

Instagrid

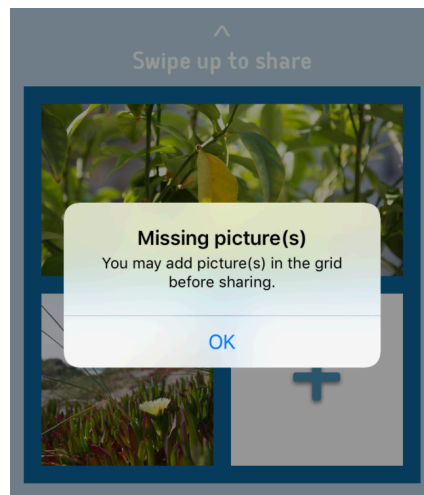
Projet 5 du parcours développeur iOS

Mes bonus pour ce jeu sont un **contrôle de la grille avant tout partage** et la possibilité de **supprimer une image de la grille**.

1// Contrôle de la grille

Selon la disposition sélectionnée, un contrôle est fait pour vérifier que les cases visibles de la grille contiennent bien des images.

En cas d'images manquantes, le partage de la création est impossible et un message d'alerte apparaît.



Message d'alerte lors du swipe d'une grille incomplète

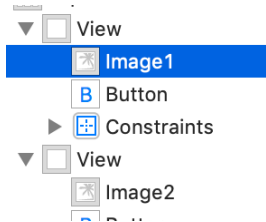
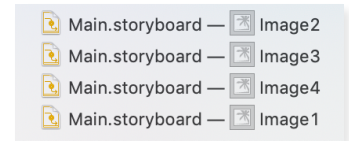
Intégration du bonus dans le code

Dans la classe **LayoutView** qui hérite de **UIView**, j'ai ajouté la méthode **missingPictureInGrid()**. Cette méthode permet de contrôler les images manquantes d'une grille :

```
func missingPictureInGrid() -> Bool {
    for picture in picturesCollectionView {
        if let parentView = picture.superview {
            if parentView.isHidden == false && picture.image == nil {
                return true
            }
        }
    }
    return false
}
```

picturesCollectionView est un tableau de type UIImageView, contenant les vues des images 1 à 4 de la grille :

```
@IBOutlet var picturesCollectionView: [UIImageView]!
```



Pour une UIImageView contenue dans ce tableau, si sa vue parente n'est pas cachée par la disposition de la grille choisie et que l'UIImageView ne contient pas d'image, la méthode renvoie **true** pour indiquer que **la grille est incomplète**.

Dans le cas contraire, la méthode renvoie **false**, il ne manque pas d'images.

Dans la classe ViewController :

```
@objc func mainViewSwiped(gesture: UISwipeGestureRecognizer) {
    UIView.animate(withDuration: 0.3) {
        if gesture.direction == .up && UIDevice.current.orientation.isPortrait {
            self.mainView.transform = CGAffineTransform(translationX: 0, y: -self.view.frame.height)
            print("swipe up")
            if self.mainView.missingPictureInGrid() {
                self.alertGridIsIncomplete()
                self.swipeLayoutSetToIdentity()
            } else {
                self.handleShare()
            }
        } else if gesture.direction == .left && UIDevice.current.orientation.isLandscape {
            self.mainView.transform = CGAffineTransform(translationX: -self.view.frame.width, y: 0)
            print("swipe left")
            if self.mainView.missingPictureInGrid() {
                self.alertGridIsIncomplete()
                self.swipeLayoutSetToIdentity()
            } else {
                self.handleShare()
            }
        }
    }
}
```

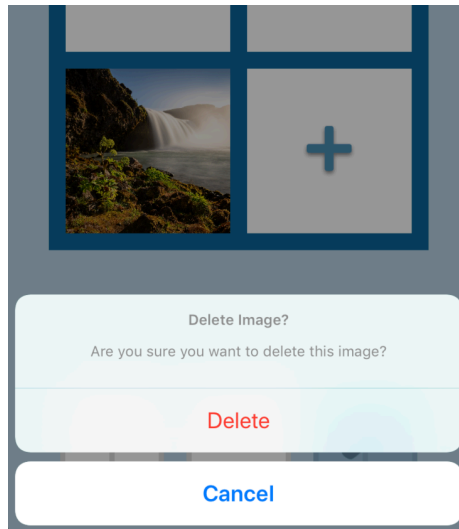
Si missingPictureInGrid() renvoie true, la grille reste centrée malgré le swipe (méthode swipeLayoutSetToIdentity()), et le message d'alerte apparaît :

```
private func alertGridIsIncomplete() {
    let alert = UIAlertController(title: "Missing picture(s)", message: "You may add picture(s) in the grid before sharing.", preferredStyle: .alert)
    alert.addAction(UIAlertAction(title: NSLocalizedString("OK", comment: "Default action"), style: .default, handler: { _ in
        NSLog("The \"OK\" alert occurred.")
    }))
    self.present(alert, animated: true, completion: nil)
}
```

Si missingPictureInGrid() renvoie false, la fenêtre de partage s'affiche avec la méthode handleShare().

2// Supprimer une image de la grille

L'utilisateur a la possibilité de supprimer une image de la grille en restant appuyé dessus. Un message d'alerte apparaît en bas de l'écran pour permettre d'effacer l'image ou bien d'annuler.



Message d'alerte lors d'un longPress sur l'image

Intégration du bonus dans le code

J'ai d'abord ajouté un gesture recognizer (UILongPressGestureRecognizer) pour chaque UIImageView de la grille.

Dans la fonction **longPressed()**, tag est de type Int?, il permet de récupérer le tag de la vue du geste. Ce tag est ensuite utilisé dans la closure deleteAction pour supprimer la bonne image.

```
private func longPressGestureRecognizer() {
    mainView.picturesCollectionImageView.forEach( { $0.addGestureRecognizer(UILongPressGestureRecognizer(target:
        self, action: #selector(longPressed(gesture:))) ) } )
}

/**
 * Allow to delete picture by long press gesture on it
 */
@objc private func longPressed(gesture: UILongPressGestureRecognizer) {
    tag = gesture.view?.tag
    guard let tag = tag else { return }
    if gesture.state == .ended { return }
    let title = "Delete Image?"
    let message = "Are you sure you want to delete this image?"
    let alert = UIAlertController(title: title, message: message, preferredStyle: .actionSheet)
    let cancelAction = UIAlertAction(title: "Cancel", style: .cancel, handler: nil)
    alert.addAction(cancelAction)
    let deleteAction = UIAlertAction(title: "Delete", style: .destructive, handler: { (action) -> Void in
        self.mainView.picturesCollectionImageView[tag].image = nil // delete image
        self.mainView.picturesButton[tag].isHidden = false // show picture selection button
    })
    alert.addAction(deleteAction)
    present(alert, animated: true, completion: nil)
}
```

Deux alertes, « deleteAction » et « cancelAction », permettent de supprimer l'image sélectionnée, ou bien d'annuler et retourner sur la grille.