



---

# DOSSIER DE SPÉCIFICATIONS

---

## PROJET TUTEURÉ S2

---

Pierre RANNOU, Alban CORNO, Samuel THIKEN,  
Orane ORVEILLON, Yoann LE GLÉAU

# Sommaire

---

<b>Introduction</b>	<b>4</b>
Objectif du document	4
Domaine d'application	4
Description du projet	4
<b>Description du métier</b>	<b>5</b>
Les acteurs	5
Les concepts	5
Créer ou modifier un exercice	5
Effectuer un exercice	5
Les processus	5
<b>Spécifications fonctionnelles détaillées</b>	<b>6</b>
Diagrammes de classe	6
Enseignant	6
Elève	7
Diagrammes de cas d'utilisation	8
Enseignant	8
Elève	9
<b>Spécifications IHM</b>	<b>10</b>
Arbres des tâches	10
Réaliser un exercice (élève)	10
Créer un exercice (enseignant)	11
Maquettes	12
Enseignant	12
Interface principale	12
Interface du menu fichier	13
Interface de l'aide	13
Interface à propos	14
Interface Etape 1	14
Interface Etape 2	15
Interface Etape 3	16
Interface Etape 4	17

<b>Algorithmes</b>	<b>18</b>
Enseignant	18
Créer un exercice	18
Associer une ressource à un exercice	18
Enregistrer un exercice	19
Modifier un exercice	19
Etudiant	20
Ouvrir un exercice	20
Voir l'aide	20
Enregistrer l'exercice	20
cryptage	22
vérification	22
 <b>Glossaire</b>	 <b>24</b>
Diagramme de classe	24
Diagramme de cas d'utilisation	24
IHM	24



# Introduction

---

## Objectif du document

Ce document a pour objectif de présenter et de définir les spécifications fonctionnelles détaillées de l'application "Reconstitution". Les spécifications fonctionnelles englobent l'ensemble des fonctionnalités de l'application, leurs relations au sein du système ainsi que les spécificités de l'interface homme-machine.

Dans le cadre de notre projet tuteuré du deuxième semestre, l'IUT de Laval nous a demandés de recréer une nouvelle version de cette application qui était utilisée par les départements MMI et TC qui soit compatible avec les systèmes d'exploitation actuels.

## Domaine d'application

Ce présent dossier de spécifications fonctionnelles détaillées est applicable pendant le phase de développement de l'application "Reconstitution".

## Description du projet

Le but de cette application est de permettre aux enseignants de langues de créer des exercices pour leurs étudiants. Ces exercices porteront sur la compréhension orale de vidéos ou de pistes audio. L'étudiant devra écrire ce qu'il entend pour dévoiler le texte original.

# Description du métier

---

## Les acteurs

Il doit y avoir deux différentes versions de l'application. Une version pour les enseignants leur permettant de créer et modifier des exercices et une version pour les étudiants leur permettant de les effectuer.

## Les concepts

### Créer ou modifier un exercice

~~L'enseignant choisit la ressource audio ou vidéo voulue pour son exercice dans son explorateur de fichiers. Il doit ensuite écrire ou coller le script correspondant à ce qui est dit dans la piste. Plusieurs options sont disponibles pour personnaliser l'exercice comme la possibilité d'afficher le temps restant pour entrer les réponses ou encore la sensibilité à la casse du texte. Une fois la création ou la modification de l'exercice terminée, il est enregistré sur l'appareil.~~

