

# Bonus de CountOnMe

Le bonus que j'ai choisi d'inclure dans l'application CountOnMe est de pouvoir effectuer une multiplication ou une division.

## La transformation des touches + et -

Pour que cela soit possible, j'ai rajouté un appui long sur le bouton plus pour le transformer en bouton multiplier :



Et sur le bouton moins pour le transformer en bouton diviser :



Une fois un des deux boutons relâchés, ils reviennent aux signes d'origines.

## Intégration du bonus dans le code


Pour que les boutons supportent la pression, plusieurs méthodes qui prennent en charge la sous classe `UILongPressGestureRecognizer` sont nécessaires et ainsi qu'une configuration du début et de la fin de la pression des boutons + et - qui se transforment en signes x ou ÷.


```
// MARK: - Configure press signs buttons
extension CalculatorViewController {


    private func longPressGestureRecognizerOnPlusOrMultiplyButton() {
        let longGesture = UILongPressGestureRecognizer(target: self, action: #selector(configurePressPlusOrMultiplyButton(_)))
        plusOrMultiplyButton.addGestureRecognizer(longGesture)
    }

    private func longPressGestureRecognizerOnMinusOrDivideButton() {
        let longGesture = UILongPressGestureRecognizer(target: self, action: #selector(configurePressMinusOrDivideButton(_)))
        minusOrDivideButton.addGestureRecognizer(longGesture)
    }

    private func changeColorButton(_ button: UIButton, _ color: UIColor) {
        button.backgroundColor = color
        button.tintColor = color
    }

    private func transformButtonSign(_ sender: UILongPressGestureRecognizer, _ button: UIButton, _ sign: String) {
        button.isSelected = true
        changeColorButton(button, )
        addOperator(sign)
    }

    @objc private func configurePressPlusOrMultiplyButton(_ sender: UILongPressGestureRecognizer) {
        switch sender.state {
        case .began:
            transformButtonSign(sender, plusOrMultiplyButton, "x")
        case .ended, .cancelled:
            plusOrMultiplyButton.isSelected = false
            changeColorButton(plusOrMultiplyButton, )
        default:
            break
        }
    }

    @objc private func configurePressMinusOrDivideButton(_ sender: UILongPressGestureRecognizer) {
        switch sender.state {
        case .began:
            transformButtonSign(sender, minusOrDivideButton, "÷")
        case .ended, .cancelled:
            minusOrDivideButton.isSelected = false
            changeColorButton(minusOrDivideButton, )
        default:
            break
        }
    }
}
```

Sans oublier de rajouter les calculs des nouveaux signes multiplier et diviser dans la méthode `calculateTotal()` de la classe `CalculationManager()` du model.

```
func calculateTotal() {  
    for (index, stringNumber) in stringNumbers.enumerated() {  
        if let number = Double(stringNumber) {  
            if operators[index] == "+" {  
                total += number  
            } else if operators[index] == "-" {  
                total -= number  
            } else if operators[index] == "x" {  
                total *= number  
            } else if operators[index] == "÷" {  
                total /= number  
            }  
        }  
    }  
}
```