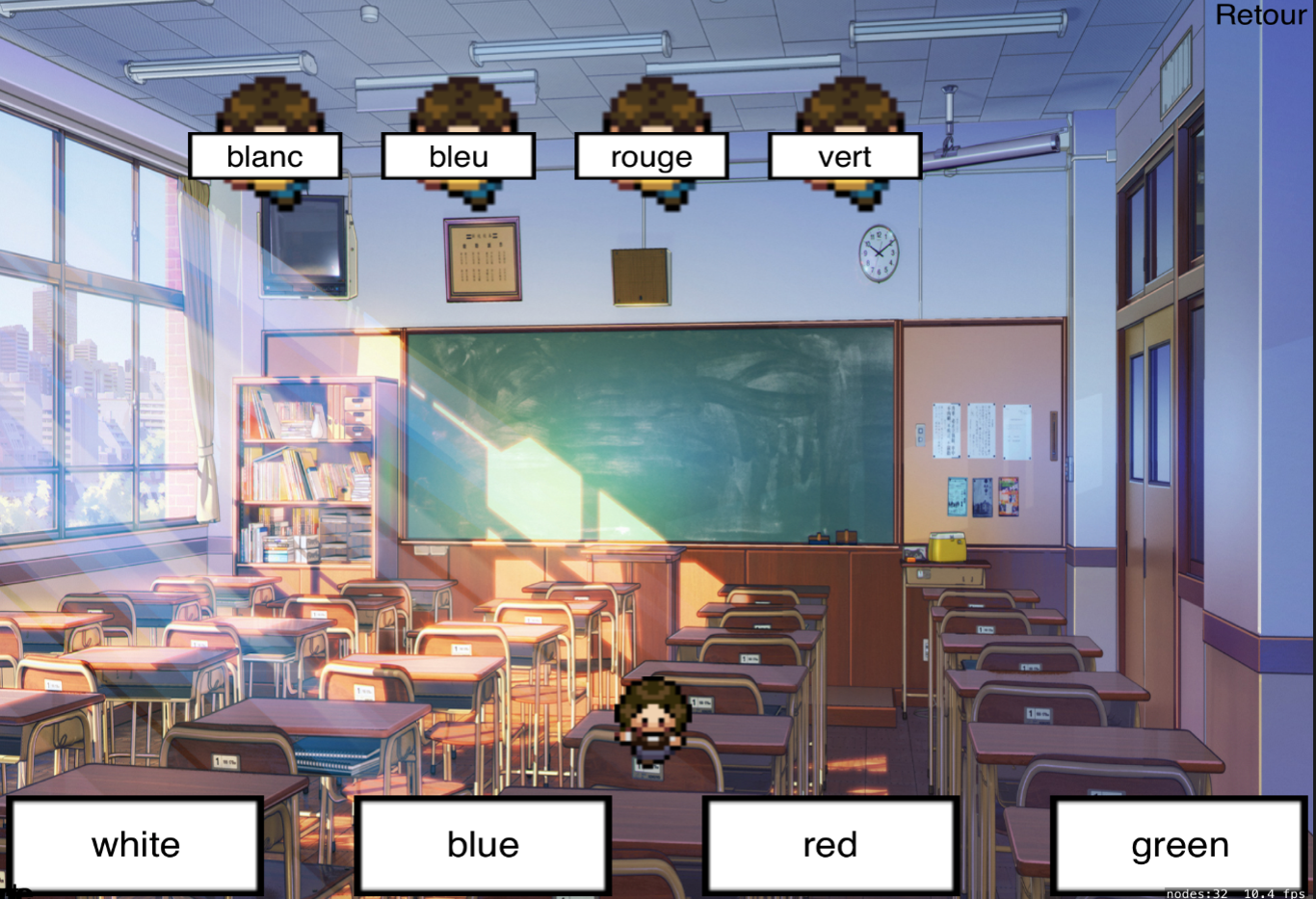
*HARNISCH Mikel*

*Mikel Harnisch*

*SI-C4a*

*Projet TPI 2018*

School Invaders



[1 Analyse préliminaire 3](#_Toc516126811)

[1.1 Introduction 3](#_Toc516126812)

[1.2 Objectifs 3](#_Toc516126813)

[1.3 Planification initiale 3](#_Toc516126814)

[2 Analyse / Conception 4](#_Toc516126815)

[2.1 Concept 4](#_Toc516126816)

[2.1.1 Vue d’ensemble 4](#_Toc516126817)

[2.1.2 MCD 5](#_Toc516126818)

[2.2 Stratégie de test 5](#_Toc516126819)

[2.3 Risques techniques 5](#_Toc516126820)

[2.4 Planification 6](#_Toc516126821)

[2.5 Dossier de conception 6](#_Toc516126822)

[2.5.1 Maquettes / Use cases / Scénarios 7](#_Toc516126823)

[7](#_Toc516126824)

[2.5.2 MLD 16](#_Toc516126825)

[2.5.3 (Particularité 1) 16](#_Toc516126826)

[2.5.4 (Particularité 2) 16](#_Toc516126827)

[3 Réalisation 17](#_Toc516126828)

[3.1 Dossier de réalisation 17](#_Toc516126829)

[3.2 Description des tests effectués 18](#_Toc516126830)

[3.3 Erreurs restantes 18](#_Toc516126831)

[3.4 Documents fournis 19](#_Toc516126832)

[4 Conclusions 19](#_Toc516126833)

[4.1 Objectifs non-atteints 19](#_Toc516126834)

[4.2 Objectifs atteints 19](#_Toc516126835)

[4.3 Points positifs 19](#_Toc516126836)

[4.4 Points négatifs 19](#_Toc516126837)

[4.5 Difficultés particulières 19](#_Toc516126838)

[4.6 Suites possibles pour le projet (évolutions & améliorations) 20](#_Toc516126839)

[5 Annexes 20](#_Toc516126840)

[5.1 Journal de bord 20](#_Toc516126841)

[5.2 Manuel d’installation 20](#_Toc516126842)

[5.3 Manuel d’utilisation 21](#_Toc516126843)

[5.4 Sources – Bibliographie 21](#_Toc516126844)

# Analyse préliminaire

## Introduction

Le but de ce projet est de réaliser un jeu sous IOS. Il s’agit d’un jeu à but pédagogique permettant aux joueurs d’apprendre le vocabulaire d’une langue étrangère. Il est basé sur le célèbre jeu d’arcade *Space Invaders*. Le projet me permettra d’approfondir mes connaissances en développement *Swift* et pourra m’être utile dans le futur et dans la vie professionnelle. Comme préparation pour ce projet j’ai eu à développer un *Tetris* lors du cours de préparation de TPI.

## Objectifs

* Le joueur peut sélectionner la langue pour le professeur et la langue pour l’élève.
* Le professeur peut se déplacer latéralement et s’arrêter à tout moment.
* Les élèves se déplacent latéralement et descendent tous ensemble quand l’un deux touchent l’un des bords.
* Un élève part en pause dès qu’il a 3 mots corrects.
* Si la mauvaise traduction touche l’élève, il avance.
* Si le professeur a réussi à envoyer tous les élèves en pause, un écran « Congratulation » avec un score apparaît.
* Si l’élève atteint le professeur, un écran « Game Over » s’affiche.

## Planification initiale

Pour ce projet la méthode agile m’a été imposée par mon chef de projet.

Tout détail peut être trouvé dans le *Trello*.