### Effectuer un exercice

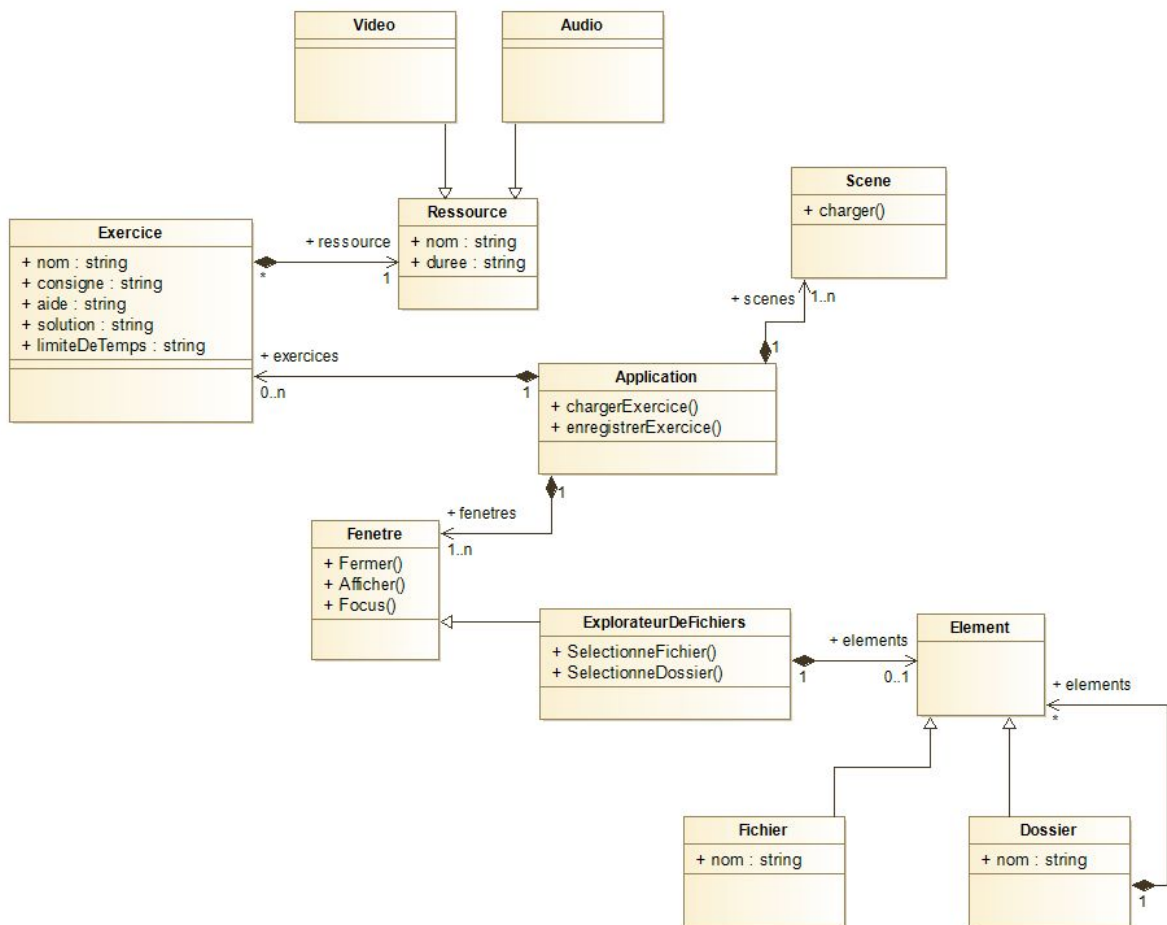
~~L'étudiant charge l'exercice à faire à partir de son explorateur de fichiers. Il dispose d'un texte occulté et d'une piste qu'il doit écouter, avec la possibilité de faire des pauses et de revenir en arrière. Lorsqu'il reconnaît un mot, il entre au clavier ce qu'il entend. Si le mot est correct, toutes ses occurrences sont révélées dans le texte. Une fois l'exercice terminé, il est enregistré sur l'appareil.~~

[les classes (exercice)]

