



//Touch Bar

{Présentation}

(CocoaHeads Montpellier)

/* 15 décembre 2016 */

/* @razvanbunea */



//Touch Bar

{Présentation}

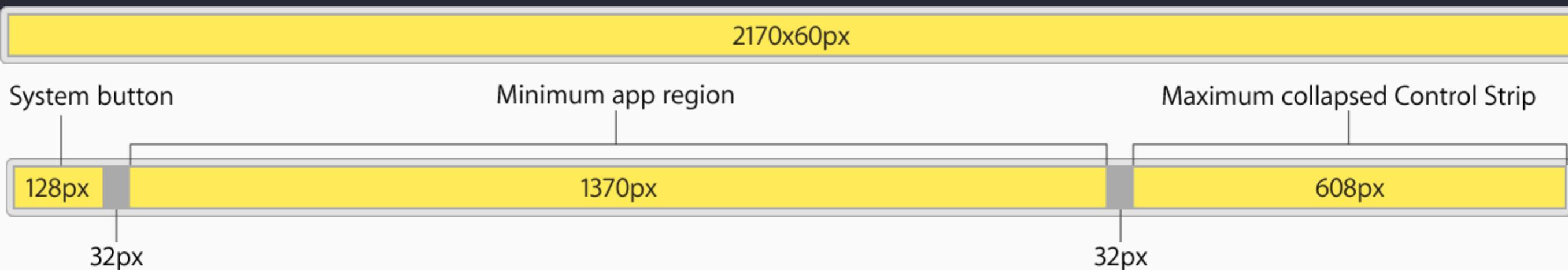
(CocoaHeads Montpellier)

//A propos



- > Permet à l'utilisateur d'interagir avec l'app
- > Nécessite Xcode 8.1 (ou supérieur) et macOS 10.12.1 build 16B2657 (ou supérieur)
- > Xcode permet d'afficher un émulateur de Touch Bar
- > API simple et complète

//Structure



- > Bouton système (Esc, Annuler, OK)
- > Partie spécifique à l'application
- > Bande des commandes systèmes (varie en fonction de la configuration de l'utilisateur)

