

Documentação do Projeto - SimpleJourneyApp

Descrição Geral

O **SimpleJourneyApp** é um aplicativo Android criado com **Kotlin** e **Jetpack Compose**. Ele simula uma jornada na qual o usuário clica em uma imagem até atingir um número aleatório de cliques para "vencer". As imagens mudam conforme o progresso, e o jogador pode optar por desistir a qualquer momento.

Funcionalidades Principais

1. **Jogo de Cliques**: O jogador deve clicar em uma imagem até atingir um número aleatório de cliques para vencer o jogo.
2. **Imagens de Progressão**: Conforme o jogador avança, as imagens mudam para refletir o progresso.
3. **Desistência**: O jogador pode optar por desistir do jogo, reiniciá-lo ou encerrá-lo.
4. **Diálogo de Jogo Finalizado**: Após a vitória ou desistência, um diálogo permite ao usuário reiniciar o jogo ou encerrar o aplicativo.

Instruções de Execução

1. Clone ou baixe este repositório em seu ambiente local: `git clone <link-do-repositório>`
2. Abra o Android Studio.
3. Clique em `File > Open` e navegue até o diretório do projeto clonado.
4. Após a abertura do projeto, certifique-se de que as dependências estão atualizadas. Clique em `File > Sync Project with Gradle Files` para garantir que tudo está corretamente configurado.
5. Conecte um dispositivo Android físico ou inicie um emulador.
6. Execute o projeto clicando no botão "Run" (ou pressione `Shift + F10`).

.....

Componentes e Funções do Código

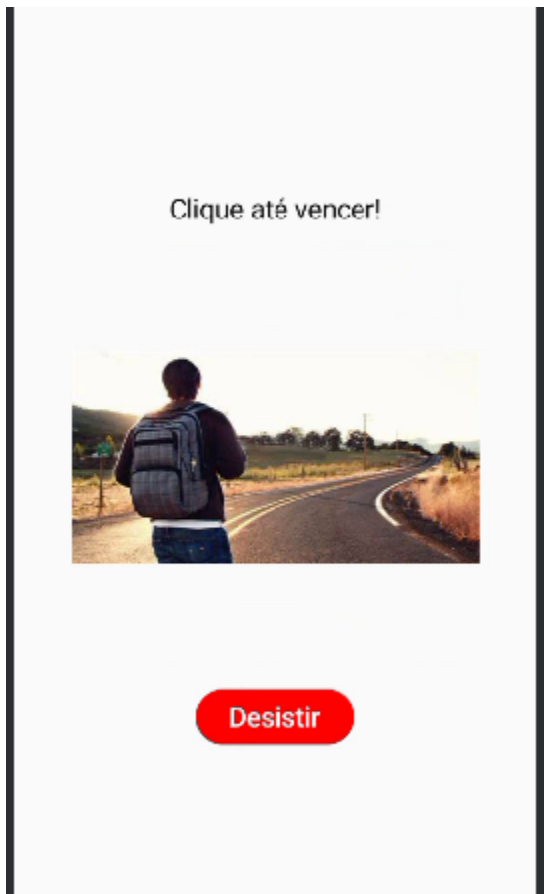
1. **MainActivity**

A `MainActivity` é a tela principal do aplicativo e contém o método `onCreate()` que inicializa o conteúdo do app.

```

```kotlin
class MainActivity : ComponentActivity() {
 override fun onCreate(savedInstanceState: Bundle?) {
 super.onCreate(savedInstanceState)
 setContent {
 SimpleJourneyApp()
 }
 }
}

```



...

### ### 2. **SimpleJourneyApp**

A função principal `SimpleJourneyApp` contém toda a lógica do jogo, incluindo a contagem de cliques, mudança de imagens e diálogos para desistência ou finalização.

