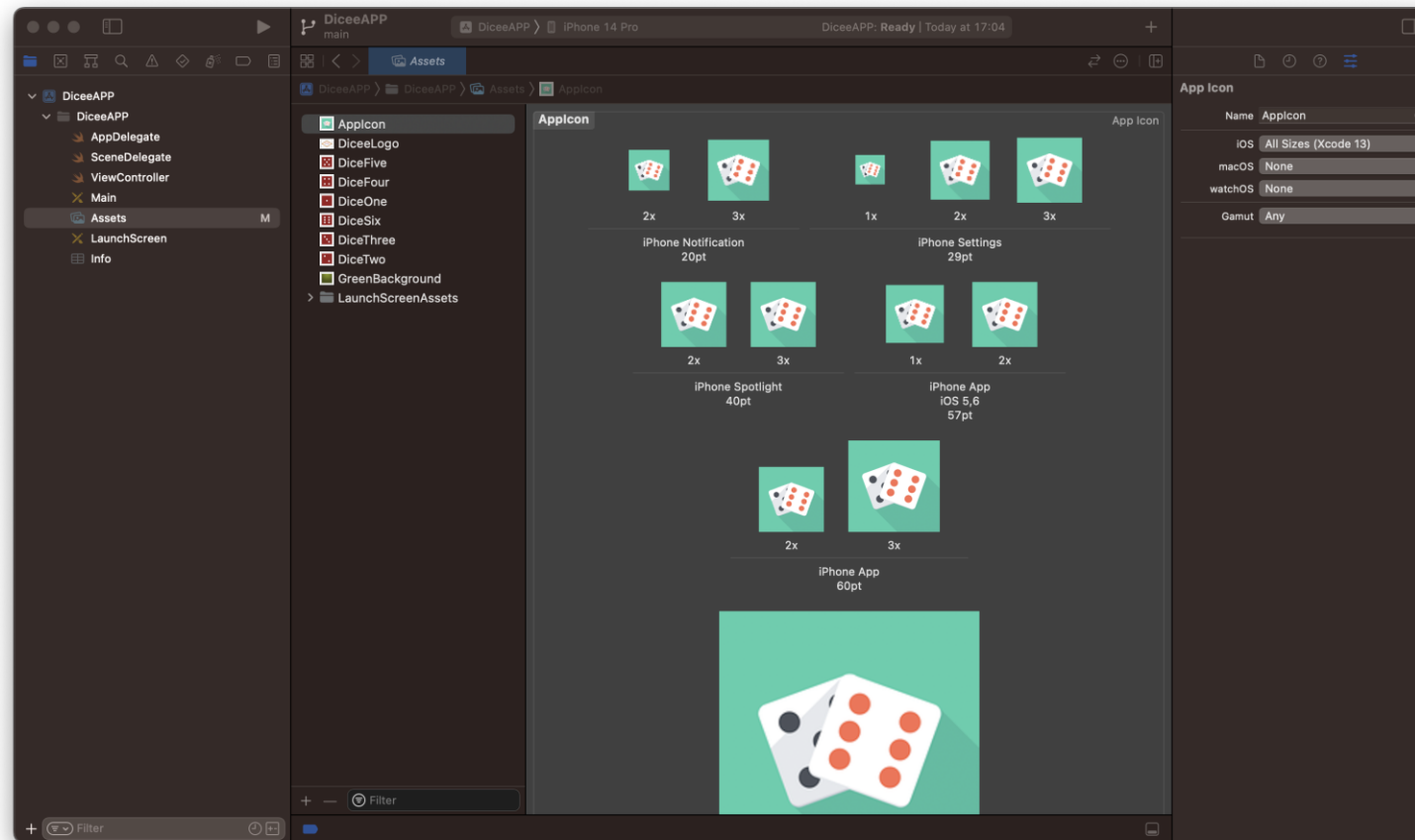
☐ Host in CloudKit

☐ Include Tests

Las Vegas Dicee

- Para começar vamos adicionar os Assets, previamente selecionados, ao projeto

Las Vegas Dicee

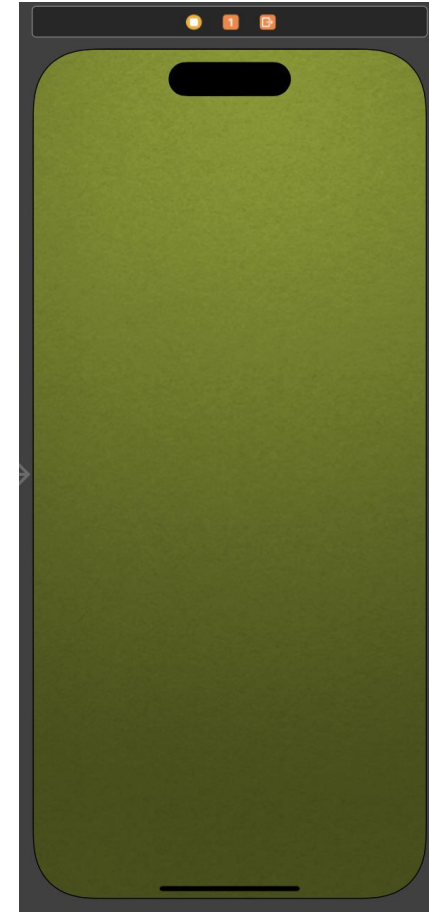


Las Vegas Dicee

- Começamos por construir a interface com um background, imagem da APP, imagens dos dados e botão de jogo

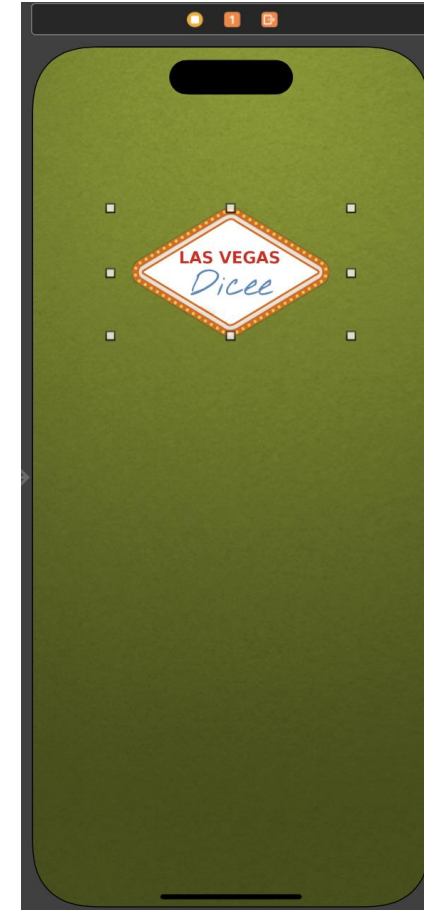
Las Vegas Dicee

- Para colocar o fundo inserimos uma Image View que irá cobrir todo o ecrã, alteramos a Image para GreenBackground e o Content Mode para Aspect Fill