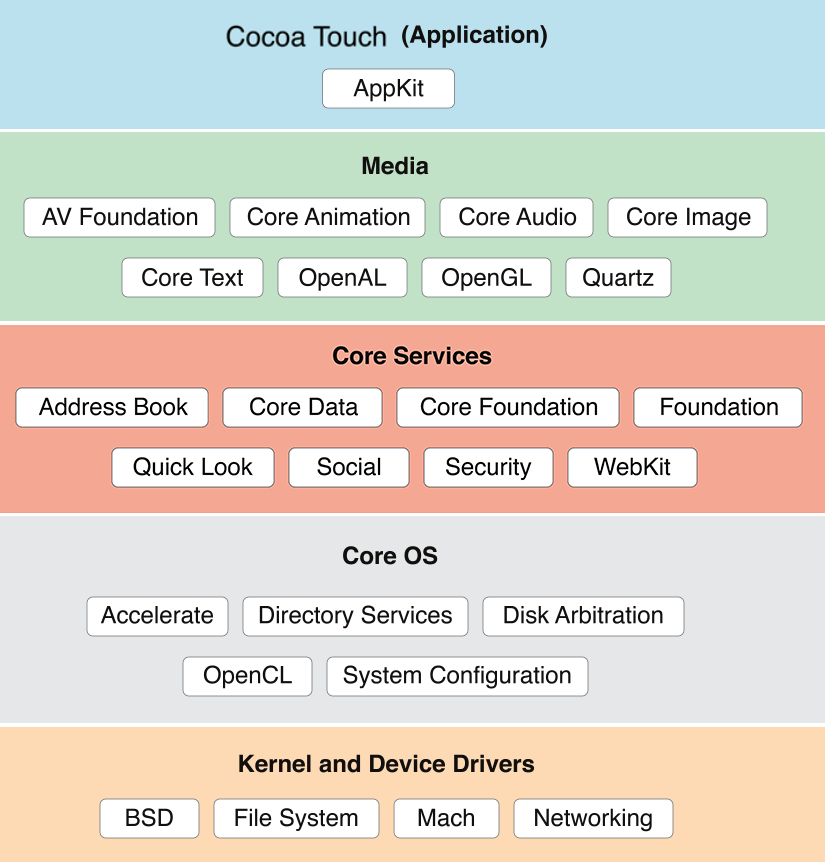
|  |  |  |
| --- | --- | --- |
|  | **Analyse** | 15 mai 2018 |
| Sprint 1 | Création des « sprites » | 17 mai 2018 |
| Sprint 2 | Création des déplacements et animations prof + élèves | 23 mai 2018 |
| Sprint 3 | Création du « gameplay » | 29 mai 2018 |
| Sprint 4 | Création de la gestion du déroulement d’une partie | 1 juin 2018 |
| Sprint 5 | Intégration du web service | 5 juin 2018 |
|  | Livraison | 7 juin 2018 |

# Analyse / Conception

## Concept

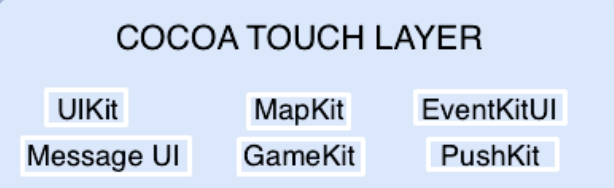
### Vue d’ensemble

Voici comment se compose l’architecture IOS :



Notre jeu se trouvera dans la partie « Application », *Cocoa Touch* est un *UI framework* développé par Apple pour créer des programmes sur IOS.

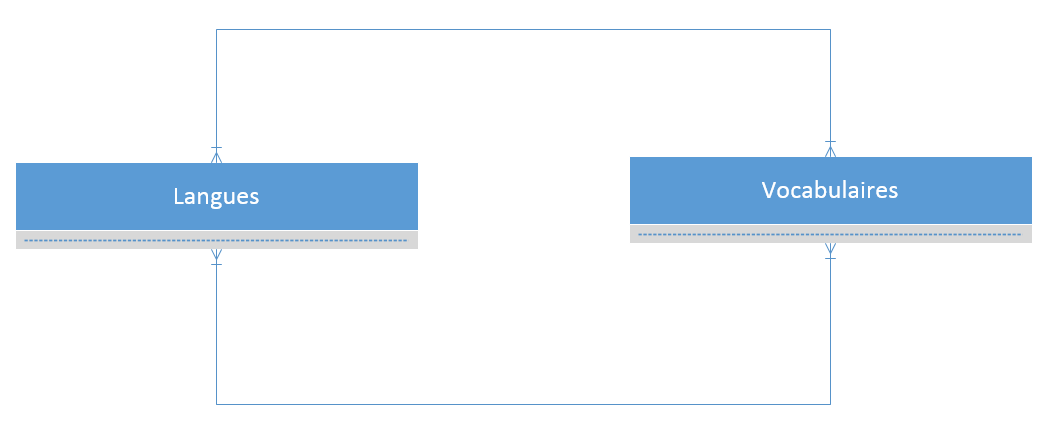
L’architecture de la couche Cocoa Touch :



*Cocoa Touch* est lui-même composé de plusieurs *framework.* Cela permet de gérer les boucles et les cycles du jeu ainsi que l’interface.

* UIKit : Construit et permet de gérer une interface utilisateur graphique pilotée par les événements pour l’application iOS ou tvOS.
* MapKit : Affiche la carte ou l'imagerie satellite directement depuis l'interface de l’application, appelle les points d'intérêt et détermine les informations de repère pour les coordonnées de la carte.
* EventKitUI : Affiche une interface pour voir, sélectionner et modifier les événements et les rappels du calendrier.
* MessageUI : Crée une interface utilisateur pour composer des e-mails et des messages texte afin que les utilisateurs puissent modifier et envoyer des messages sans quitter l’application.
* GameKit : Crée des expériences qui permettent aux joueurs de revenir dans le jeu. Ajoute des classements, des succès, des matchmaking, des défis et plus encore.
* Crée des notifications push pour l’application

### MCD



Le MCD ne concerne que les vocabulaires et les langues. Chaque vocabulaire appartient à deux langues et chaque langue à un ou plusieurs vocabulaires.

Le MCD ne concerne que les vocabulaires et les langues.

## Stratégie de test

Les tests seront effectués sur l’IPad qui m’a été fourni pour ce TPI. Je testerai le jeu moi-même et le chef de projet participera à plusieurs tests qui seront effectués lors de la démo des différents *Sprint*. Ce sera des tests fonctionnels qui seront effectués à l’aide des différents scénarios.

## Risques techniques

N’ayant suivi aucune formation pour développer ce projet, certains risques sont présents au niveau des compétences mais des mesures ont été prises afin de réduire ces risques. J’ai pu effectuer un projet de préparation au TPI qui était de développer un *Tetris* en *Swift*.

Nous avons aussi priorisé le développement du jeu, c’est-à-dire que les vocabulaires seront au départ « hardcodé » et à la fin du développement du jeu, le web service fourni par notre chef de projet sera implémenter si le jeu est parfaitement fonctionnel.

## Planification

|  |  |  |  |
| --- | --- | --- | --- |
| **Sprint** | **Date prévue** | **Résultat** | **Date effective** |
| Sprint 1 | 17.05 | Réalisé avec succès | 17.05 |
| Sprint 2 | 23.05 | Réalisé avec succès | 29.05 |
| Sprint 3 | 29.05 | Réalisé avec succès | 01.06 |
| Sprint 4 | 01.06 | Réalisé avec succès | 07.06 |
| Sprint 5 | 05.06 | Non réalisé | Pas atteint |
| Livraison | 07.06 | Réalisé avec succès | 07.06 |

## Dossier de conception

J’ai choisi un MacBook Pro car j’en possède un et car il est plus facile de développer celui-ci que sur les mini Mac de l’école professionnelle, qui sont moins puissants et qui risquaient d’être ralentis lors des simulations.

Le système d’exploitation et celui que j’ai eu lors de l’achat de ce MacBook, il supporte tous les outils qui sont utilisés pour ce projet je ne l’ai donc pas modifié.

J’ai choisi la dernière version de XCode vu que je n’ai jamais développé sur d’autres versions.