```

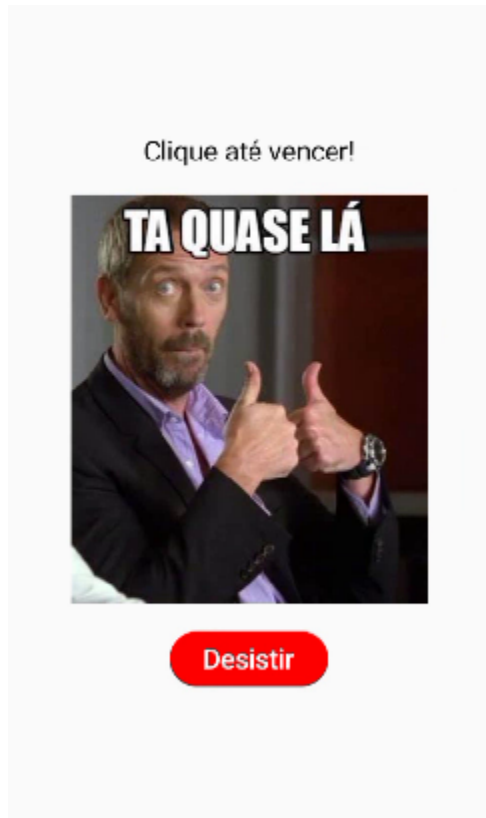
```kotlin
@Composable
fun SimpleJourneyApp() {

```

```

var click by rememberSaveable { mutableStateOf(0) }
val randomClicksRequired = rememberSaveable { Random.nextInt(1, 51) }
var currentImage by rememberSaveable { mutableStateOf(R.drawable.inicial) }
var fimJogo by rememberSaveable { mutableStateOf(false) }
var vitoria by rememberSaveable { mutableStateOf(false) }
var showDesistirDialog by rememberSaveable { mutableStateOf(false) }
var showFinalDialog by rememberSaveable { mutableStateOf(false) }
val activity = (LocalContext.current as? Activity)
...

```



3. **Imagens de Progresso**

As imagens do jogo são alteradas conforme o número de cliques aumenta. Essas imagens são definidas de acordo com o estágio do progresso do jogador.

```

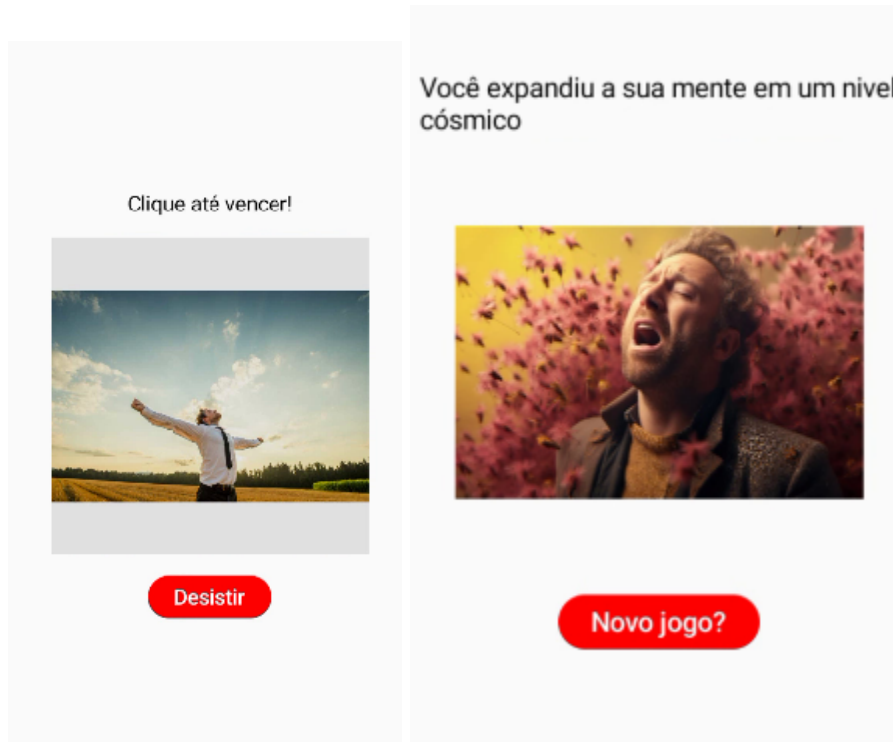
```kotlin
when {
 click >= randomClicksRequired -> {
 currentImage = R.drawable.fim
 fimJogo = true
 vitoria = true
 showFinalDialog = true
 }
}

```

```

click >= (randomClicksRequired * 2 / 3) -> currentImage = R.drawable.vitoria
click >= (randomClicksRequired / 3) -> currentImage = R.drawable.mediana
else -> currentImage = R.drawable.inicial
}
...

```



Primeira imagem: fim

Segunda imagem: vitoria

#### ### 4. \*\*Diálogo de Desistência\*\*

Quando o jogador clica no botão "Desistir", um diálogo é exibido com as opções "Sim" e "Não". Caso o jogador escolha "Sim", o jogo é reiniciado. Se escolher "Não", a Imagem de Desistência é exibida e o jogo termina.