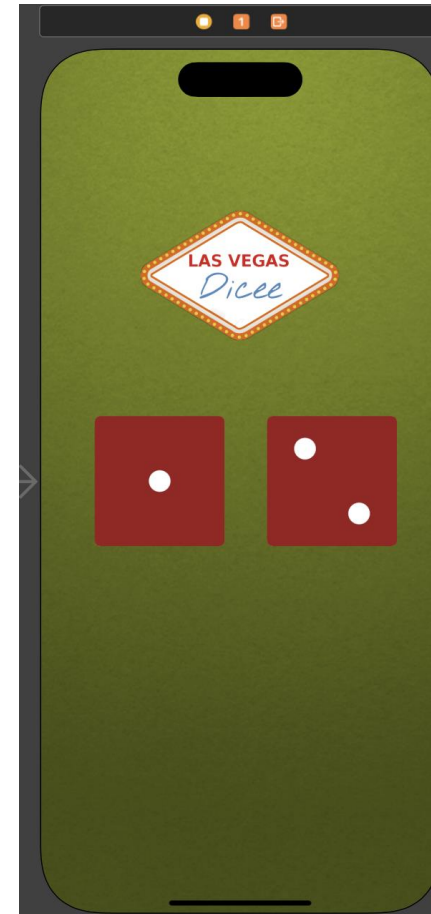
Las Vegas Dicee

- Adicionamos agora uma nova Image View para colocar o logo da APP
- Esta imagem está centrada horizontalmente



Las Vegas Dicee

- Adicionamos ainda duas Image View, lado a lado, para apresentação dos dois dados, colocando os dados 1 e 2 como Image selecionadas



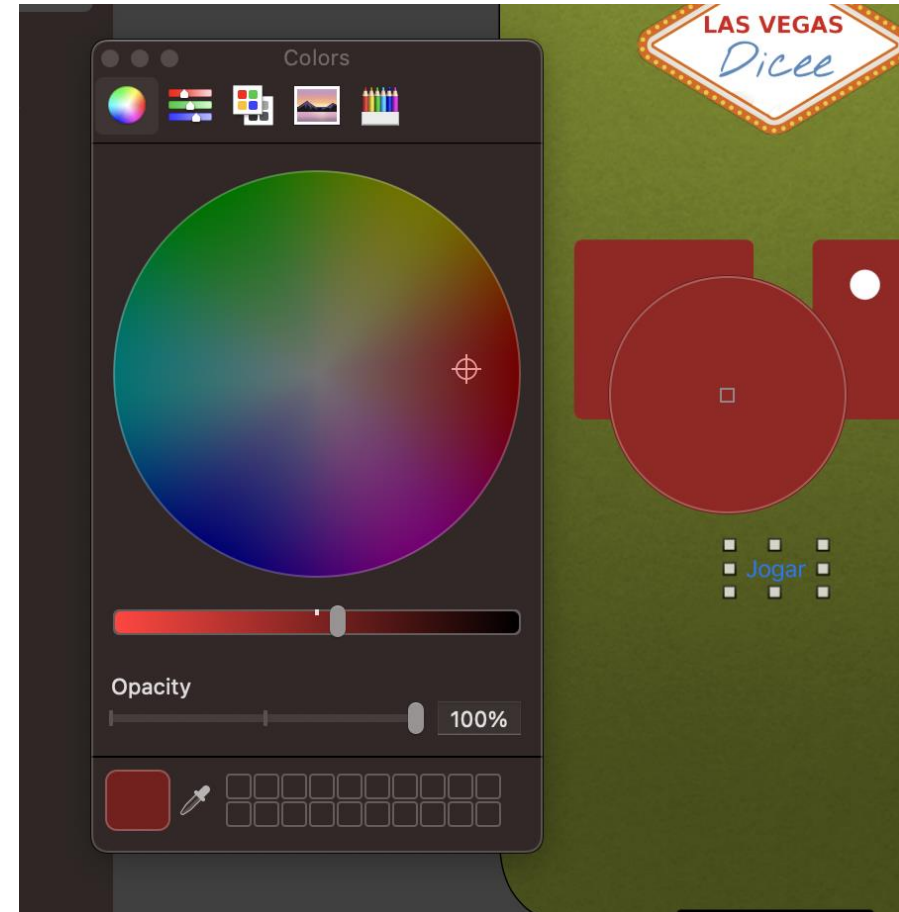
Las Vegas Dicee

- Finalmente é colocado um botão com o texto jogar, abaixo dos dados.
- A personalização do botão será feita de seguida.



Las Vegas Dicee

- Para que o botão tenha a mesma cor de fundo dos dados vamos selecionar o mesmo, no parâmetro Background selecionar Custom e em Fill também Custom.
- Na nova janela utilizamos o Color Picker para selecionar a cor vermelha do dado



Las Vegas Dicee

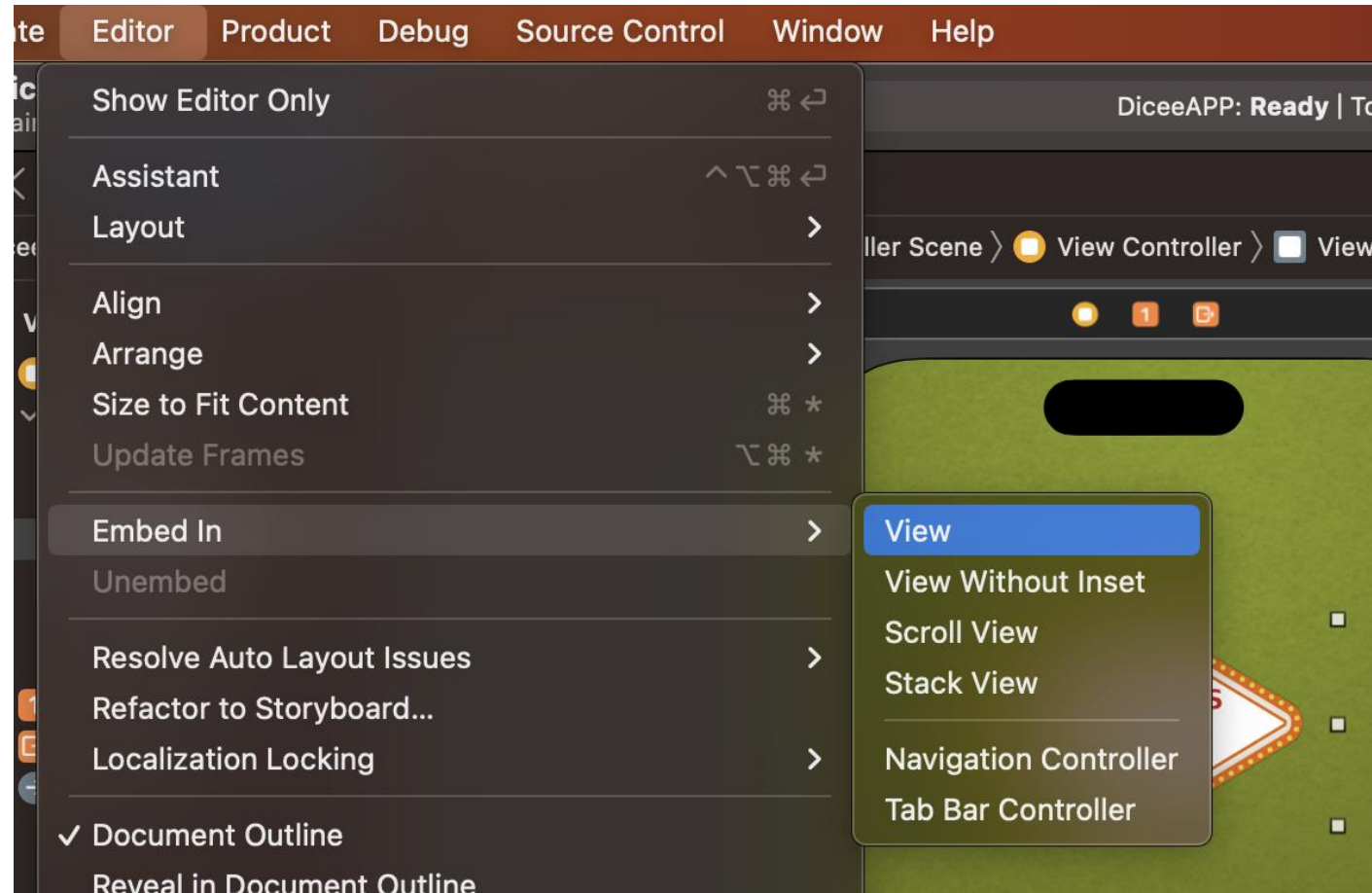
- Alteramos o Foreground para Secondary System Background Color e a Font para System com tamanho 40.