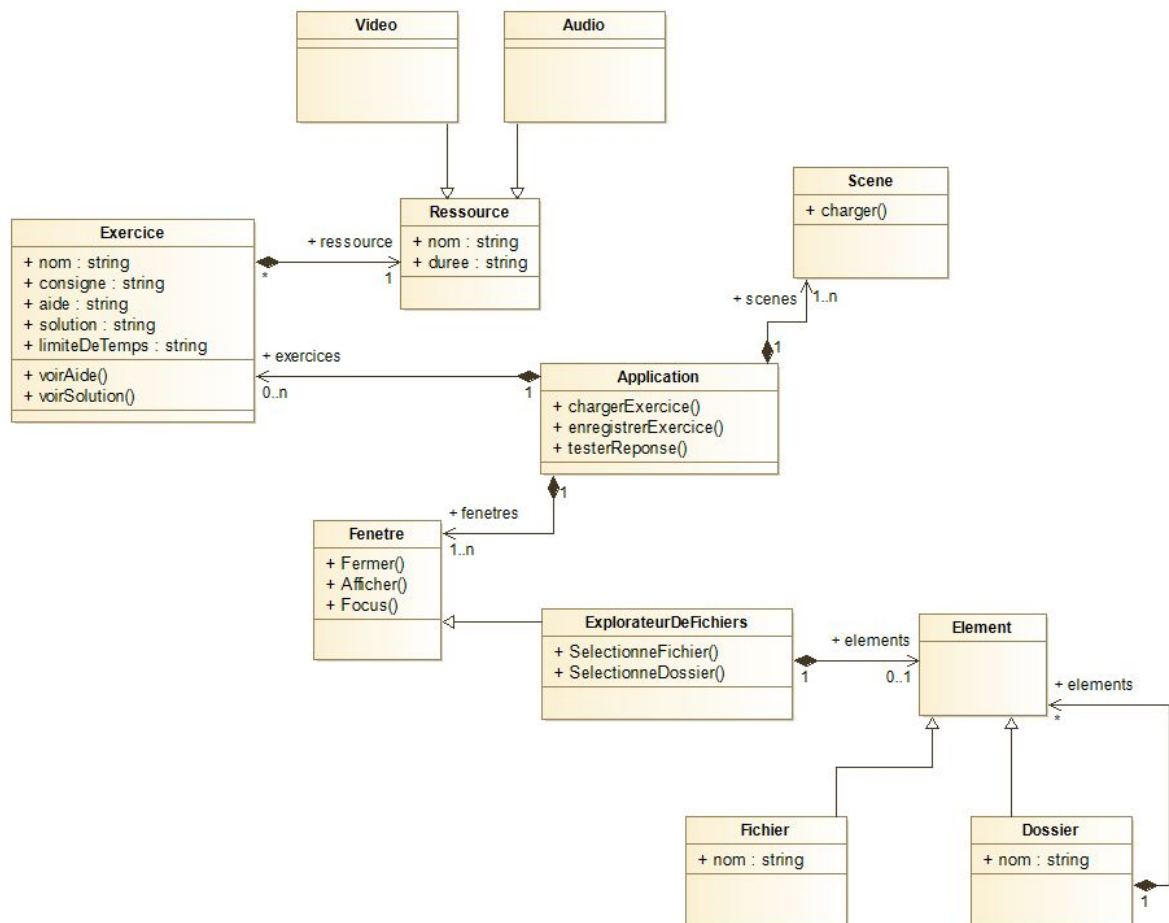
# Spécifications fonctionnelles détaillées