//Spécifications

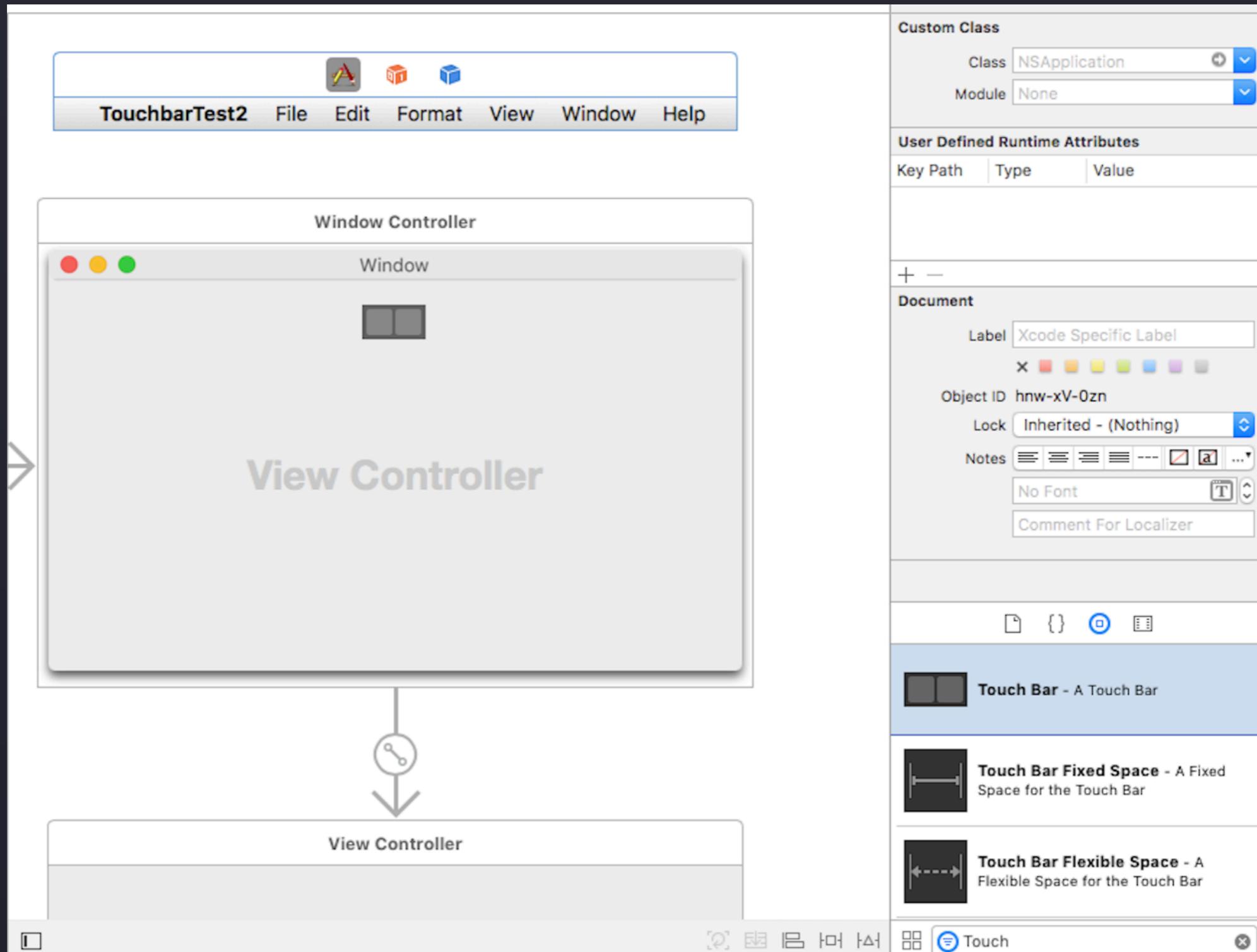
- > Support des gestes
 - > Touché
 - > Touché et rester appuyé
 - > Swipe horizontal
 - > Multitouch

Guidelines

- > Faire une expérience contextuelle
- > Utiliser la TB comme extension du clavier / trackpad (pas un affichage)
- > Essayer de ressembler à un clavier physique
- > Ne pas proposer des fonctionnalités uniquement dans la Touch Bar
- > Fournir des contrôles qui donnent un résultat immédiat
- > ...

