

Stanford CS193p

Developing Applications for iOS
Spring 2023
Lecture 10



This Week

• Demo

Start a brand-new app called EmojiArt

• Drag and Drop

How to transfer data directly between (and within) apps using drag and drop gestures

Demo: Drag and drop of the background of our Emoji Art document

Demo: Drag and drop of the emojis that make up our Emoji Art document

• Gestures

Handling drags, pinches, taps, etc.

Demo: Resizing and panning around in our document

• Multiple MVVMs

Demo: Add an emoji palette-choosing UI to Emoji Art (an entirely new MVVM)

Demo: context menus, .id and .clipped view modifiers

• Colors, Images, Multithreading, Error Handling



Drag and Drop

• Transferable

Drag and drop is easy.

You just have to represent the data you want to transfer as a Transferable.

Common structs like String, Int, Data, etc. implement Transferable.

A View can be a drag initiator simply by applying the `.draggable(Transferable)` modifier.

```
Text(emoji).draggable(emoji) // emoji is a String
```

When the user presses and holds and then drags, the whole Text will start dragging.

The data being dragged (i.e. the Transferable) will be the emoji String.



Drag and Drop

Transferable

A View can be dropped upon if they have a **dropDestination** modifier.

```
@State private var highlighted = false
Rectangle()
    .stroke(lineWidth: highlighted ? 5 : 1)
    .dropDestination(for: String.self) { [String], CGPoint in
        // deal with those Strings being dropped at CGPoint in the Rectangle
    } isTargeted: { isTargeted: Bool in
        // highlight something or otherwise indicate a drop is overhead
        highlighted = isTargeted
    }
```

Unfortunately only one **dropDestination** per View is allowed.

So if you want to drop Strings and Ints on a View, you have to create a custom Transferable.
We'll do this in the demo.