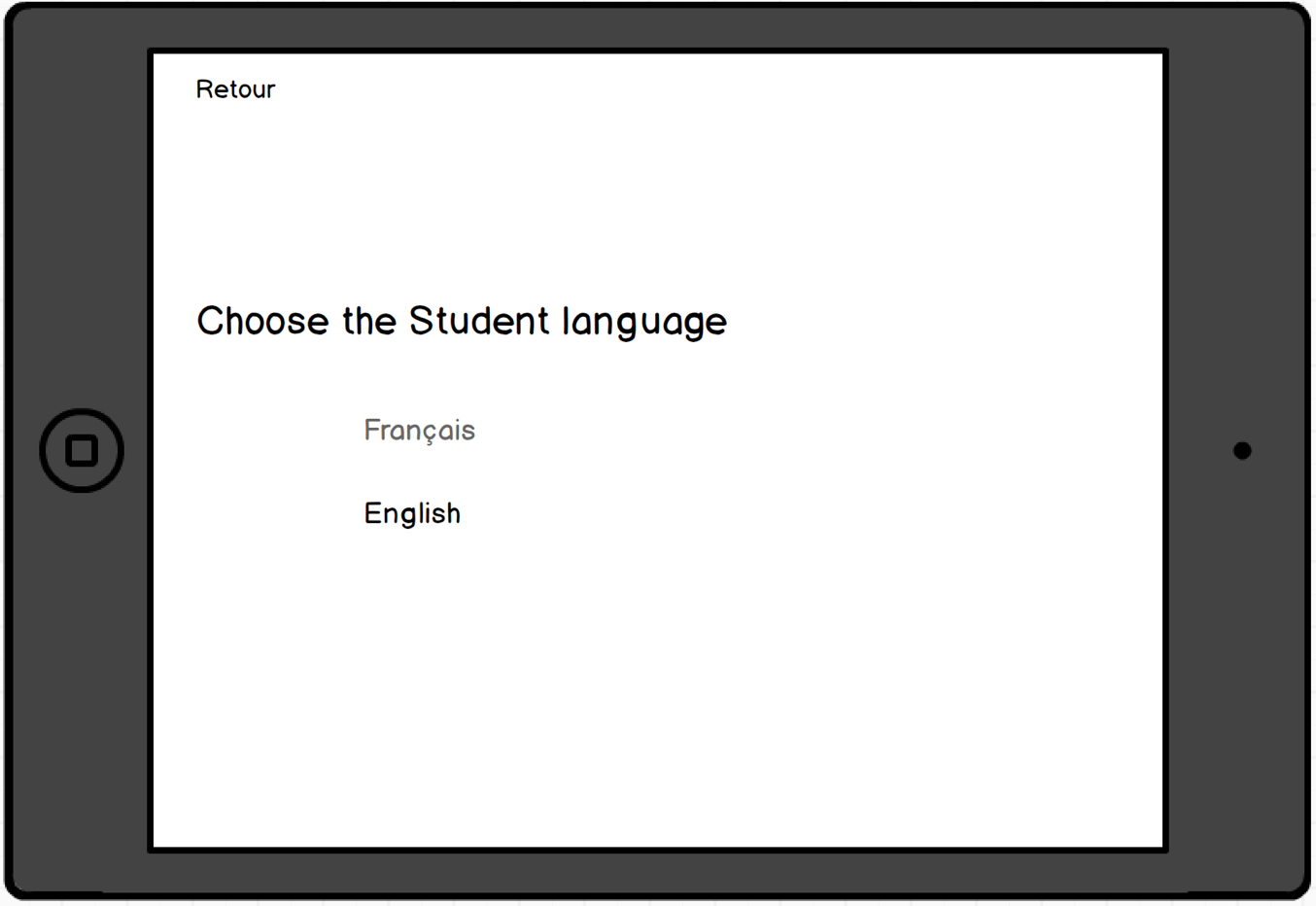
### Maquettes / Use cases / Scénarios

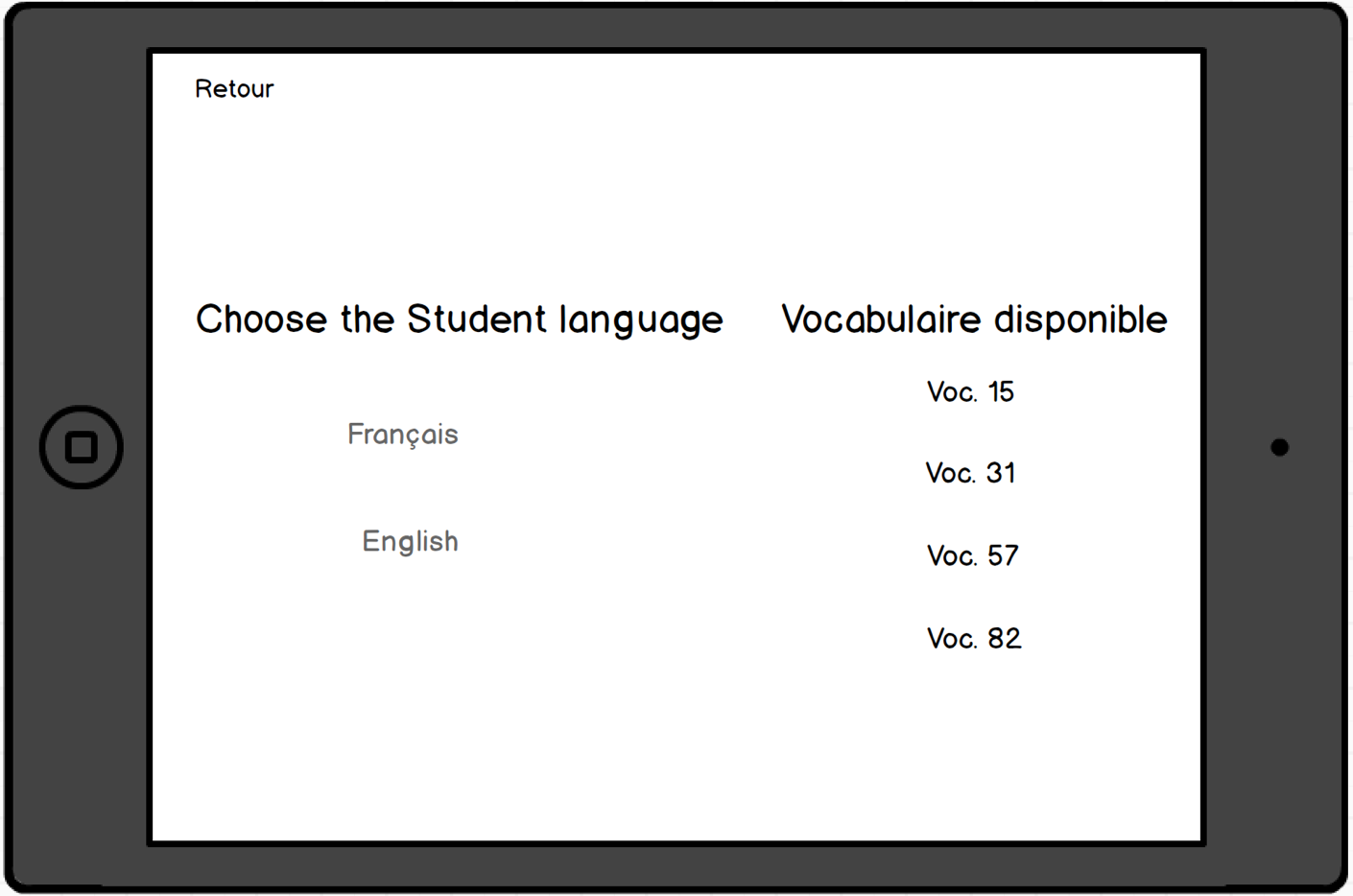