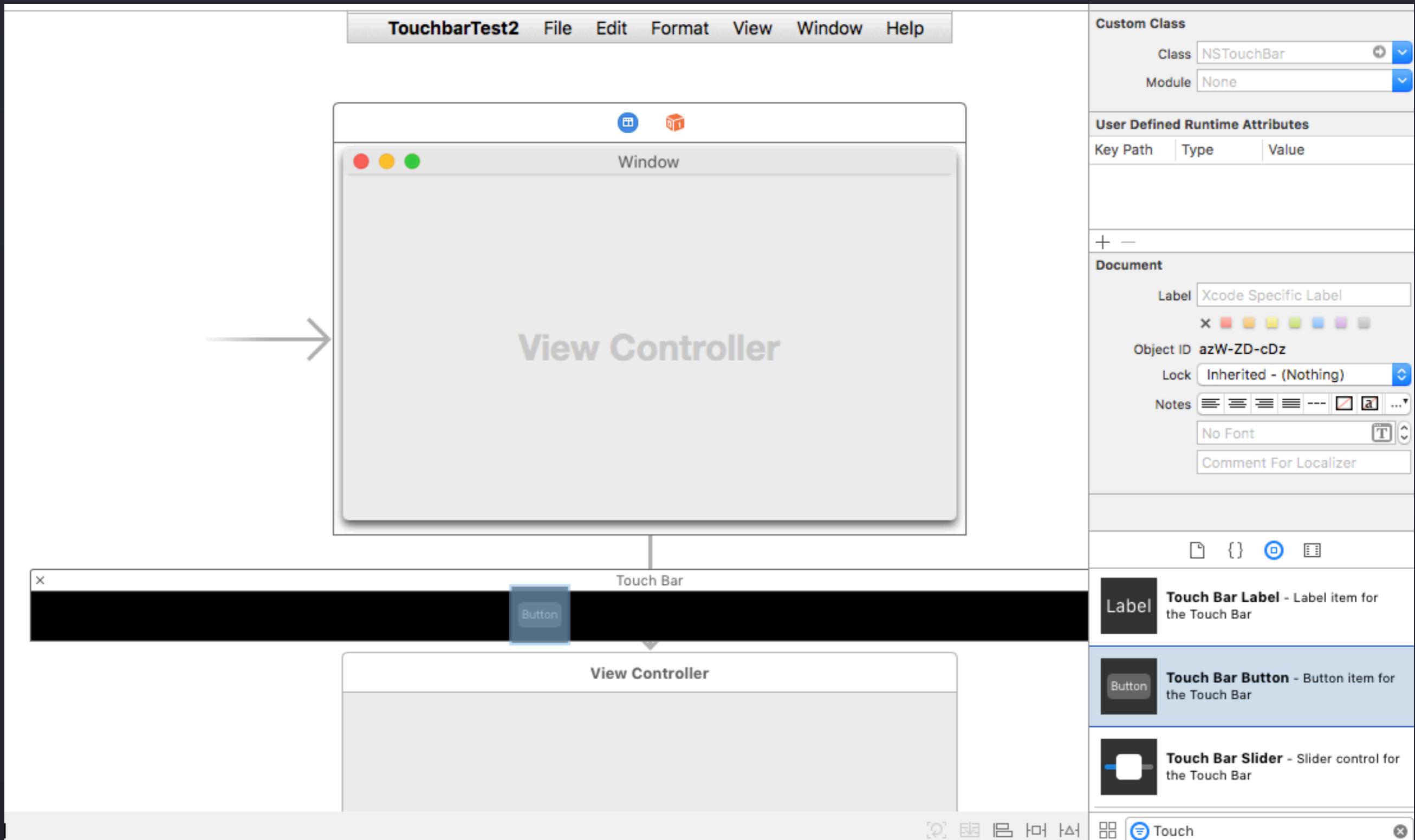
//Intégration - Storyboard

> Glisser l'objet Touch Bar sur le Window Controller



//Intégration - Storyboard

> Glisser l'objet Touch Bar sur le Window Controller



//Intégration – Code (1)

- > Activer l'utilisation de la Touch Bar dans l'AppDelegate

```
class AppDelegate: NSObject, NSApplicationDelegate {  
  
    func applicationDidFinishLaunching(_ aNotification: Notification) {  
        if #available(OSX 10.12.1, *) {  
            NSApplication.shared().isAutomaticCustomizeTouchBarMenuItemEnabled = true  
        }  
    }  
}
```

- > Créer une custom WindowController et l'affecter à la fenêtre dans Storyboard

```
class MyWindowController: NSWindowController {  
  
    //Demande à la fenêtre d'utiliser la Touch Bar déclaré dans le viewController  
    @available(OSX 10.12.1, *)  
    override func makeTouchBar() -> NSTouchBar? {  
        guard let viewController = contentViewController as? ViewController else {  
            return nil  
        }  
        return viewController.makeTouchBar()  
    }  
}
```

//Intégration - Code (2)

> Dans la classe ViewController

> Définir les identifiants Touch Bar et Items

```
//Définir l'identifiant de la Touch Bar
fileprivate extension NSTouchBarCustomizationIdentifier {
    static let customTouchBar = NSTouchBarCustomizationIdentifier("net.razvan.touchbar.customTouchBar")
}
```

```
//Définir l'identifiant des éléments personnalisés
fileprivate extension NSTouchBarItemIdentifier {
    static let helloItem = NSTouchBarItemIdentifier("net.razvan.touchbar.items.helloItem")
    static let imageItem = NSTouchBarItemIdentifier("net.razvan.touchbar.items.imageItem")
    static let okItem = NSTouchBarItemIdentifier("net.razvan.touchbar.items.okItem")
}
```