Las Vegas Dicee

- Para que a APP se consiga reajustar minimamente independentemente no ecrã em que seja apresentada vamos trabalhar as Constraints, para isso vamos colocar os nossos elementos dentro de 3 Views que irão organizar os conteúdos.
- Selecionando o logo vamos ao menu Editor, selecionamos Embed In e de seguida, View

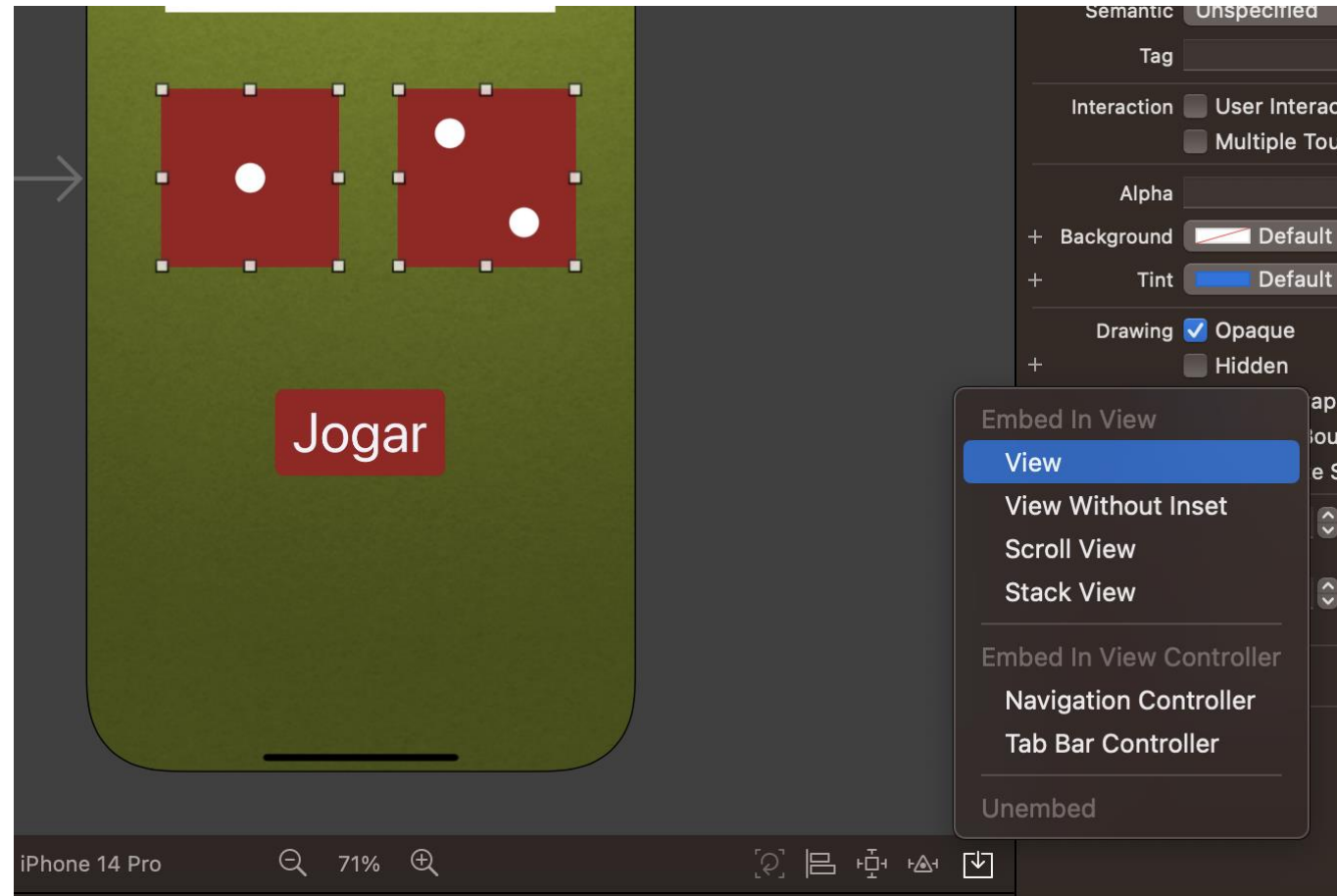
Las Vegas Dicee



Las Vegas Dicee

- Uma forma alternativa é utilizar o atalho Embed In no fundo da página, algo que vamos fazer após selecionar os dois dados que vamos querer colocar numa nova View

