Instagrid

Concept

Instagrid permet aux utilisateurs de créer des collages photographiques et de les partager.

Les images peuvent provenir de 3 sources

1. la bibliothèque photo
2. prendre des photos avec l'appareil photo du téléphone (bonus 2)
3. d'une recherche dans les images de stock de `Pixabay` (Bonus 3)

Vous pouvez trouver la documentation du Pixabay RESTAPI à l'adresse suivante:
https://pixabay.com/api/docs/

Une fois votre collage créer vous pouvez éditer/améliorer vos photos :

- en appliquant des filtres (Bonus 4)
- en ajoutant des émoticones (bonus 5)
- en écrivant un message sur l'image (bonus 6)

Bonus 1 : Compatibilité Ipad

The UIImagePickerController est présenté dans un Popover et non de manière modale. Nous avons donc mis en place une verification du device

```
if UI_USER_INTERFACE_IDIOM() == UIUserInterfaceIdiom.pad {
    if let PopOver = controller.popoverPresentationController {
        PopOver.sourceView = self.view
    }
        present( controller, animated: true, completion: nil )
    } else{
        present(controller, animated: true, completion: nil)
    }
}
```

Nous vérifions aussi l'orientation afin de présenter le UIActivityViewController dans le bon sens Nous détectons l'orientation.

```
if UIDevice.current.orientation.isLandscape {
    landscapeOrientation = true
    print("Landscape")
} else{
    print("Portrait")
    landscapeOrientation = false
}
```

Nous forçons l'orientation

```
let landscapeRawValue = UIInterfaceOrientation.landscapeLeft.rawValue
UIDevice.current.setValue(landscapeRawValue, forKey: "orientation")
```

1-Importer des contenus

Bonus 2 : Prendre des photos

Nous avons rajouté une source .camera au UIImagePickerController et autorisé l'utilisationde l'appareil photo dans l'info.plist.

```
if(UIImagePickerController .isSourceTypeAvailable(UIImagePickerControllerSourceTyp
e.camera))
    {
      image.sourceType = UIImagePickerControllerSourceType.camera
    }
    else
    {
      let alert  = UIAlertController(title: "Warning", message: "You don't have ca
mera", preferredStyle: .alert)
      alert.addAction(UIAlertAction(title: "OK", style: .default, handler: nil))
      self.present(alert, animated: true, completion: nil)
    }
```

Bonus 3: WebSearch Conenxion API Pixabay

Dans l'AlertViewController, une option propose de chercher sur Pixabay. Une fois sélectionner nous présentons un controller avec une navigation proposant un champ input , un bouton search et un bouton cancel. Vous entrez votre recherche dans le champs input.

1. Nous avons créer un struct APIClient en charge de se connecter et poster une requete GET à Pixabay
2. Un objet Model pour formatter les réponses
3. Une UICollectionView pour afficher les résultats

Quand l'utilisateur tapent sur une image, elle est renvoyer dans son container originel dans le

ViewController principal.

Les valeurs de propriété imageToSend sont passées par délégation.

2-Modifier des contenus

Un controller nommé EditorViewController est affiché de manière modale. Il présente:

- l'image que l'utilisateur souhaite éditer
- 2 boutons en partie basse (cancel pou annuler et use pour renvoyer l'image modifiée dans le controller principal)
- 3 boutons en partie supérieure (un bouton pour appliquer des filtres, un bouton pour ajouter un emoticone, un bouton pour rajouter du texte)

Bonus 4 : Les filtres

Ce bonus s'appuie sur les fonctionnalités du framework CoreImages

```
- CIPhotoEffectNoir
- CIPhotoEffectChrome
- CIPhotoEffectInstant
- CIPhotoEffectTransfer
- CIPhotoEffectProcess
- CIPhotoEffectTonal
```

Une barre de boutons proposant une preview de chaque effet apparait sous l'image, si l'utilisateur appuie dessus, l'effet est appliqué. Tant que l'utilisateur n'a pas cliqué sur use il peut modifié son choix.

Bonus 5 : ajouter des emoticones sur une image

En cliquant sur l'icone supérieure représentatn un smiley, un clavier de smiley est affiché modalement dans la partie inférieure.

techniquement, il s'agit une collectionView qui affiche le contenu d'un tableau d'image stocké dans le dossier xcAsset.

Une fois que l'utilisateur tape sur un emoticone, il apparait au centre de l'image.

Dès lors l'utilisateur peut à l'aide d'un PanGesture déplacé l'image dans les limites du cadre de celle-ci

et/ou agrandir, rétrécir grace à un PinchGesture.

Bonus 6 : ajouter un texte sur une image

En cliquant sur l'icone supérieure représentatn un smiley, un textField apparait au centre de l'image. L'utilisateur peut alors ecrire le texte de son choix.

Un panneau apparait en partie inférieur proposant différentes polices de caratères et couleurs.

Dès lors l'utilisateur peut à l'aide d'un second PanGesture déplacé l'image dans les limites du cadre de celle-ci

et/ou agrandir, rétrécir grace à un second PinchGesture.

C'est deux dernière fonctionnalités reposent sur les limites suivantes

```
  // Clamp the textField movement in the container  frame
    if newOrigin.x >= emoji.frame.width / 2 && newOrigin.y >= emoji.frame.height /
 2 && newOrigin.x <= container.frame.width - emoji.frame.width / 2 && newOrigin.y
<= container.frame.height - emoji.frame.height / 2{

      // update the coordinates of the textField
      let newFrame = CGRect(x: newOrigin.x - (emoji.frame.width / 2), y: newOrigin
.y - (emoji.frame.height / 2), width: emoji.frame.width, height: emoji.frame.heigh
t)
      emoji.frame = newFrame
    }
    else {
      print("Out of bounds")
    }
```