> Se déclarer comme TouchBarDelegate

```
@available(OSX 10.12.1, *)
extension ViewController: NSTouchBarDelegate {
    // ... code des pages suivantes
}
```

//Intégration - Code (3)

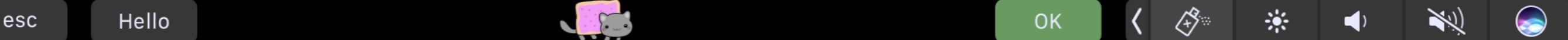
> Construire la Touch Bar

```
override func makeTouchBar() -> NSTouchBar? {  
    //Créer la TouchBar  
    let touchBar = NSTouchBar()  
    touchBar.delegate = self  
    //affecter l'identifiant de la barre  
    touchBar.customizationIdentifier = .customTouchBar  
    //affecter les items à afficher et l'ordre d'affichage dans la touch bar  
    touchBar.defaultItemIdentifiers = [.helloItem,.flexibleSpace,.imageItem,.flexibleSpace,.okItem]  
    //liste des items personnalisables  
    touchBar.customizationAllowedItemIdentifiers = [.helloItem, .imageItem, .okItem]  
  
    return touchBar  
}
```

> Implementer la méthode du NSTouchBarDelegate (construit les items personnalisés)

//Intégration – Code (4)

```
//NSTouchBarDelegate - method
//création des items personnalisés
func touchBar(_ touchBar: NSTouchBar, makeItemForIdentifier identifier: NSTouchBarItemIdentifier) ->
NSTouchBarItem? {
    switch identifier {
        case NSTouchBarItemIdentifier.helloItem:
            let customViewItem = NSCustomTouchBarItem(identifier: identifier)
            customViewItem.view = NSButton(title: "Hello", target: self, action: #selector(sayHello))
            return customViewItem
        case NSTouchBarItemIdentifier.okItem:
            let okItem = NSCustomTouchBarItem(identifier: identifier)
            let button = NSButton(title: "OK", target: self, action: #selector(sayOk))
            button.bezelColor = NSColor(red:0.35, green:0.61, blue:0.35, alpha:1.00)
            okItem.view = button
            return okItem
        case NSTouchBarItemIdentifier.imageItem:
            let viewItem = NSCustomTouchBarItem(identifier: identifier)
            let image = NSImageView(image: NSImage(named: "nyan1")!)
            viewItem.view = image
            return viewItem
        default:
            return nil
    }
}
```



//Contrôles (1)

esc



Move to...



Button

esc



Add Time



Add Location



Buttons with icons and titles

esc



"quick"