## 

Menu 0

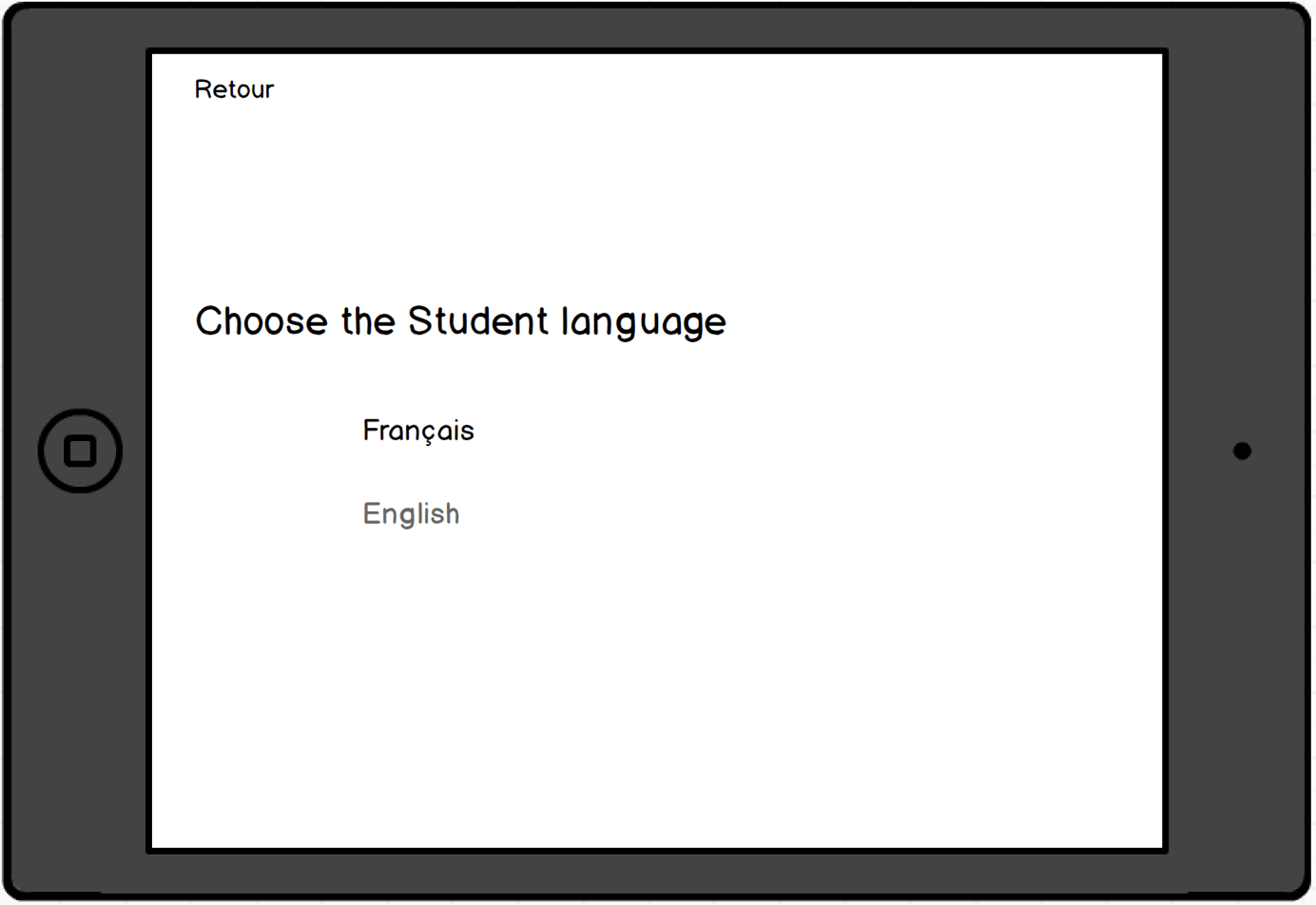
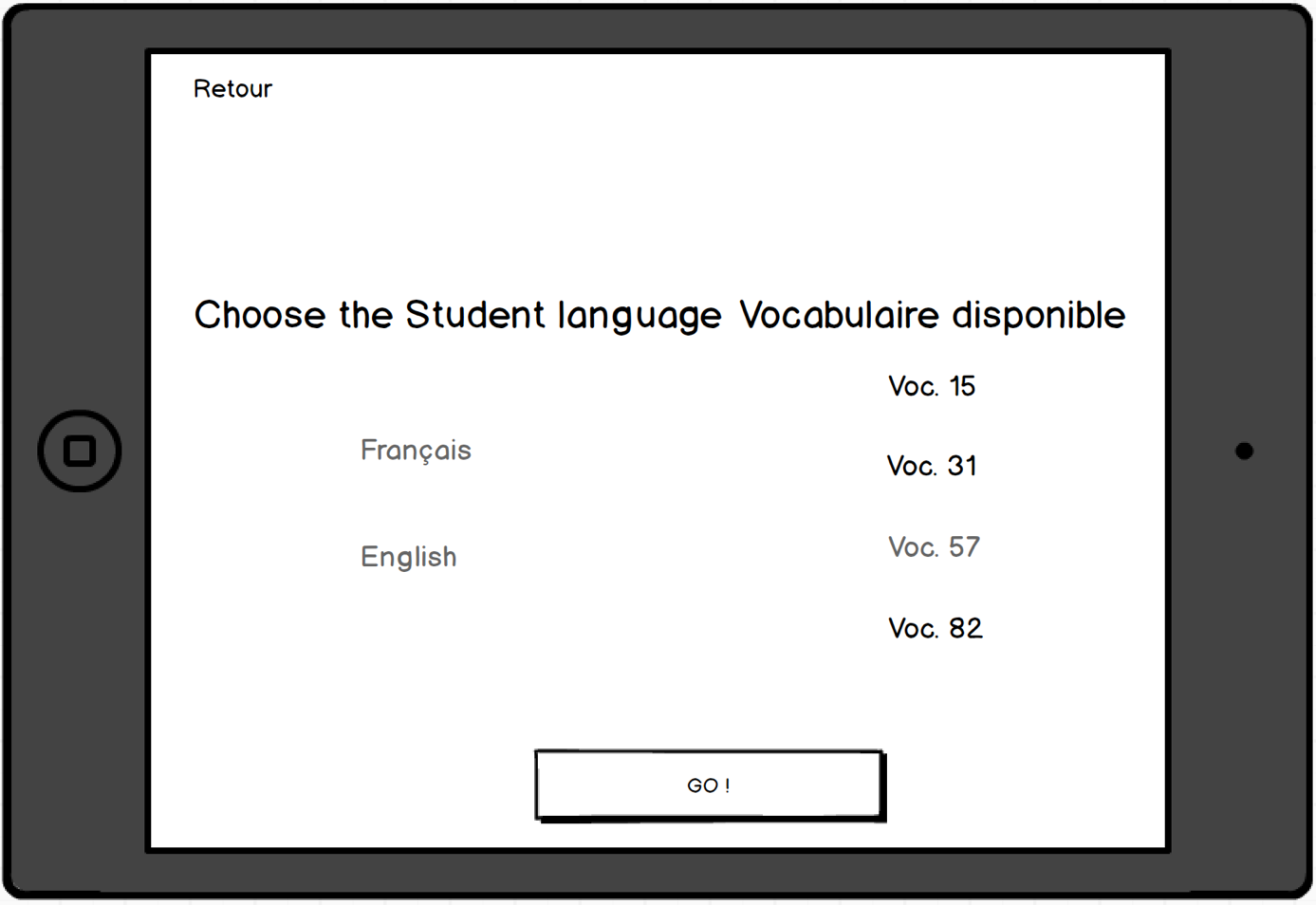
Menu 1