```

```kotlin
if (showDesistirDialog) {
    AlertDialog(
        onDismissRequest = { showDesistirDialog = false },
        title = { Text("Desistir do jogo") },
        text = { Text("Você quer iniciar um novo jogo?") },
        confirmButton = {
            TextButton(onClick = {
                click = 0
                currentImage = R.drawable.inicial
            })
        }
    )
}
}

```

```

        fimJogo = false
        vitoria = false
        showDesistirDialog = false
    }) {
        Text("Sim")
    }
},
dismissButton = {
    TextButton(onClick = {
        currentImage = R.drawable.giveup
        fimJogo = true
        vitoria = false
        showDesistirDialog = false
    }) {
        Text("Não")
    }
}
)
}
...

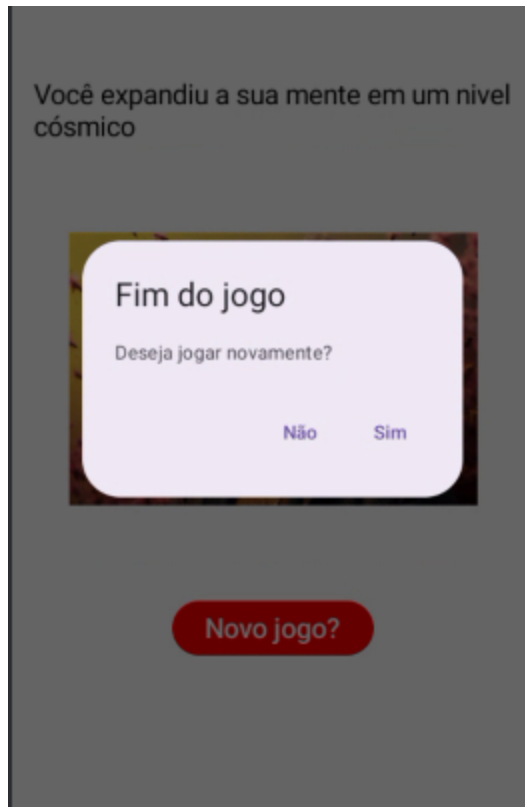
```



5. **Finalização do Jogo**

Ao atingir o número necessário de cliques, o jogador vence, e um diálogo final pergunta se ele deseja jogar novamente. Se o jogador escolher "Sim", o jogo é reiniciado. Se "Não", o aplicativo é fechado.

```
```kotlin
if (showFinalDialog) {
 AlertDialog(
 onDismissRequest = { showFinalDialog = false },
 title = { Text("Fim do jogo") },
 text = { Text("Deseja jogar novamente?") },
 confirmButton = {
 TextButton(onClick = {
 click = 0
 currentImage = R.drawable.inicial
 fimJogo = false
 vitoria = false
 showFinalDialog = false
 }) {
 Text("Sim")
 }
 },
 dismissButton = {
 TextButton(onClick = {
 activity?.finish()
 showFinalDialog = false
 }) {
 Text("Não")
 }
 }
)
}
```
```



6. **Botão de Novo Jogo**

Ao final do jogo (seja por desistência ou vitória), o botão "Novo jogo?" é exibido para reiniciar o jogo. Ele redefine o estado e gera um novo número aleatório de cliques necessários para vencer.

```
```kotlin
Button(onClick = {
 click = 0
 currentImage = R.drawable.inicial
 fimJogo = false
 vitoria = false
 showFinalDialog = false
}, colors = ButtonDefaults.buttonColors(
 containerColor = Color.Red
)) {
 Text(text = "Novo jogo?",
 color = Color.White,
 fontSize = 20.sp
)
}
```
```



Novo jogo?

Conclusão

Projeto de jogo simples.