## Diagrammes de classe [dans une autre partie]

### Enseignant

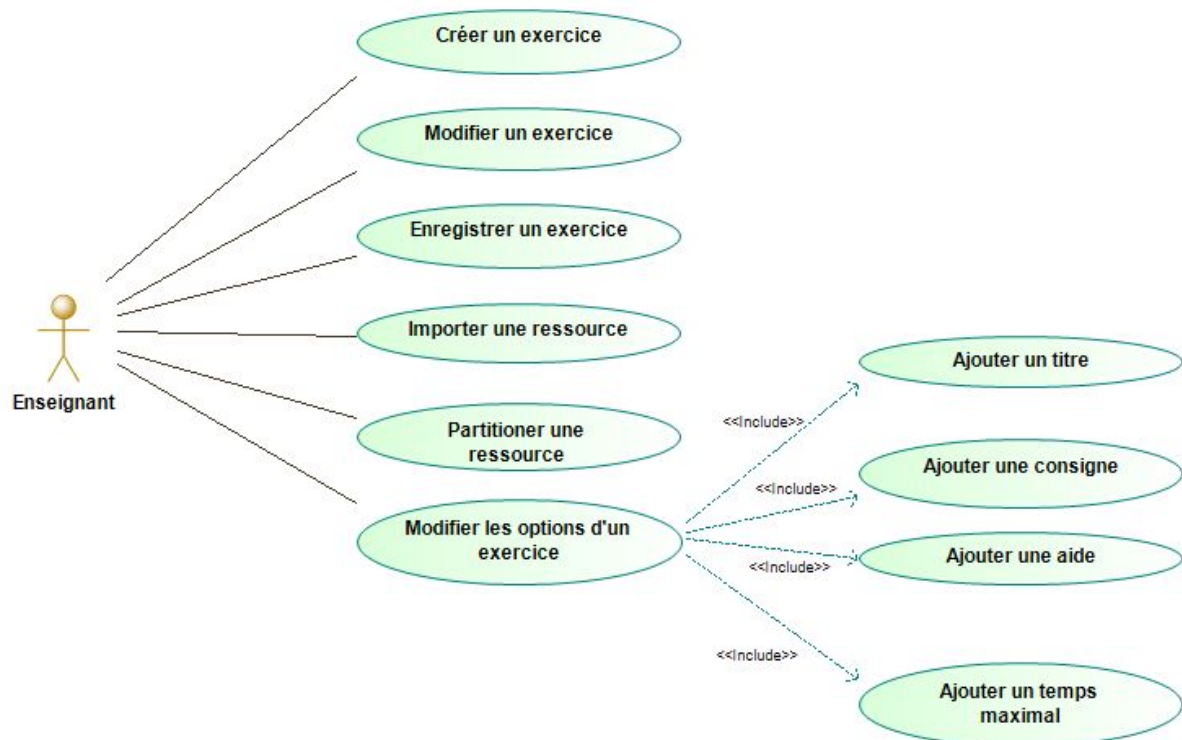


## Elève



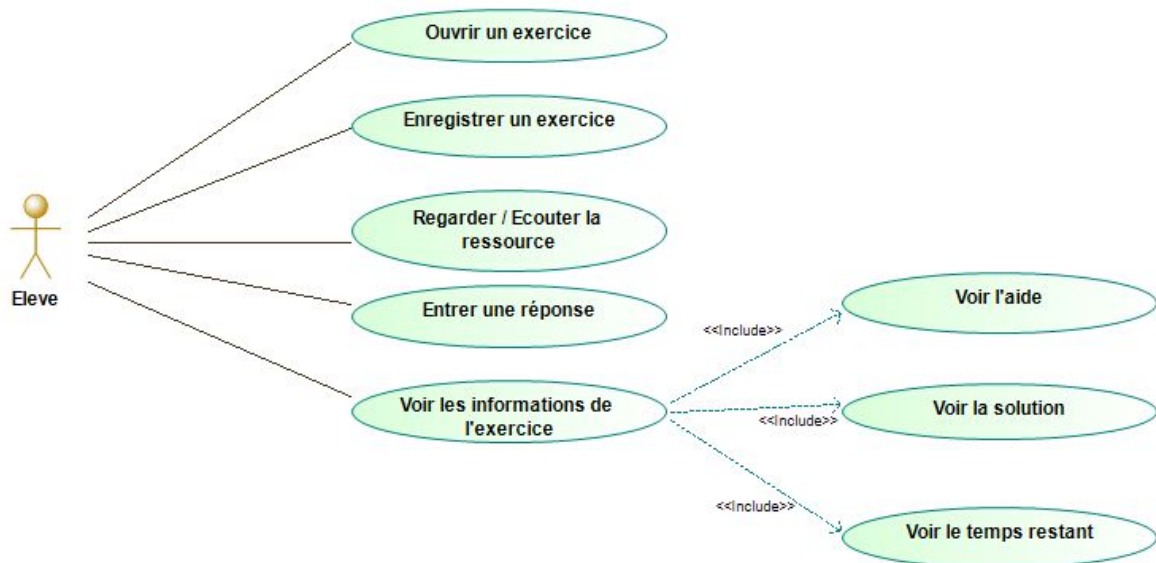
## Diagrammes de cas d'utilisation

### Enseignant





## Elève

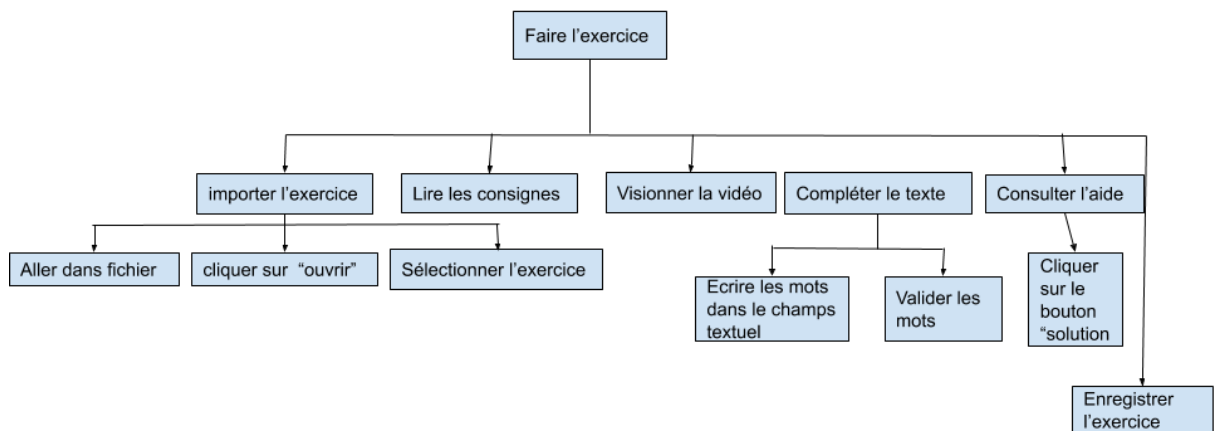