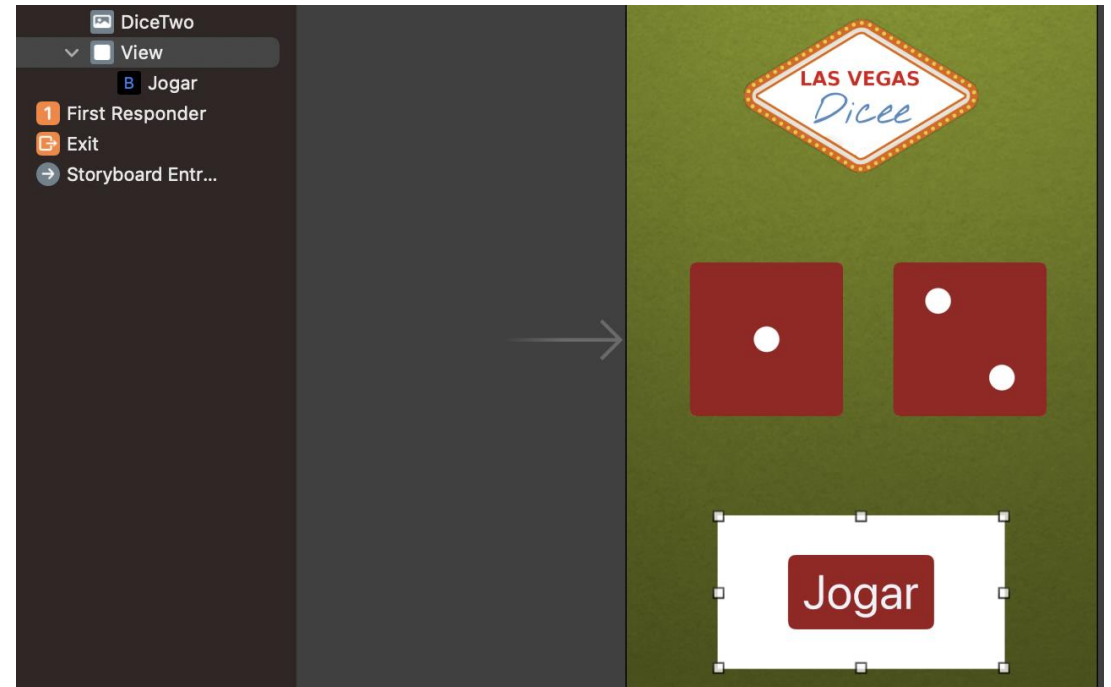
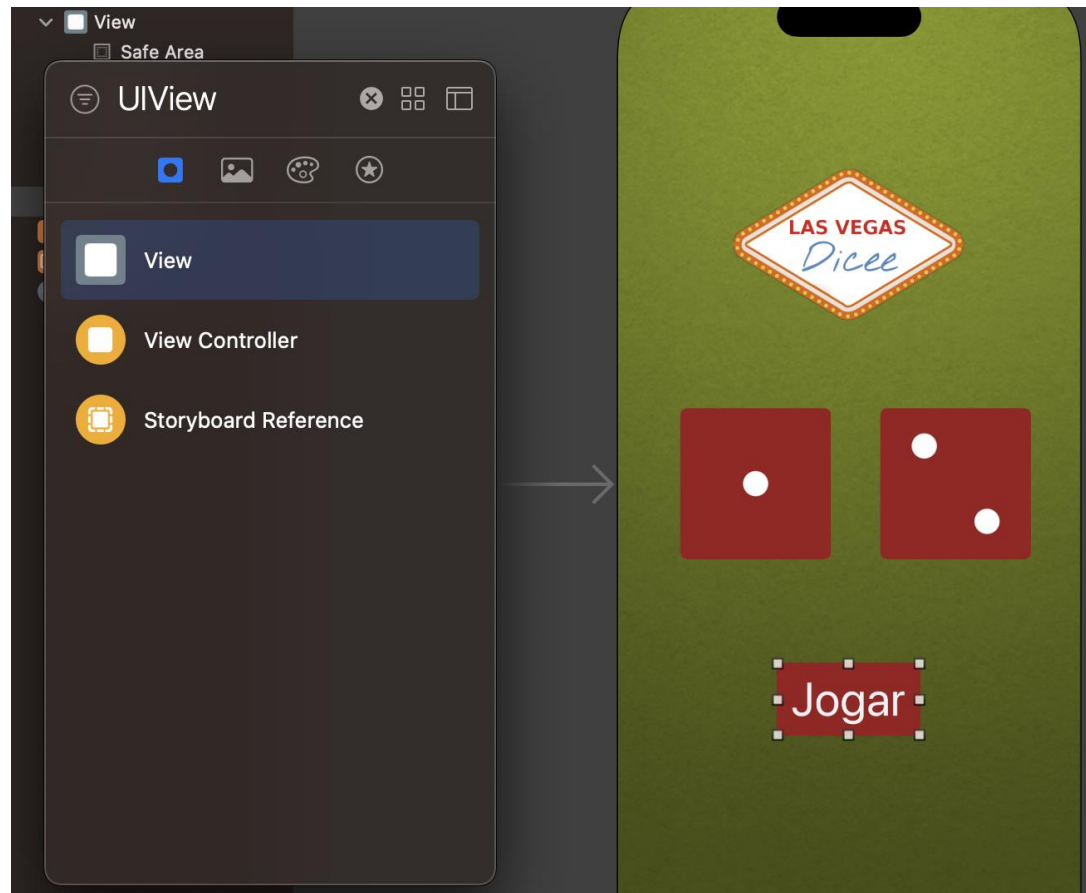
Las Vegas Dicee



Las Vegas Dicee

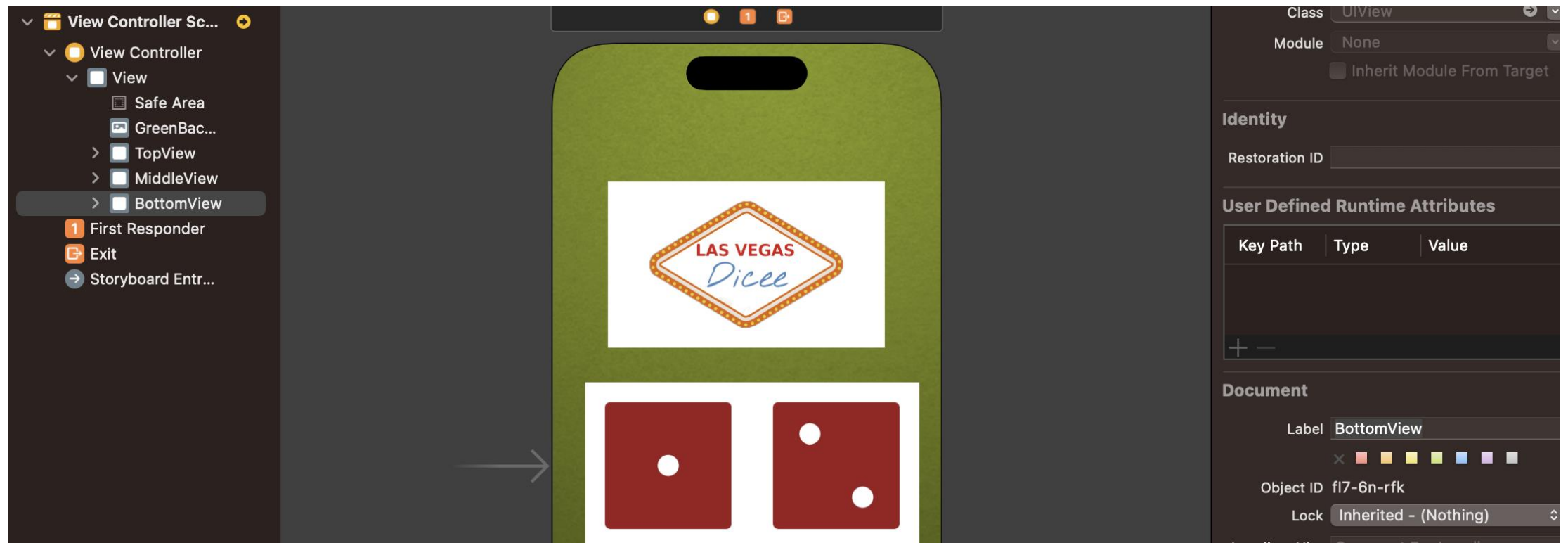
- Temos ainda uma terceira forma que será adicionar uma View da nossa Component Library e colocar o elemento dentro, algo que vamos fazer com o botão Jogar