Menu 2



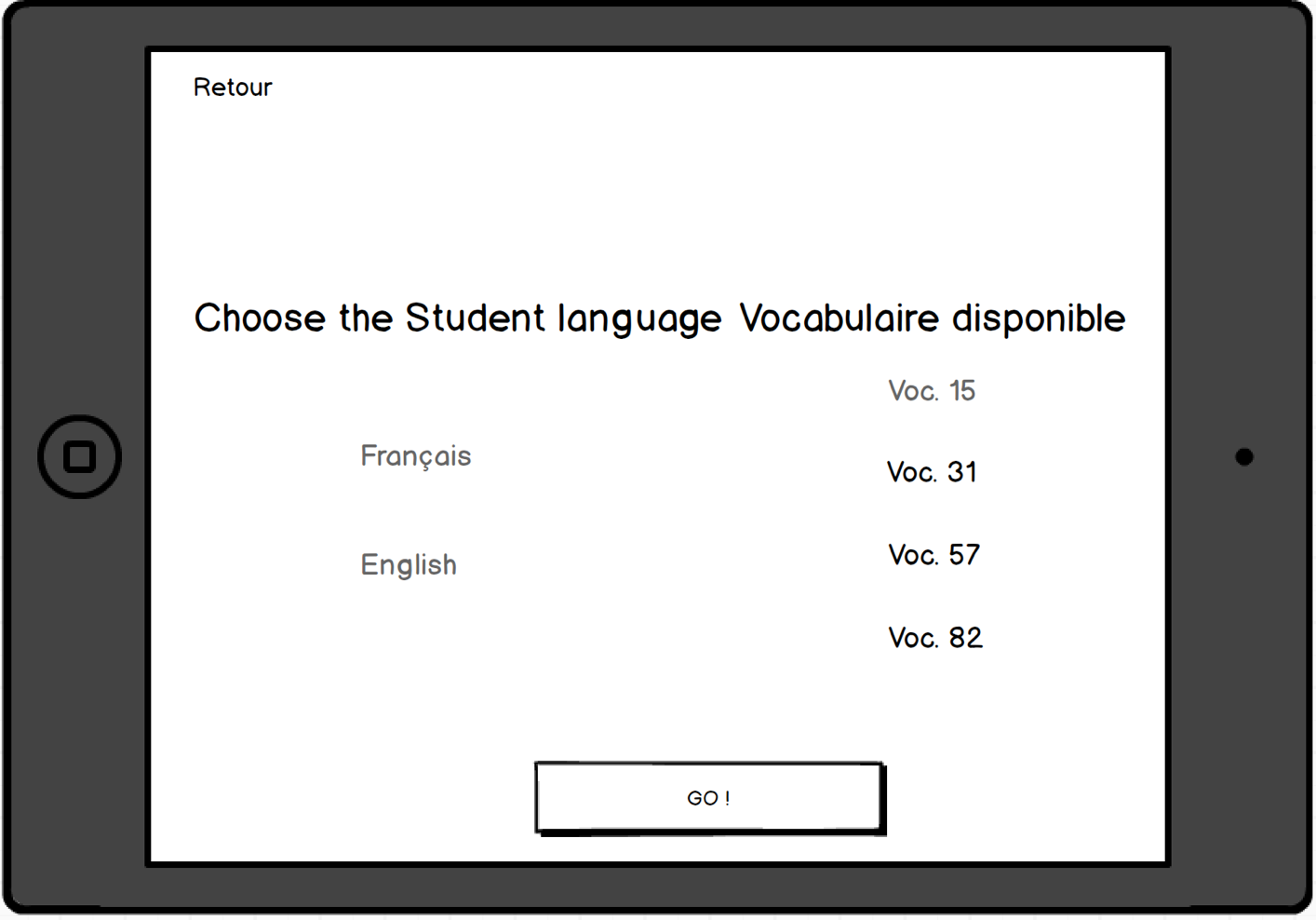


Menu 3

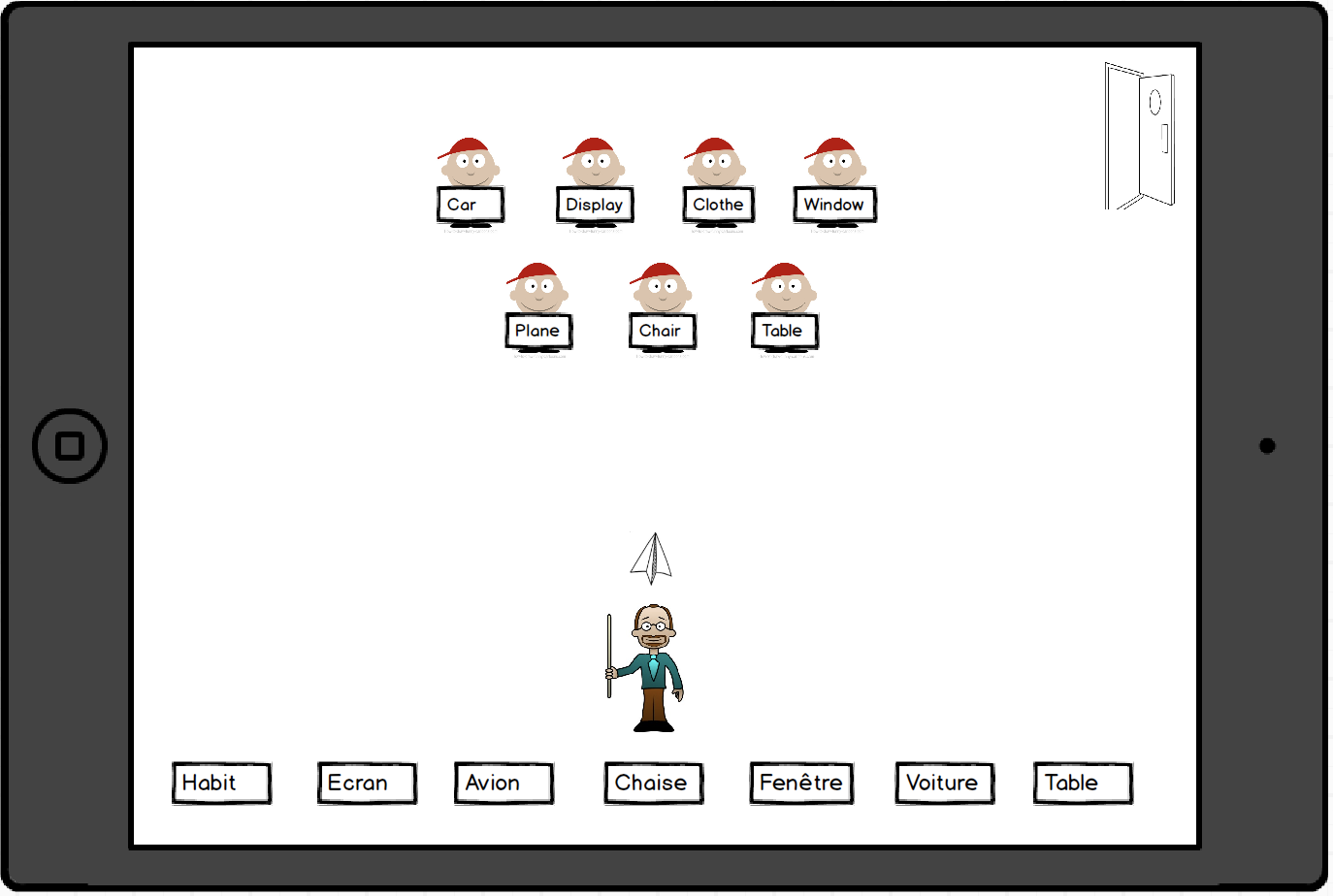


Menu 5

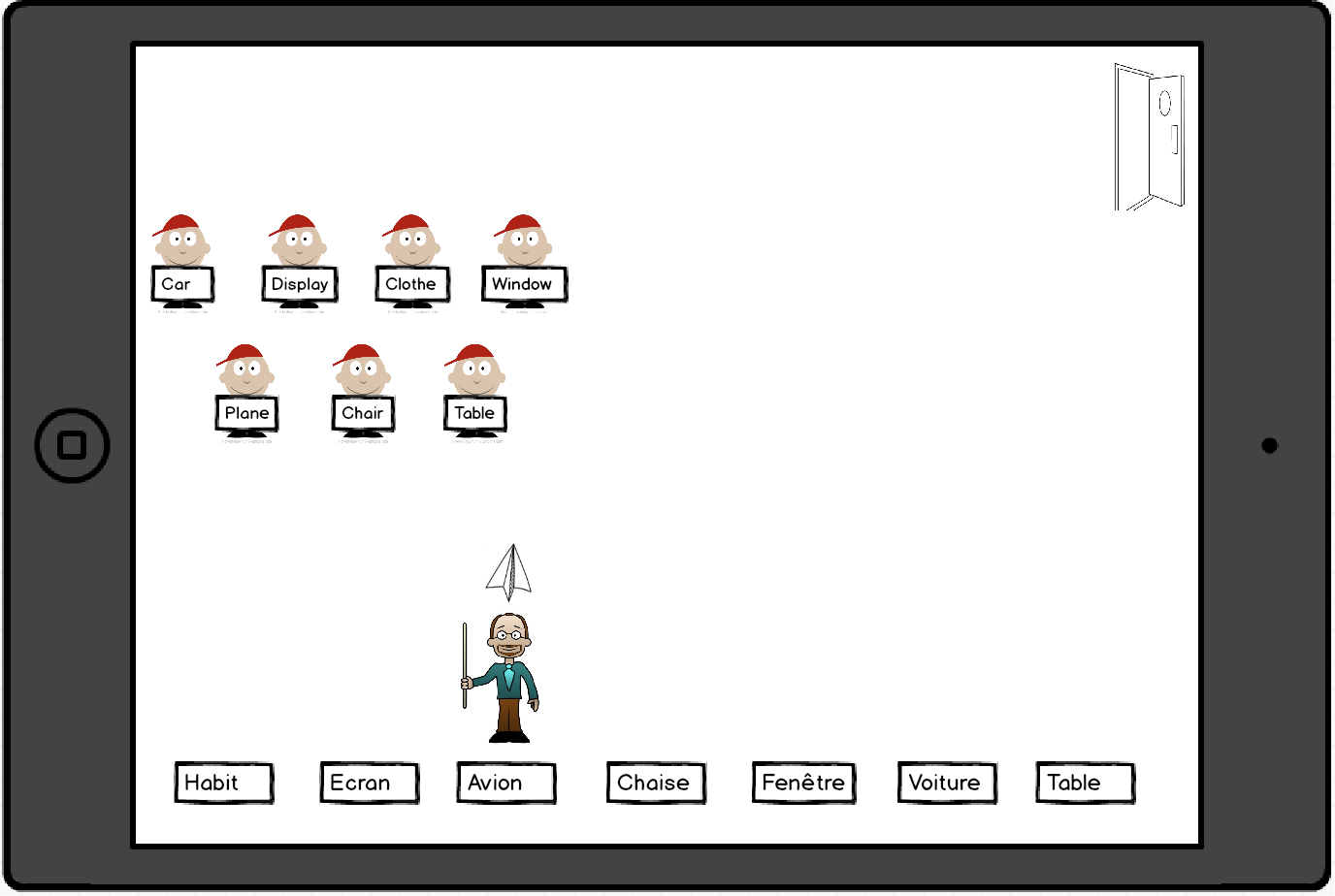
Menu 4



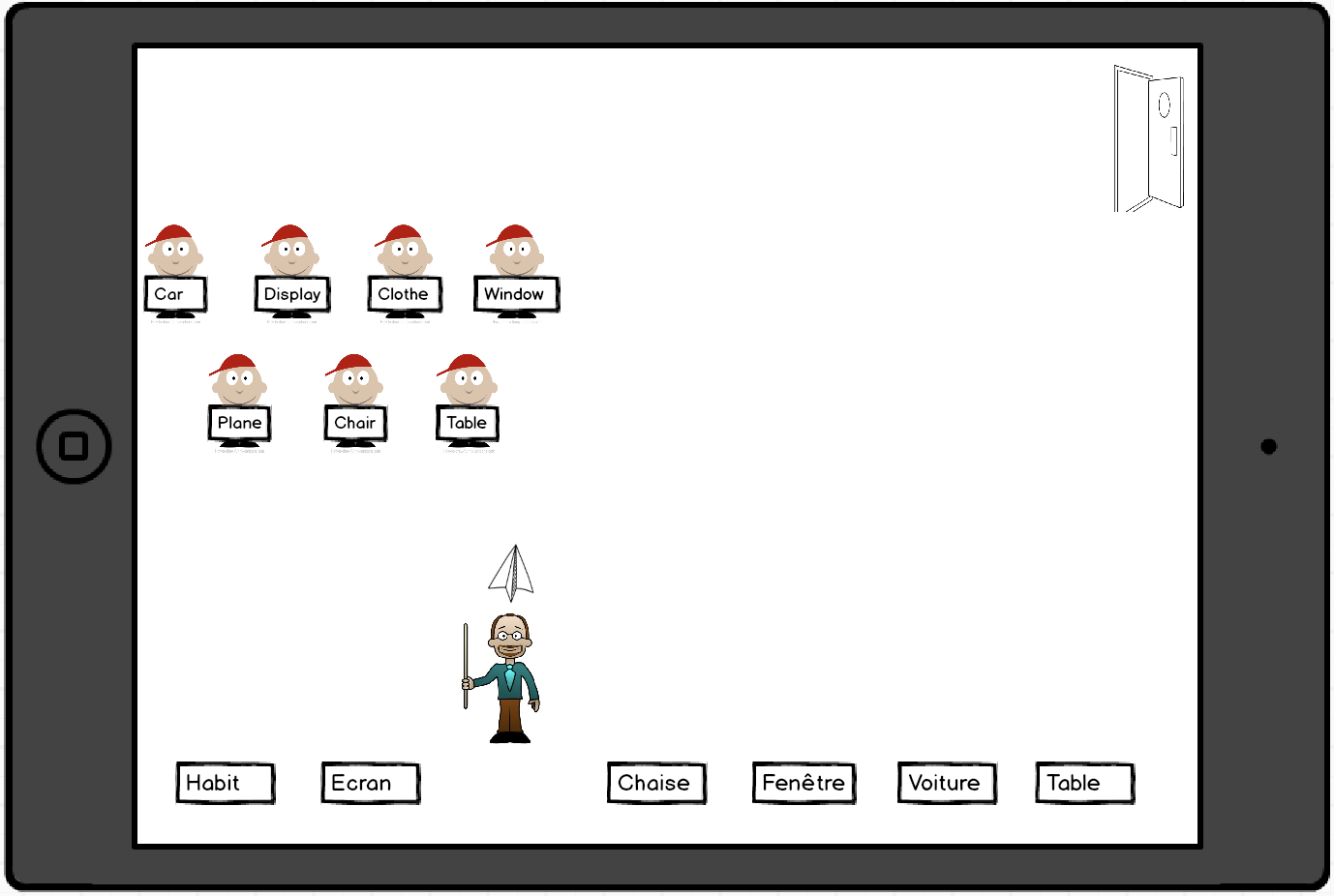
Menu 6



Game 1

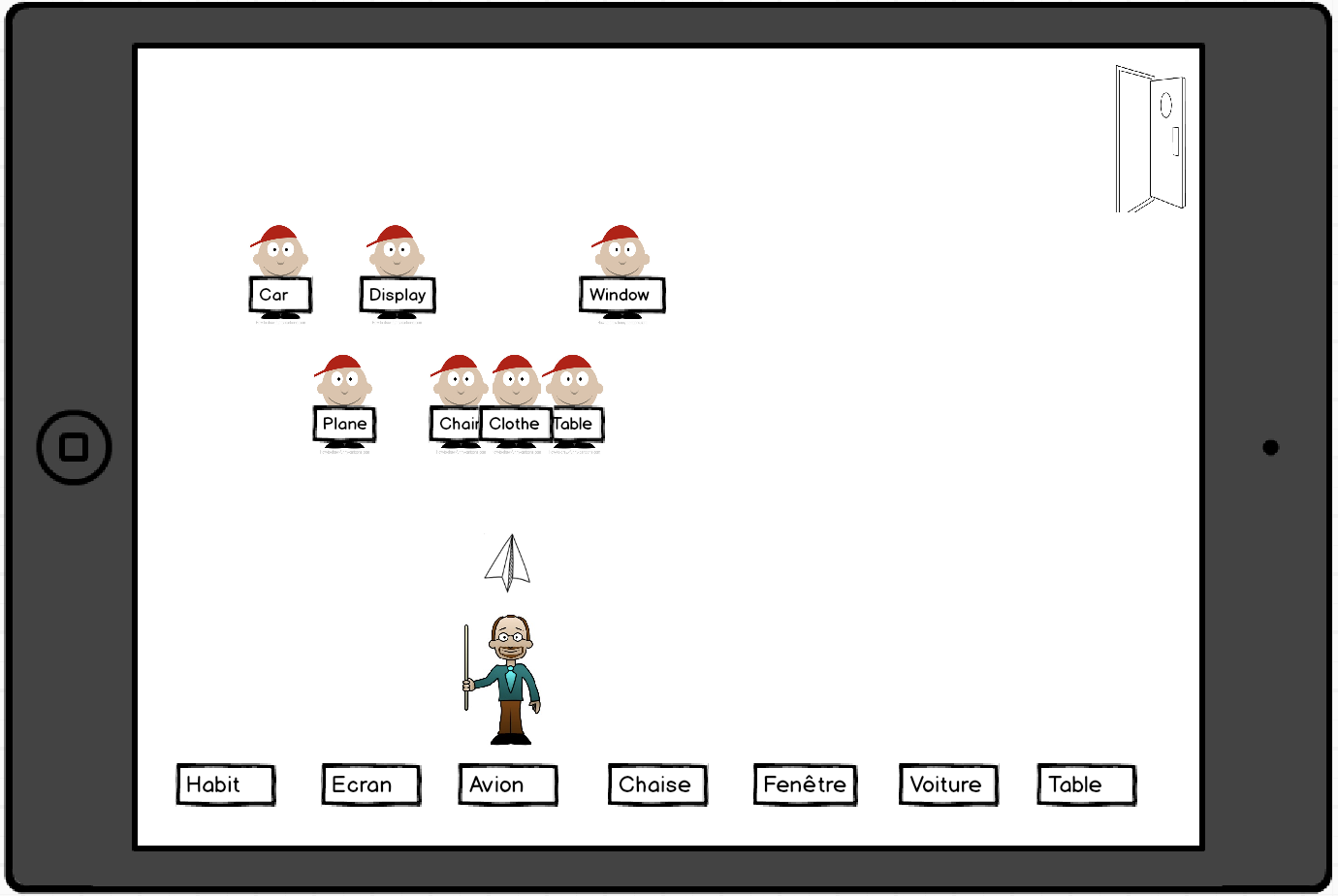


Game 2



Game 3

Game 3



Game 4

**Use Case : Lancement d’une partie**

Scénario

|  |  |  |
| --- | --- | --- |
| 1. Lancement d’une partie pour apprendre une langue | | |
| Action | Condition particulière | Réponse |
| Clique sur « Space Invaders » sur l’écran « Menu 0 ». |  | Affiche « Menu 1 » |
| Clique sur « English » dans « Menu 1 ». | Aucune langue n’est sélectionnée. | Affiche « Menu 2 » |
| Clique sur « Français ». |  | Affiche « Menu 3 » |
| Clique sur « Voc. 15 ». | Aucun « Voc » n’est sélectionné. | Affiche « Menu 6 » |
| Clique sur « GO ! ». |  | Affiche « Game 1 » |

|  |  |  |
| --- | --- | --- |
| 1. Lancement d’une partie en se trompant de langue à apprendre | | |
| Action | Condition particulière | Réponse |
