

@State is only for temporary animation, it is not for storing the state of the game.

@State is a property wrapper in SwiftUI that allows you to create state variables for your views. State variables are mutable data that is associated with a view. When the state changes, the view is automatically updated.

@State is typically used for temporary animations, such as animating a view's position or color. For example, the following code animates the position of a Text view:

```
Swift
struct AnimationView: View {
    @State private var offset = 0.0

    var body: some View {
        Text("Hello, world!")
            .offset(x: offset, y: 0.0)
    }
}
```

When the user taps the screen, the `offset` state variable will be changed and the Text view will be animated to its new position.

However, @State is not for storing the state of the game. The state of the game should be stored in a more permanent location, such as a database or a file. This is because @State variables are lost when the view is recreated.

For example, if the user navigates away from the view and then back to it, the @State variables will be reset to their initial values. This would cause the game to lose its state.

To store the state of the game, you can use a global variable, a database, or a file. For example, the following code stores the state of the game in a global variable:

```
Swift
struct GameView: View {
    var body: some View {
        Text("\(gameState)")
    }
}

var gameState = 0
```

The `gameState` variable is a global variable that stores the state of the game. When the game state changes, the `gameState` variable will be updated and the Text view in the `GameView` will be automatically updated.

Storing the state of the game in a global variable is a simple solution, but it is not very scalable. For more complex games, it is better to use a database or a file to store the state of the game.

Overall, `@State` is a powerful tool for temporary animations, but it is not for storing the state of the game. The state of the game should be stored in a more permanent location, such as a database or a file.