Las Vegas Dicee



Las Vegas Dicee

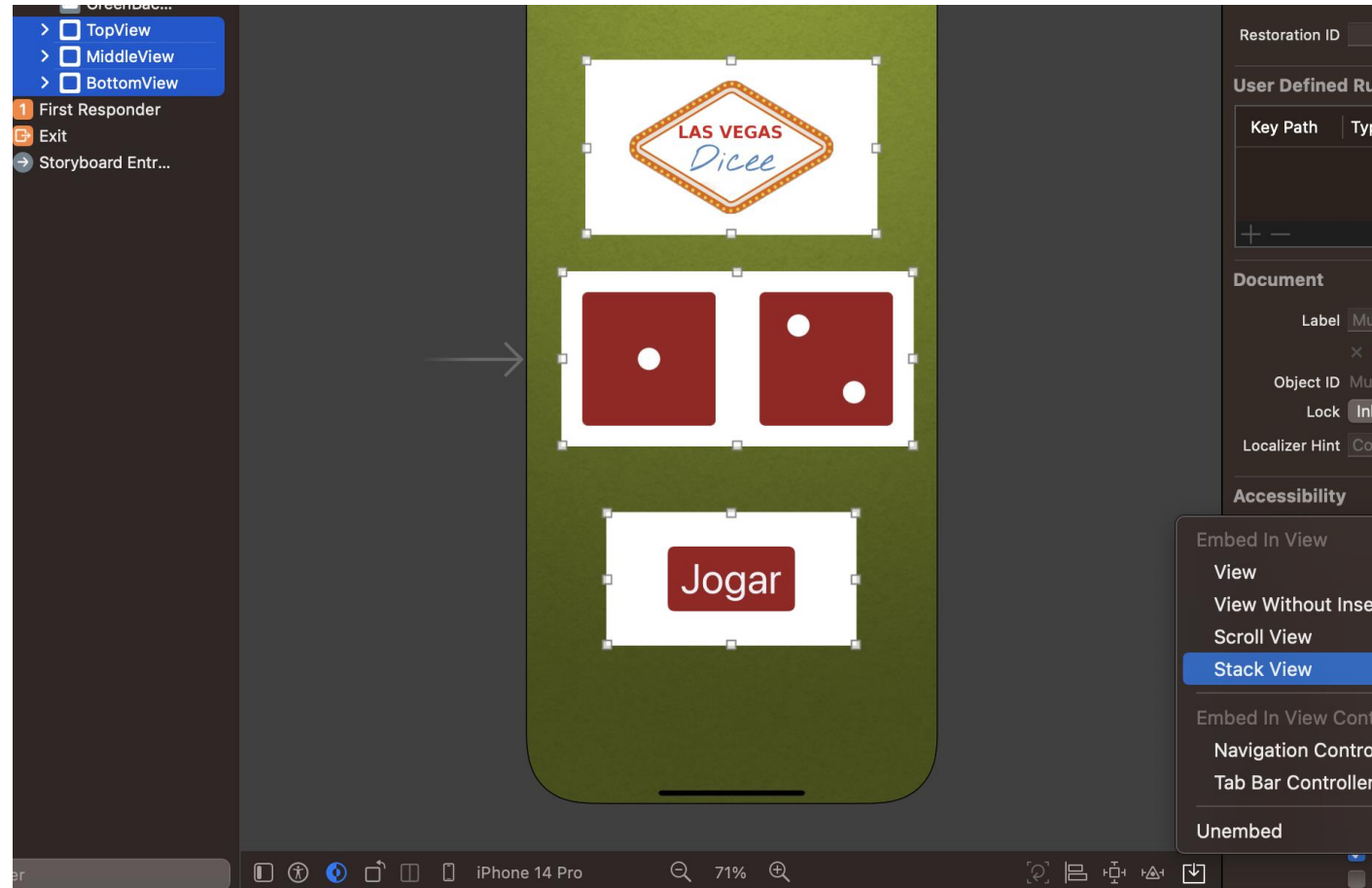
- Para melhor organização vamos dar nomes às View e ordená-las



Las Vegas Dicee

- Para garantir que temos todos os elementos organizados vamos embeber todas as Views anteriormente criadas numa StackView

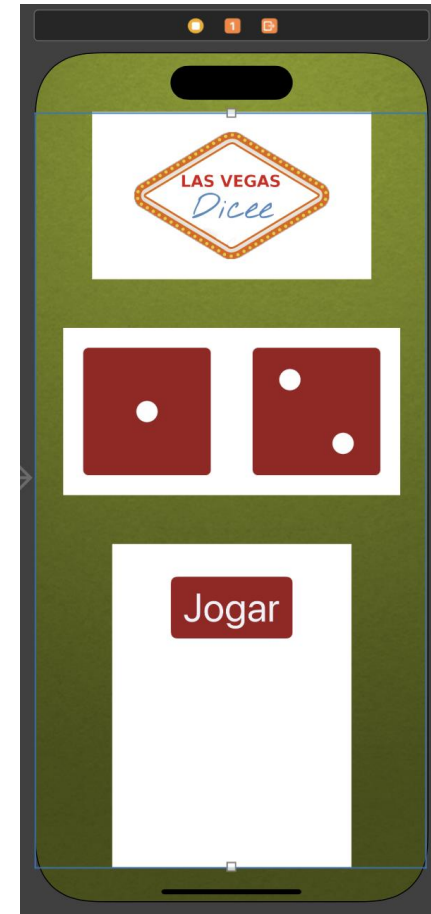
Las Vegas Dicee



Las Vegas Dicee

- Para garantir que a StackView ocupe toda a Safe Area vamos adicionar um conjunto de Constraints.
- Importante garantir que estamos a criar Constraints para a Safe Area e não para a View