# Spécifications IHM

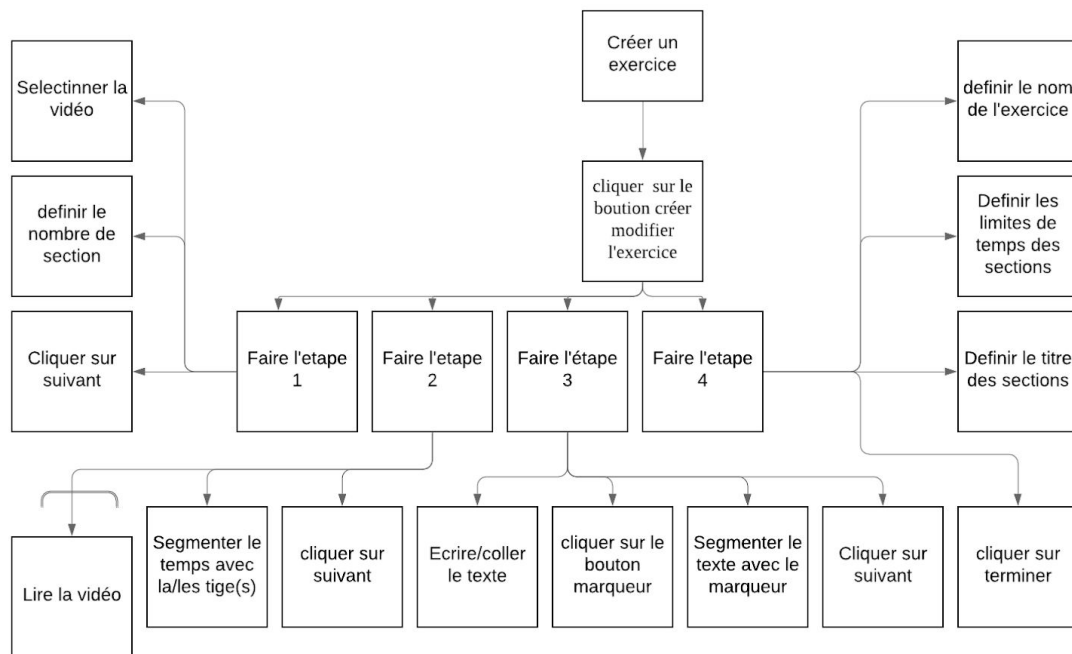
---

## Arbres des tâches

### Réaliser un exercice (élève)



## Créer un exercice (enseignant)



## Maquettes

### Enseignant

Interface principale

Reconstitution

i

?