| Clique sur « Space Invaders » sur l’écran « Menu 0 ». |  | Affiche « Menu 1 » |
| Clique sur « Français » dans « Menu 1 ». | Aucune langue n’est sélectionnée. | Affiche « Menu 2 » |
| Clique sur « Retour » dans « Menu 2 ». |  | Affiche « Menu 1 » |
| Clique sur « English ». |  | Affiche « Menu 4 » |
| Clique sur « Français ». |  | Affiche « Menu 3 » |
| Clique sur « Voc. 15 ». |  | Affiche « Menu 6 » |
| Clique sur « GO ! ». |  | Affiche « Game 1 » |

|  |  |  |
| --- | --- | --- |
| 1. Lancement d’une partie en se trompant de vocabulaire à apprendre | | |
| Action | Condition particulière | Réponse |
| Clique sur « Space Invaders » sur l’écran « Menu 0 ». |  | Affiche « Menu 1 » |
| Clique sur « Français » dans « Menu 1 ». | Aucune langue n’est sélectionnée. | Affiche « Menu 2 » |
| Clique sur « English ». |  | Affiche « Menu 3 » |
| Clique sur « Voc. 15 ». |  | Affiche « Menu 6 » |
| Clique sur « Retour ». |  | Affiche « Menu 3 » |
| Clique sur « Voc.57 ». |  | Affiche « Menu 5 » |
| Clique sur « GO !. |  | Affiche « Game 1 » |