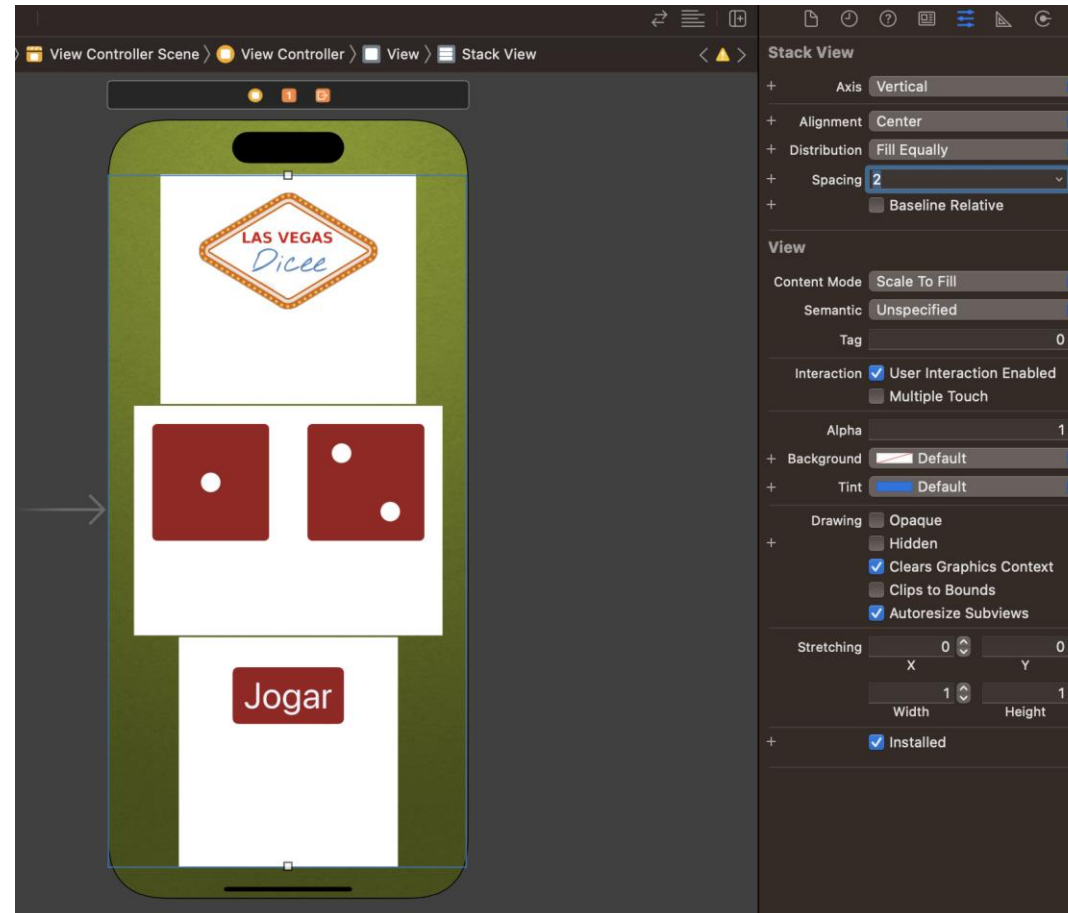
Las Vegas Dicee



Las Vegas Dicee

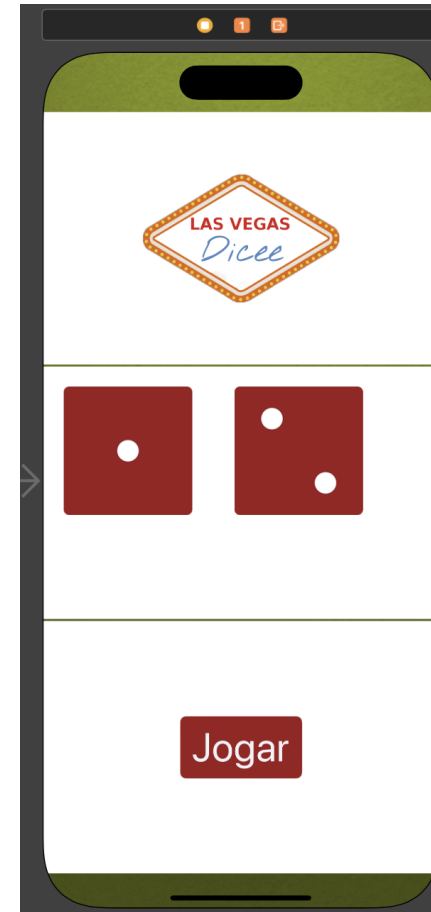
- Na StackView vamos ainda definir que queremos que cada elemento ocupe uma parte igual aos outros e definir um pequeno espaçamento entre eles, assim:
 - Distribution: Fill Equally
 - Spacing: 2

Las Vegas Dicee



Exercício 1

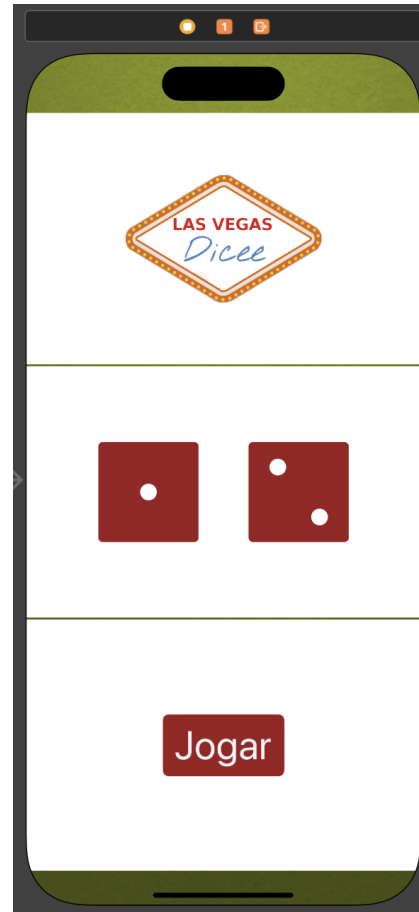
- Coloque as três Views com distância 0 horizontal de forma a ocupar todo o espaço e centre o logo e o botão de jogar



Las Vegas Dicee

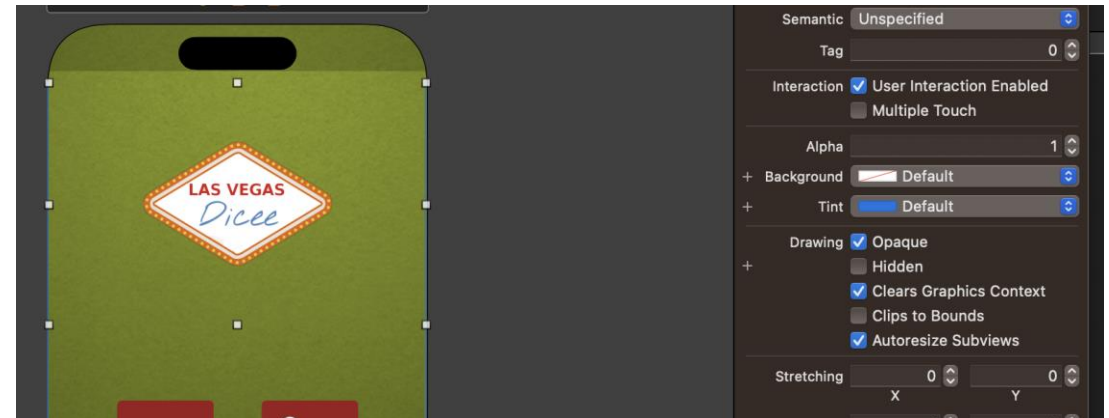
- Para colocar os dados a ocupar metade da View do meio vamos coloca-los numa nova StackView, desta vez horizontal.
- Esta StackView terá toda a dimensão da View em que se encontra e daremos Fill Equally para os seus elementos. Como spacing vamos atribuir 50.
- Vamos também centrar a StackView (vertical e horizontal)

Las Vegas Dicee



Las Vegas Dicee

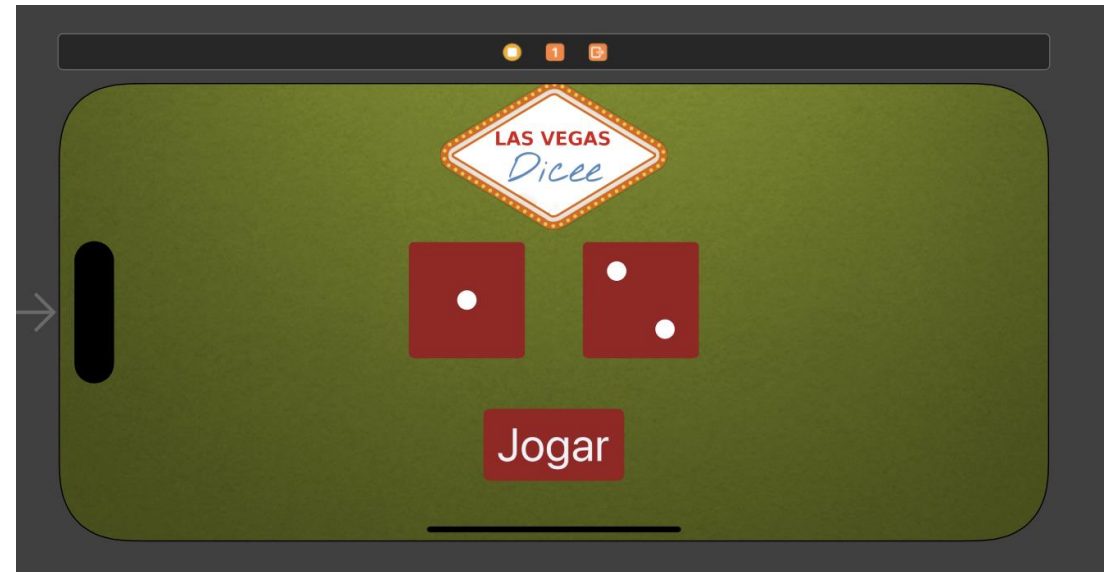
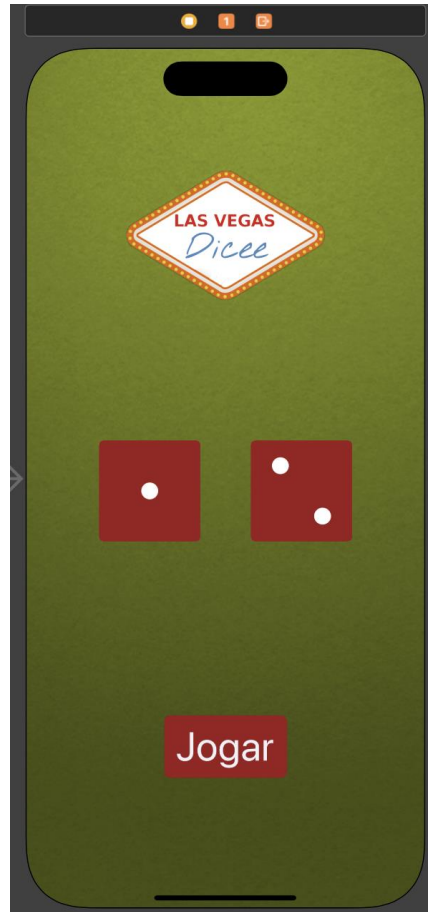
- Finalmente vamos retirar o background das Views de forma a conseguirmos ver o fundo da nossa APP