quickest

quickness



Candidate list

esc



A

B

I

U

≡

≡



Character picker

x

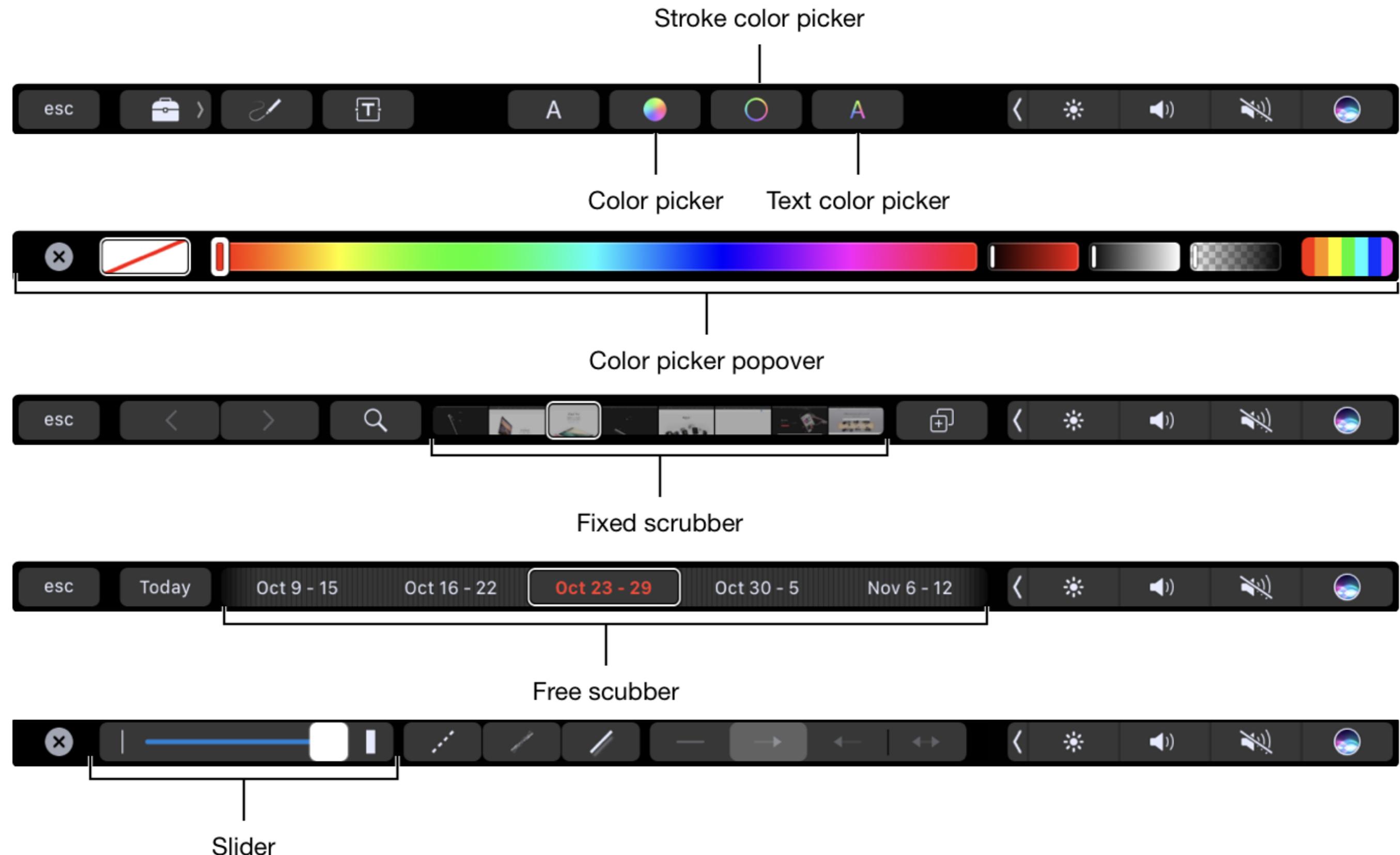