**Use Case : Exercer un langage**

Scénario

|  |  |  |
| --- | --- | --- |
| 1. Déplacer le prof et l’avion en papier de gauche à droite | | |
| Action | Condition particulière | Réponse |
| Clique sur « GO ! » sur l’écran « Menu 0 ». |  | Affiche « Game 1 ». |
| Clique sur le côté gauche de l’écran à hauteur du prof. | Le prof et l’avion en papier ne sont pas au bord gauche de l’écran. | Le prof et l’avion en papier commencent à se déplacer vers la gauche. |
| Clique sur le côté droite de l’écran à hauteur du prof. |  | Le prof et l’avion en papier commencent à se déplacer vers la droite de l’écran. |

|  |  |  |
| --- | --- | --- |
| 1. Arrêter un déplacement du prof | | |
| Action | Condition particulière | Réponse |
| Clique sur « GO ! » sur l’écran « Menu 0 ». |  | Affiche « Game 1 » |
| Clique sur le côté gauche de l’écran à hauteur du prof. | Le prof et l’avion en papier ne sont pas au bord gauche de l’écran. | Le prof et l’avion en papier commencent à se déplacer vers la gauche. |
| Clique sur le milieu de l’écran à hauteur du prof. |  | Le prof et l’avion en papier s’arrêtent. |

|  |  |  |
| --- | --- | --- |
| 1. Déplacer le prof à droite (atteint le bord de l’écran) | | |
| Action | Condition particulière | Réponse |
| Clique sur « GO ! » sur l’écran « Menu 0 ». |  | Affiche « Game 1 » |
| Clique sur le côté droite de l’écran à hauteur du prof. | Le prof et l’avion en papier ne sont pas au bord droit de l’écran. | Le prof et l’avion en papier commencent à se déplacer vers la droite. |
|  | Le prof atteint le bord droit de l’écran. | Le prof et l’avion en papier s’arrêtent. |

|  |  |  |
| --- | --- | --- |
| 1. Déplacer le prof à gauche (atteint le bord de l’écran) | | |
| Action | Condition particulière | Réponse |
| Clique sur « GO ! » sur l’écran « Menu 0 ». |  | Affiche « Game 1 » |
| Clique sur le côté gauche de l’écran à hauteur du prof. | Le prof et l’avion en papier ne sont pas au bord gauche de l’écran. | Le prof et l’avion en papier commencent à se déplacer vers la gauche. |
|  | Le prof atteint le bord gauche de l’écran. | Le prof et l’avion en papier s’arrêtent. |

|  |  |  |
| --- | --- | --- |
| 1. Choisir un mot et lancer l’avion en papier | | |
| Action | Condition particulière | Réponse |
| Clique sur « GO ! » sur l’écran « Menu 0 ». |  | Affiche « Game 1 » |
| Clique à gauche de l’écran à hauteur du prof. | Le prof n’est pas au bord de l’écran gauche. | Le prof commence à se déplacer vers la gauche. |
| Clique sur le milieu de l’écran à hauteur du prof. |  | Le prof s’arrête. |
| Clique sur le bas de l’écran, sur un des mots. | Aucun mot n’est sélectionné. | Le prof se prépare à lancer l’avion. |
| Clique sur l’écran au-dessus du prof. | Aucun avion est lancé. | L’avion se lance et monte jusqu’à atteindre un élève ou le haut de l’écran. |

|  |  |  |
| --- | --- | --- |
| 9.Toucher un élève avec la bonne traduction | | |
| Action | Condition particulière | Réponse |
| Scénario 8 | | |
|  | L’avion touche un élève. | L’élève change de mot et l’avion en papier disparaît. |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| 10. Toucher un élève avec la mauvaise traduction | | |
| Action | Condition particulière | Réponse |
| Scénario 8 | | |
|  | L’avion touche un élève. | L’élève n’est pas content et s’avance d’un pas vers le prof. |

|  |  |  |
| --- | --- | --- |
| 11. Toucher l’élève avec le bon mot | | |
| Action | Condition particulière | Réponse |
| Scénario 8 et Scénario 9 - 3 fois | | |
|  |  | L’élève est content et part à la pause. |

### MLD



### (Particularité 1)

Pour de l’affichage des bons vocabulaires dans le menu il m’était impossible d’assigner un ID aux différentes langues qui s’affichaient à l’écran pour ensuite les reprendre quand le joueur clique dessus afin d’afficher les bons vocabulaires correspondants. J’ai donc utilisé la fonction « .name » sur les labels des langues qui est un « string », je devais donc à chaque tour de la boucle « for » transformer le « .name » du label précédent en valeur numérique et l’incrémenter puis, le retransformer en « string » pour pouvoir le réassigner au nouveau label. Ensuite, il m’a suffi de reprendre la valeur du « .name » du label où je cliquais et le stocker dans une variable « prof » pour la langue choisie du professeur et « élève » pour la langue choisie de l’élève pour pouvoir les réutiliser plus tard.

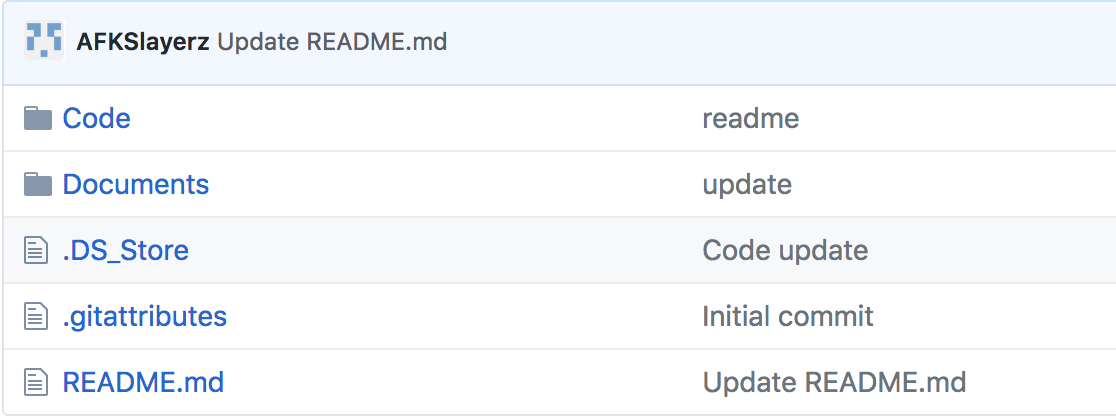
### (Particularité 2)

Pour l’affichage des élèves, j’ai décidé de créer un « groupe » d’élève qui sont tous indépendants les uns des autres mais leur déplacement est géré en groupe. Pour ce faire, je leur ai assigné un nom qui est identique pour tous et cela me permet d’utiliser ce mot pour énumérer les *Sprite* puis leur faire faire l’action que j’ai envie. Cela me permet de recréer le même déplacement qui est utilisé dans le jeu *Space Invader*.

# Réalisation

## Dossier de réalisation

Pour ce projet, j’ai utilisé Github, voici le lien d’accès à mon repository <https://github.com/AFKSlayerz/TPI>.



Le repo est composé de deux dossiers principaux et un README.md:

* Le dossier **Code** contient tous les fichiers utiles et nécessaires pour exécuter l’application.
* Le dossier **Documents** contient le cahier des charges, le rapport de TPI et le journal de travail.
* Le fichier **README.md** contient le manuel d’installation ainsi que les directives pour installer l’application et l’utiliser.

Le projet est réalisé sur un MacBook Pro sous macOS High Sierra 10.13.3 avec le logiciel XCode 9.3. Le logiciel est installé dans le dossier *application* du Mac.

Les fichiers du projet se trouvent sur le bureau qui est synchronisé avec l’ICloud et mis à jour régulièrement sur mon repo GitHub.

Aucun autres logiciels tierces ou librairies externes ne sont utilisés.

Repo Git : <https://github.com/AFKSlayerz/TPI>

## Description des tests effectués

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Scénario* | *17.5 CdP* | *29.5 CdP* | *06.06 Dév* | *07.06 CdP* |
| *Scénario 1* | *OK* | *OK* | *OK* |  |
| *Scénario 2* | *OK* | *NOK* | *NOK* |  |
| *Scénario 3* | *OK* | *OK* | *OK* |  |
| *Scénario 4* |  | *OK* | *OK* |  |
| *Scénario 5* |  | *OK* | *OK* |  |
| *Scénario 6* |  | *OK* | *OK* |  |
| *Scénario 7* |  | *OK* | *OK* |  |
| *Scénario 8* |  |  | *OK* |  |
| *Scénario 9* |  |  | *OK* |  |
| *Scénario 10* |  |  | *OK* |  |
| *Scénario 11* |  |  | *OK* |  |

## Erreurs restantes

**Menu** :

* Impossible d’utiliser le bouton « retour », l’application se bloque quand on appuie sur le bouton « retour » pour choisir une autre langue, mais cela fonctionne pour choisir un autre vocabulaire.