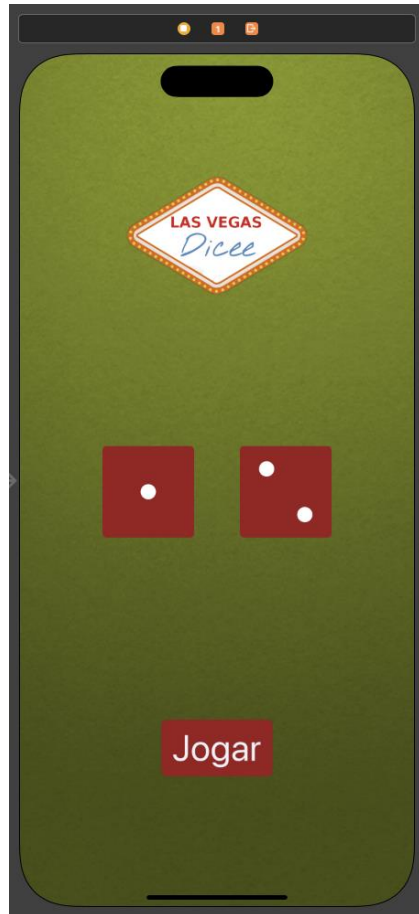
Las Vegas Dicee

- Faltou ainda ajustar a Image View de fundo, vamos adicionar as constraint de 0 para a View (não Safe Area)
- Finalmente testar a APP vertical e horizontal em mais do que um dispositivo

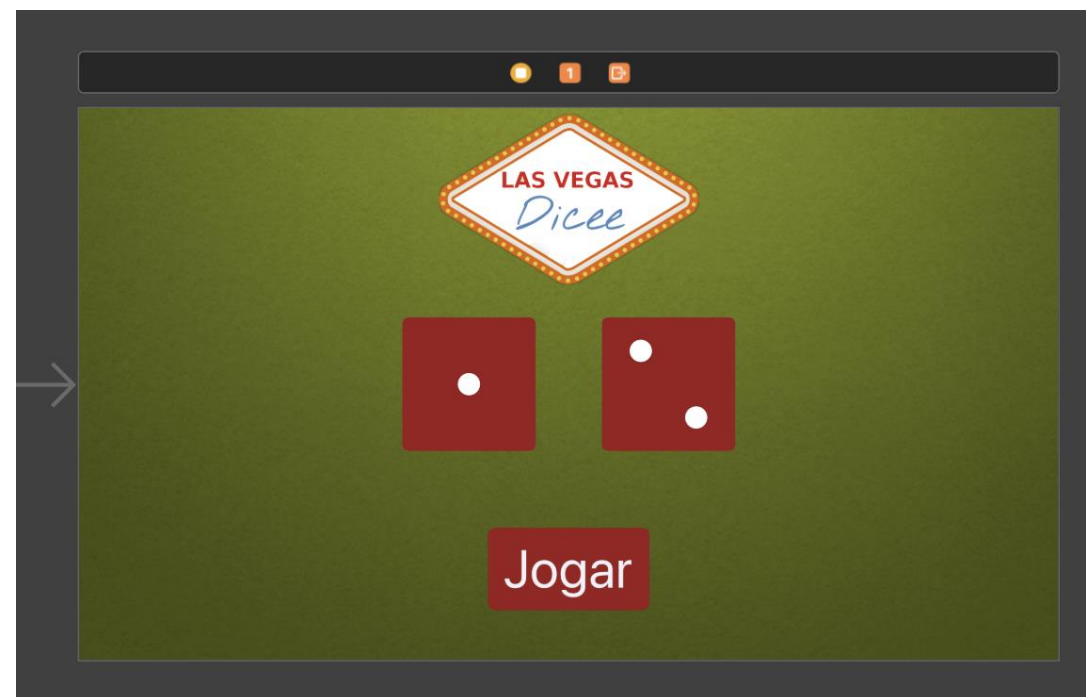
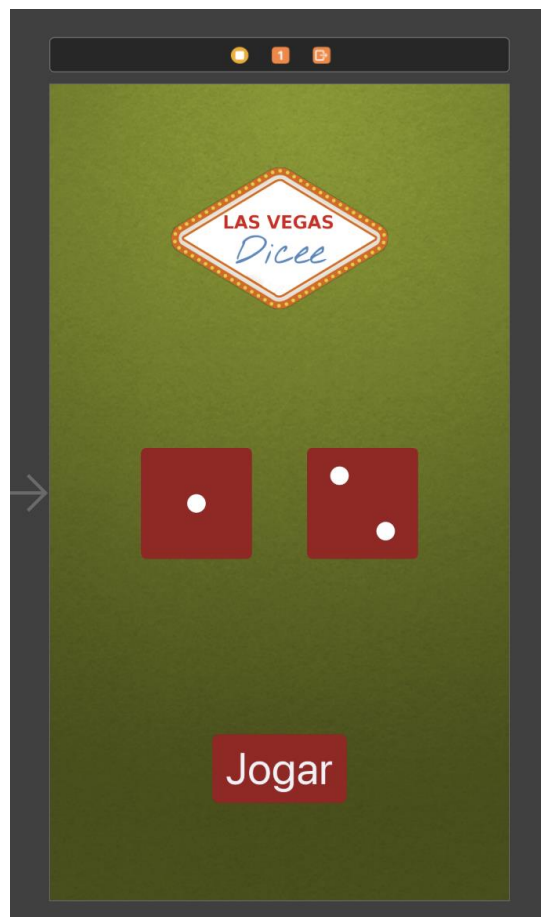
Las Vegas Dicee – iPhone 14 Pro



