

# Puzzle

## Développement pour mobiles M1

Payet Jeremy

November 9, 2018

# Sommaire

## Sommaire

### Introduction

Quel est le projet ?

Comment a-t-il été mis en place ?

### Détail du projet

Réalisation pour Android

Réalisation pour iOS

### Conclusion

- Introduction
  - Quel est le projet ?
  - Comment a-t-il été mis en place ?
- Détail du projet
  - Réalisation pour Android
  - Réalisation pour iOS
- Conclusion

# Quel est le projet ?

## Sommaire

### Introduction

Quel est le projet ?

Comment a-t-il été mis en place ?

### Détail du projet

Réalisation pour Android

Réalisation pour iOS

### Conclusion

Pour ce projet, nous avons différentes notions à implémenter dans une application de notre choix:

- Changements de configuration
- Internationalisation
- Géolocalisation
- Capteurs
- Lecture audio
- Gestes courants
- Appareil photo

# Comment a-t-il été mis en place ?

## Sommaire

### Introduction

Quel est le projet ?

Comment a-t-il été mis en place ?

### Détail du projet

Réalisation pour Android

Réalisation pour iOS

### Conclusion

Comme il fallait implémenter tout ces éléments, il a fallu savoir à quoi peut servir chaque notion. Le changement de configuration et l'internationalisation ont été utilisés pour la langue et fixer au format portrait, les capteurs pour le mélange de puzzle en secouant l'appareil mobile, les gestes courants pour déplacer les images, et l'appareil photo pour créer l'image qui servira de puzzle. La géolocalisation a été considérée pour connaître l'origine des photos, mais n'a pas été implémentée.

# Réalisation pour Android

## Sommaire

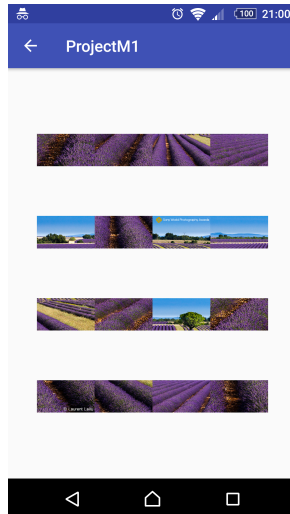
### Introduction

Quel est le projet ?  
Comment a-t-il été mis en place ?

### Détail du projet

Réalisation pour Android  
Réalisation pour iOS

### Conclusion



# Réalisation pour Android

## Sommaire

### Introduction

Quel est le projet ?

Comment a-t-il été mis en place ?

### Détail du projet

Réalisation pour Android

Réalisation pour iOS

### Conclusion

```
intent.putExtra( name: "data", extras.getBytes( key: "data" ));

settings = getSharedPreferences ( name: "rand", MODE_WORLD_READABLE );
SharedPreferences.Editor editor = settings.edit();
editor.remove("rand");
new_positions(ids,imageChunks,listrand);

//System.out.println("filepath : "+ listrand[0]);
// Apply the edits!
for (int i=0;i<listrand.length;i++){
    editor.putInt( s: "Place "+i,listrand[i]);
    System.out.println("liste["+i+"] = "+listrand[i]);
}
editor.putString( s: "path",path);
editor.commit();

System.out.println("Filepath : "+ settings.getString( s: "path", s1: "" ) + " commit " + editor.commit());
}
```

# Réalisation pour iOS

## Sommaire

## Introduction

Quel est le projet ?  
Comment a-t-il été mis en place ?