Section 1

Section 2

Section 3

Section 4

Consigne

Aide

Options

☐ sensibilité à la casse

☒ iMode apprentissage

☐ Mot incomplet autoriser

☐ Autoriser l'affichage de la solution

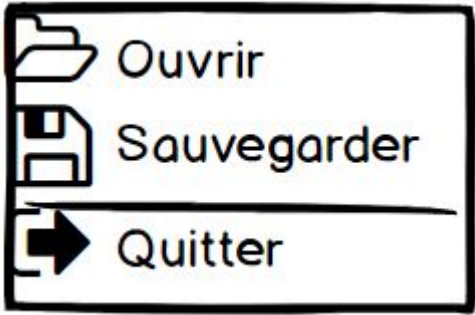
☐ disabled indeterminate

☐ Mode évaluation

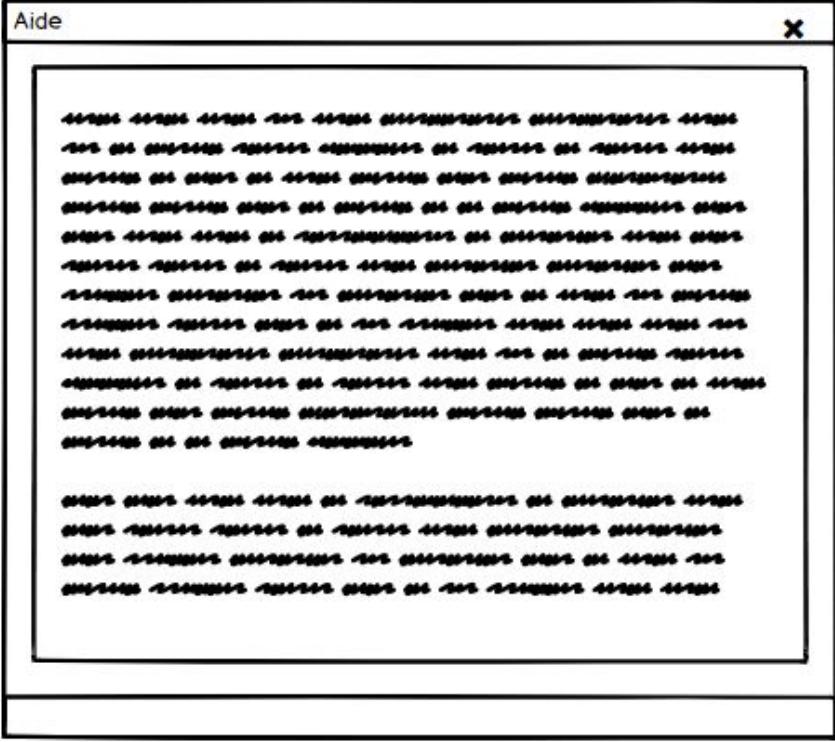
☐ Limite de temps

	Section 1	Section 2	Section 3	Section 4	Total
Temps					

Interface du menu fichier



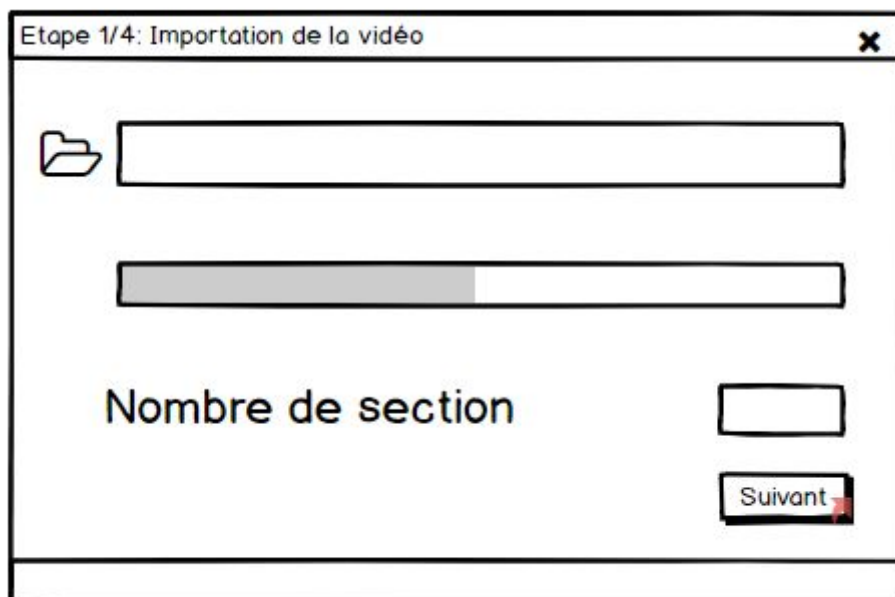
Interface de l'aide



Interface à propos



Interface Etape 1





Interface Etape 2

### Interface Etape 3

Etape 3/4: texte

Marqueur

suivant



## Interface Etape 4

Etape 4/4: Enregistrement du fichier

Nom du fichier :

Limite de temps :

	Titre	Temps
Section 1		
Section 2		
Section 3		
Section 4		

☐ sensibilité à la casse

☒ iMode apprentissage

☐ Mot incomplet autoriser

☐ Autoriser l'affichage de la solution

☒ disabled indeterminate

☐ Mode évaluation

Terminer

# Algorithmes

---

## Enseignant

[méthodes diagramme de classe]

## Créer un exercice

### VARIABLES

Utilisateur : enseignant  
Système : systeme  
Fichier : exercice

### DEBUT

**SI** (enseignant.CliqueSur("Fichier")) **ALORS**  
    **SI** (enseignant.CliqueSur("Nouveau")) **ALORS**  
        systeme.chargerScene("Edition")