Frequently Used

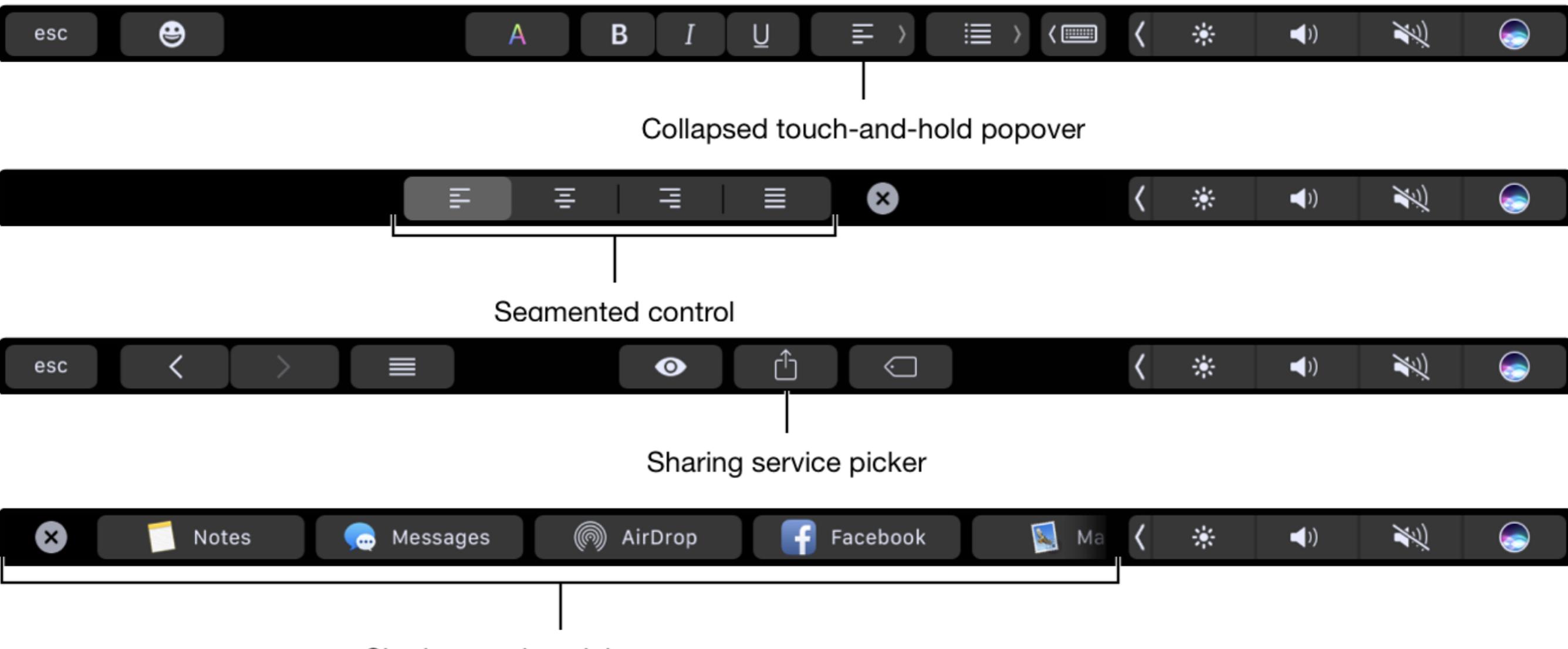


Character picker popover

//Contrôles (2)



//Contrôles (3)