**Solution** : Réécrire une partie du code afin de pouvoir mieux gérer les différentes variables qui sont incrémentées au fur et à mesure des langues sélectionnées.

**Game** :

* Quand un mot est déjà sélectionné et que l’on en sélectionne un autre, le jeu crash.
* Si l’avion est lancé et que l’on sélectionne un autre mot, le jeu crash.

**Solution** : Trouver quel code n’est pas supprimé lors du choix d’un mot et le supprimer, le jeu crash à cause de cela.

* Il n’est pas possible de sélectionner deux fois de suite le même mot d’affilée.

**Solution** : Réécrire une partie du code pour vider la variable contenant le mot précédent. Actuellement, le mot reste stocké dans une des variables et cela ne permet pas de réécrire deux fois le même mot d’affilée dedans.

* Si d’autres langues que le français et l’anglais sont sélectionnées, les élèves ne s’affichent pas.

**Solution** : Trouver et réécrire une partie du code où l’incrémentation est incorrecte pour aller rechercher les mots des élèves.

* Si l’on touche un élève à un certain moment au bord de l’écran avec le bon mot, cela va créer un bug où les pancartes vont se décaler avec les élèves.

**Solution** : Mieux gérer les hitbox afin que les élèves et les pancartes aient exactement les mêmes.

## Documents fournis

* Le rapport de projet
* Le manuel d'Installation (en annexe)
* Le journal de travail (en annexe)
* Le résumé du rapport de projet (en annexe)
* Une archive contenant le code source par e-mail

# Conclusions

## Objectifs non-atteints

Plusieurs objectifs n’ont pas été atteints et cela nuit à la jouabilité du jeu. Comme pour les élèves qui doivent être touché 3x d’affilée avant de pouvoir aller en pause ou l’implémentation du Web Service qui permet d’avoir plus que quelques élèves et quelques mots sur le terrain de jeu.

## Objectifs atteints

Les objectifs du point de vue fonctionnel sont présents malgré plusieurs bug qui n’ont pas été corrigés. Le joueur peu sélectionner une langue pour l’élève et une pour le professeur et choisir un des vocabulaires proposés. Les vocabulaires sont gérés de la bonne manière pour que les mots du professeur et de l’élève correspondent aux données fournies par notre chef de projet. Il est possible de toucher les élèves et de les faire aller en pause quand ils sont touchés une fois par le mot correspondant dans l’autre langue et de les faire s’avancer quand celui-ci ne correspond pas. Quand il n’y a plus d’élèves restants, la partie se termine, tout comme quand un élève touche le professeur.

## Points positifs

J’ai beaucoup aimé réaliser ce projet hormis le fait que je n’avais que très peu de connaissances dans le langage que j’ai utilisé. Je pense continuer ce projet par la suite pendant mon temps libre ou m’en inspirer pour effectuer un autre jeu du même style.

## Points négatifs

Un des points que je trouve négatif, c’est de ne pas avoir réussi à finaliser le jeu dû au manque de connaissances. Certains bug sont présents et peuvent nuire à l’expérience de jeu du joueur.

## Difficultés particulières

Plusieurs difficultés ont été rencontrées pendant le projet mais celle qui m’ont fait perdre le plus de temps étaient dans la majorité des cas les plus simple à résoudre avec des connaissances en Swift, comme pour des fonctions spécifiques à Swift, qui facilite beaucoup le travail ou des manières d’écrire le code qui elles aussi simplifient et évitent beaucoup de problème par la suite quand elles sont connues.

## Suites possibles pour le projet (évolutions & améliorations)

Faire une compatibilité sur iPhone pour que le jeu soit parfaitement jouable dessus. Quand le projet sera fini, il peut être possible d’ajouter de nouvelles langues ou de nouveaux vocabulaires. De plus, il peut être ajouté des niveaux de difficultés avec des mots plus complexes et plus d’élèves sur le terrain de jeu.

Il serait aussi possible de faire cela avec des verbes et des temps comme par exemple en anglais, le professeur pourrait choisir un temps de verbe et il devrait tirer sur le verbe ayant ce temps-ci. Cela permettrait d’apprendre les temps des verbes et les verbes irréguliers.

# Annexes

## Journal de bord

|  |  |
| --- | --- |
| **Date** | **Événement** |
| 8.05 | Rencontre du chef de projet 2 fois dans la journée pour des clarifications sur le fonctionnement du jeu ainsi que pour la validation de la planification initiale. |
| 18.05 | Rendez-vous avec le chef de projet pour le Sprint 1 -> Sprint validé |
| 23.05 | Deuxième rendez-vous avec un des experts |
| 24.05 | Discussion avec le chef de projet : Le bouton « retour » du menu ainsi que les langues qui se grisent, après avoir cliqué dessus et passent en deuxième priorité. Priorité n°1 -> s’attaquer au vif du jeu. |
| 25.05 | Aide externe -> Aide de Jarod, pour le fonctionnement logique du code. |
| 29.05 | Rendez-vous avec le chef de projet pour le Sprint 2 -> Sprint validé. Discussion sur les objectifs à faire pour le Sprint 3. |
| 01.06 | Rendez-vous avec le chef de projet pour le Sprint 3 -> Sprint validé, discussion sur les objectifs à faire pour le dernier Sprint. |
| 07.06 | Rendez-vous avec le chef de projet pour le dernier sprint -> Sprint validé |

## Manuel d’installation

1. Téléchargez le projet depuis le repository Git
2. Dans le dossier /TPI-master/Code/, exécutez "School Invaders.xcodeproj"
3. Choisissez votre compte dévloppeur Apple dans Xcode
4. Choisissez la device sur laquelle vous voulez le jeu
5. Lancez le build

Une fois ces étapes réalisées il vous faut aller dans les paramètres généraux de votre device sous "Gestion des appareils" et autorisez le compte dévloppeur.

## Manuel d’utilisation

Quand le jeu est lancé, il vous demande de choisir « Teacher langage » à ce moment-ci choisissez une langue (prenez français pour éviter tout bug).

Après choisissez une langue pour le « Student langage ». Puis choississez un des vocabulaires proposer et pour finir appuyé sur « GO ! ».

Quand le jeu est lancé vous pouvez déplacement le professeur latéralement en cliquant a droit ou à gauche de l’écran à la hauteur du professeur et en appuyant au milieu il s’arrête.

Pour choisir un mot il faut se trouver sur la case contenant ce moment, et il suffit de cliquer dessus pour le sélectionner.

Pour lancer l’avion il faut appuyer n’importe où sur le haut de l’écran.

Pour finir une partie il faut avoir fait aller en pause tous les élèves ou qu’un des élèves touche le professeur.

## Sources – Bibliographie

<https://code.tutsplus.com/tutorials/create-space-invaders-with-swift-and-sprite-kit-implementing-gameplay--cms-23372>

<https://martinmitrevski.com/2016/10/12/swift-class-diagrams-and-more/>

<https://stackoverflow.com/questions/43231158/swift-ios-how-to-hide-label-then-make-it-appear-after-a-certain-time-period>

<https://docs.swift.org/swift-book/LanguageGuide/Functions.html>

<https://developer.apple.com/documentation/spritekit/skscene?changes=_4>

<https://docs.swift.org/swift-book/ReferenceManual/Statements.html>

<https://docs.swift.org/swift-book/ReferenceManual/Types.html>

<https://docs.swift.org/swift-book/ReferenceManual/Declarations.html>

<https://stackoverflow.com/questions/37258081/adding-a-sprite-to-gamescene-from-another-class>

<https://stackoverflow.com/questions/24115141/converting-string-to-int-with-swift>

<https://www.ioscreator.com/tutorials/sprite-movement-actions-sprite-kit-ios-tutorial-ios10>

<https://stackoverflow.com/questions/34382979/how-to-make-a-sprite-move-up-and-down-forever-in-swift-spritekit/34383154>

<https://www.hackingwithswift.com/example-code/arrays/how-to-loop-through-items-in-an-array>

<https://docs.swift.org/swift-book/LanguageGuide/CollectionTypes.html>

<https://docs.swift.org/swift-book/LanguageGuide/Enumerations.html>

<https://code.tutsplus.com/tutorials/create-space-invaders-with-swift-and-sprite-kit-finishing-gameplay--cms-23381>

<https://www.hackingwithswift.com/example-code/system/how-to-copy-objects-in-swift-using-copy>

<https://www.youtube.com/watch?v=miQ4YmPtFJE>

<https://stackoverflow.com/questions/26438108/ios-swift-didbegincontact-not-being-called>

<https://developer.apple.com/documentation/spritekit/skaction/1417768-move>

<https://www.dotnettricks.com/learn/xamarin/understanding-xamarin-ios-build-native-ios-app>

<https://www.dotnettricks.com/learn/xamarin/understanding-xamarin-ios-build-native-ios-app>