Gridy



# Launch Screen

Plain green screen with the Icon

# Introduction Screen

Buttons to start game with either Photo from Gridy, the Photo Library or take a new photo.

Button to view scoreboard

Button to view instruction on how to play

### Scoreboard

Show top 10 Scores with button back to main menu

### Instructions

Simple how to play with button back to main menu

# Photo Selection

### Photo Library

Opens Photo library selection

### Gridy Photos

Provides grid of photos held in app

### Camera Photo

Opens Camera to allow a new photo to be taken

Once item selected – button to goto next screen

### Selected Image View

Shows selected image, with ability to pull to resize and crop.

Selection of Grid sizes to choose to play (Larger grid selection for larger screen like iPad)

# Game Play

Image in scrambled grid at top of screen

Blank grid for photo to be pulled into piece by piece

Hint Button to show complete image on screen

Help button to place 1 piece in correctly

Count up timer

Counter for moves taken

Rotate button – once piece is placed on grid so player can rotate image piece

Quit Game button

# Game Complete

Shows score with moves and time taken

Input for name to record score

Once name is entered goes to scoreboard

## Things to do -

* Add Photos and icons to Asset Catalog – add atleast 5 of my own photos
* Import TimeBurner tff and update plist to include font in selection
* Add Unique colours to palette to make selection easier
* Library, Camera – UIPIckerController

func displayLibrary() {

let sourceType = UIImagePickerControllerSourceType.photoLibrary

if UIImagePickerController.isSourceTypeAvailable(sourceType) {

let status = PHPhotoLibrary.authorizationStatus()

let noPermissionMessage = "Looks like the app does not have access to you phot library"

switch status {

case.notDetermined:

PHPhotoLibrary.requestAuthorization({ (newStatus) in

if newStatus == .authorized {

self.presentImagePicker(sourceType: sourceType)

}

else {

self.troubleAlert(message: noPermissionMessage)

}

})

case .authorized:

self.presentImagePicker(sourceType: sourceType)

case . denied, .restricted:

self.troubleAlert(message: noPermissionMessage)

}

}

else {

troubleAlert(message: "Cannot access Photo Library")

}

}

func presentImagePicker(sourceType: UIImagePickerControllerSourceType){

let imagePicker = UIImagePickerController()

imagePicker.delegate = self

imagePicker.sourceType = sourceType

present(imagePicker, animated: true, completion: nil)

}

* For Camera – check to see if access has been granted or not – grant access
* Reusable views
  + Photo Grid (Game Play)
    - Image grid
    - Image holders
    - Picked image holder – to compare
  + Image Crop
    - Image
    - View
  + Scoreboard
* Save scores – even when app closed (AppDelegate)
* View Models
  + Score calculation
  + Photo split