> ■ ■ ■

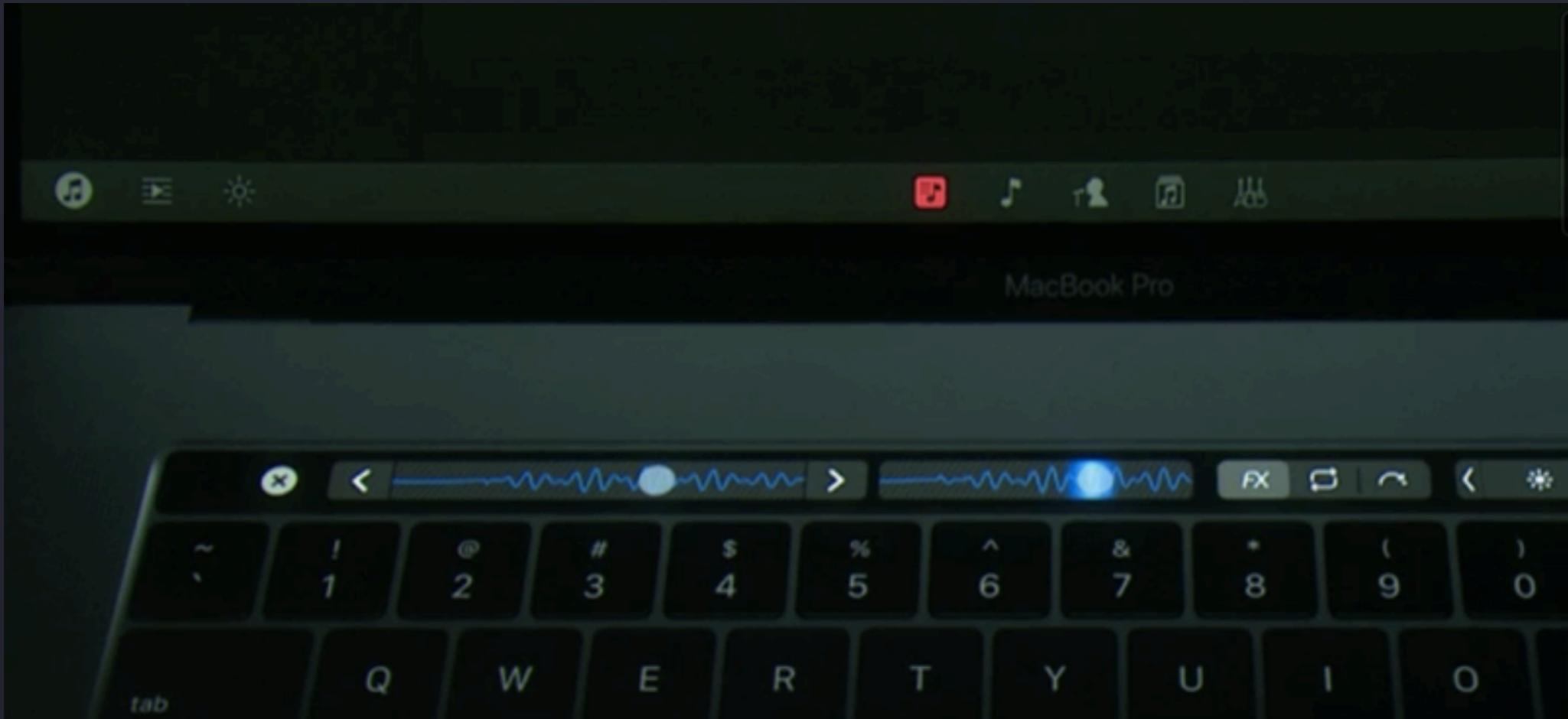
//Contrôles (4)