## Détail du projet

Réalisation pour Android  
Réalisation pour iOS

## Conclusion

```
<plugin identifier="com.apple.InterfaceBuilder.IBocoaTouchPlugin" version="13772"/>
<capability name="Safe area layout guides" minToolsVersion="9.0"/>
<capability name="documents saved in the Xcode 8 format" minToolsVersion="8.0"/>
</dependencies>
<scenes>
  <!--View Controller-->
  <scene sceneID="tne-QT-ifu">
    <objects>
      <viewController id="BYZ-38-t0r" customClass="ViewController" customModule="ProjectM1" customModuleProvider="target" sceneM
      <view key="view" contentMode="scaleToFill" id="8bC-Xf-vdC">
        <rect key="frame" x="0.0" y="0.0" width="375" height="667"/>
        <autoresizingMask key="autoresizingMask" widthSizable="YES" heightSizable="YES"/>
        <subviews>
          <button opaque="NO" contentMode="scaleToFill" contentHorizontalAlignment="center" contentVerticalAlignment="ce
          <rect key="frame" x="16" y="146" width="343" height="30"/>
          <state key="normal" title="Continuer"/>
          </button>
          <button opaque="NO" contentMode="scaleToFill" misplaced="YES" contentHorizontalAlignment="center" contentVerti
          <rect key="frame" x="16" y="84" width="343" height="30"/>
          <state key="normal" title="Nouvelle Partie"/>
          <connections>
            <segue destination="CLR-yV-vTp" kind="show" id="jla-qm-iaz"/>
          </connections>
          </button>
          <label opaque="NO" userInteractionEnabled="NO" contentMode="left" horizontalHuggingPriority="251" verticalHugg
          <rect key="frame" x="145" y="36" width="85" height="36"/>
          <fontDescription key="fontDescription" type="system" pointSize="30"/>
          <nil key="textColor"/>
          <nil key="highlightedColor"/>
          </label>
        </subviews>
        <color key="backgroundColor" red="1" green="1" blue="1" alpha="1" colorSpace="custom" customColorSpace="sRGB"/>
        <constraints>
          <constraint firstItem="BYZ-38-t0r" firstAttribute="top" secondItem="8bC-Xf-vdC" secondAttribute="top" priority="1" />
          <constraint firstItem="BYZ-38-t0r" firstAttribute="bottom" secondItem="8bC-Xf-vdC" secondAttribute="bottom" priority="1" />
          <constraint firstItem="BYZ-38-t0r" firstAttribute="centerX" secondItem="8bC-Xf-vdC" secondAttribute="centerX" />
          <constraint firstItem="BYZ-38-t0r" firstAttribute="centerY" secondItem="8bC-Xf-vdC" secondAttribute="centerY" />
        </constraints>
      </viewController>
    </objects>
  </scene>
</scenes>
```

## Sommaire

### Introduction

Quel est le projet ?

Comment a-t-il été mis en place ?

### Détail du projet

Réalisation pour Android

Réalisation pour iOS

### Conclusion

```
8 import UIKit
9
10 class ViewController: UIViewController, UIImagePickerControllerDelegate , UINavigationControllerDelegate {
11
12     @IBOutlet weak var myImg: UIImageView!
13
14     @IBAction func takePhoto(_ sender: AnyObject) {
15         if UIImagePickerController.isSourceTypeAvailable(UIMediaPickerSourceType.camera) {
16             let imagePicker = UIImagePickerController()
17             imagePicker.delegate = self
18             imagePicker.sourceType = UIImagePickerControllerSourceType.camera
19             imagePicker.allowsEditing = false
20             self.present(imagePicker, animated: true, completion: nil)
21         }
22     }
23
24     override func viewDidLoad() {
25         super.viewDidLoad()
26         // Do any additional setup after loading the view.
27     }
28
29     override func didReceiveMemoryWarning() {
30         super.didReceiveMemoryWarning()
31         // Dispose of any resources that can be recreated.
32     }
33
34
35     @IBAction func retour(_ sender: Any) {
36         performSegue(withIdentifier: "back", sender: nil)
37     }
38
39
40
41     @IBAction func cancel(_ sender: UIBarButtonItem) {
```



# Conclusion

## Sommaire

### Introduction

Quel est le projet ?

Comment a-t-il été mis en place ?

### Détail du projet

Réalisation pour Android

Réalisation pour iOS

### Conclusion

Au final, réaliser ce projet m'a posé des difficultés, surtout au niveau iOS. Les méthodes sous Android n'ont pas toutes été implémentées, même si elles ont été pensées.