Las Vegas Dicee – iPhone 14 Pro Max

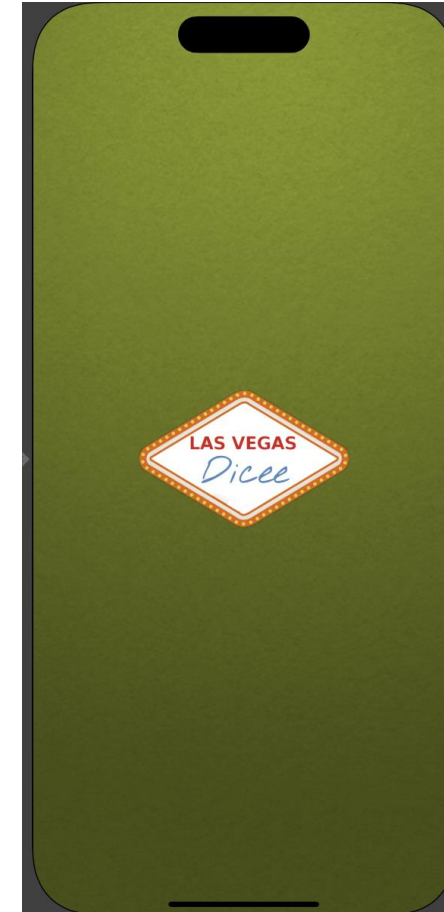


Las Vegas Dicee – iPhone 8 Plus



Exercício 2

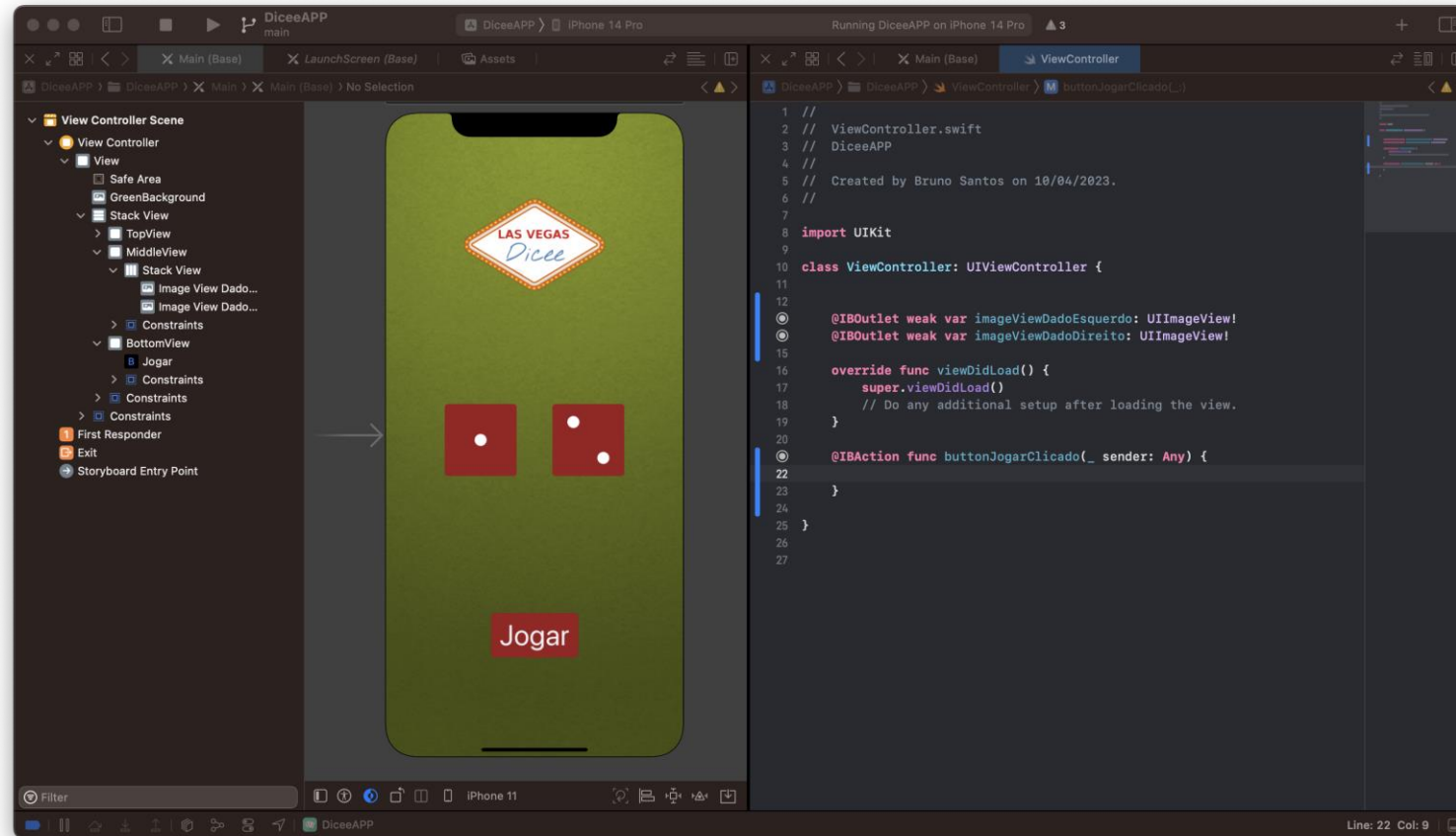
- Crie um LaunchScreen semelhante ao apresentado, tendo em consideração a aplicação de Constraints:
 - O logo deve estar centrado;
 - O background deve ocupar toda a View.



Las Vegas Dicee

- Necessitamos de interatividade na nossa APP, para isso vamos mapear os elementos de layout no ViewController

Las Vegas Dicee



Las Vegas Dicee

- Adicione um array com as imagens

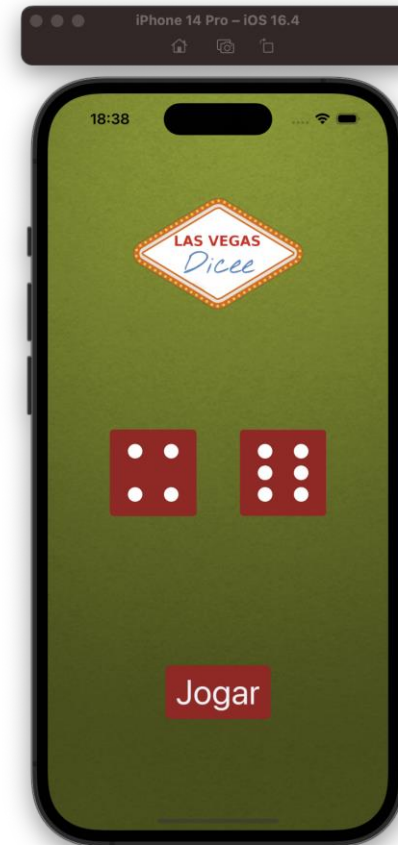
```
@IBOutlet weak var imageViewDadoEsquerdo: UIImageView!  
@IBOutlet weak var imageViewDadoDireito: UIImageView!  
let imagens = [UIImage(named: "DiceOne"),UIImage(named: "DiceTwo"),UIImage(named: "DiceThree"),  
               UIImage(named: "DiceFour"),UIImage(named: "DiceFive"),UIImage(named: "DiceSix")]
```

Las Vegas Dicee

- Programe o clique no botão para que seja gerado um valor aleatório e que a imagem de cada dado seja gerada

```
@IBAction func buttonJogarClicado(_ sender: Any) {  
    imageViewDadoEsquerdo.image = imagens[Int.random(in: 0..    imageViewDadoDireito.image = imagens[Int.random(in: 0..}
```

Las Vegas Dicee



Exercício 3

- Altere a aplicação anterior de forma a que quando os dois dados saírem iguais seja apresentada uma mensagem ao utilizador com a informação DUPLO: X (número que saiu)

Exercício 3