> Custom views

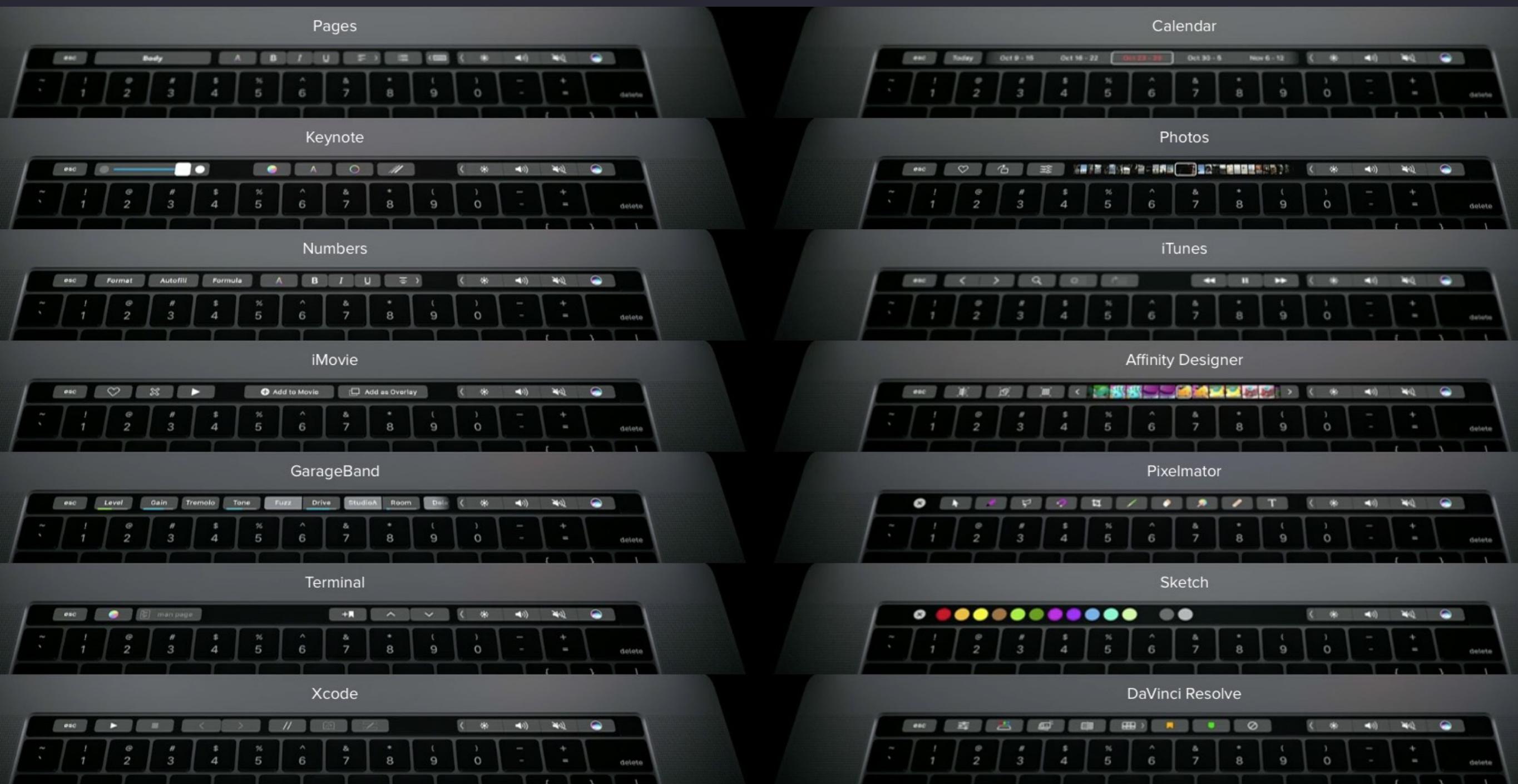


//Contrôles (4)

> Custom views



//Intégration



//Touché



Touché

Touch Bar for everyone.

Simulate, show and hide, take screen captures.

Free Download for Mac. Use and enjoy. [More info.](#)

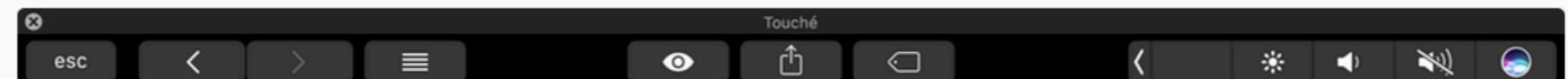
Easy On, Easy Off

Touché lets you assign a global keyboard shortcut for easy toggling of the simulated Touch Bar window. Leave Touché running, and when inspiration to see the Touch Bar strikes? Just press the keyboard shortcut.

Streamlined Screenshots

See something in the Touch Bar that you want to share with a friend or colleague? Touché provides global keyboard shortcuts for saving a screen capture to disk, or copying it directly to the pasteboard.

Screenshot



Touché Touch Bar.
Click to view larger [image](#).

<https://red-sweater.com/touche/>

//Liens

- > Mise jour macOS Sierra 10.12.1 (build 16B2657)
https://support.apple.com/kb/dl1897?locale=fr_FR
- > macOS Human Interface Guidelines
<https://developer.apple.com/library/content/documentation/UserExperience/Conceptual/OSXHIGuidelines/AbouttheTouchBar.html>
- > Touché
<https://red-sweater.com/touche/>
- > GitHub projet
<https://github.com/razvn/touchbardemo-macOS/>

//Live Demo