**FIN SI**

**FIN SI**

**FIN**

## Associer une ressource à un exercice

### VARIABLES

Utilisateur : enseignant  
Système : systeme  
Fenêtre : explorateurDeFichiers  
Fichier : ressource

### DEBUT

**SI** (enseignant.CliqueSur("Insertion")) **ALORS**  
    **SI** (enseignant.CliqueSur("Ressource")) **ALORS**  
        systeme.AfficheFenêtre(explorateurDeFichiers)  
        **TANT QUE** (NON(enseignant.CliqueSur("Fermer")))  
            **SI** (enseignant.SelectionneFichier()) **ALORS**  
                systeme.ChargerFichierRessource(ressource)

**FIN SI**

**FIN TANT QUE**

    systeme.FermeFenêtre(explorateurDeFichiers)

**FIN SI**

**FIN SI**

**FIN**

## Enregistrer un exercice

### VARIABLES

Utilisateur : enseignant  
Système : systeme  
Fenêtre : explorateurDeFichiers  
Fichier : exercice  
Dossier : destination

### DEBUT

**SI** (enseignant.CliqueSur("Fichier")) **ALORS**

```

    SI (enseignant.CliqueSur("Enregistrer")) ALORS
        systeme.AfficheFenêtre(explorateurDeFichiers)
        TANT QUE (NON(enseignant.CliqueSur("Fermer")))
            SI (enseignant.SelectionneDossier(destination)) ALORS
                systeme.EnregistreFichier(exercice)
            FIN SI
        FIN TANT QUE
        systeme.FermeFenêtre(explorateurDeFichiers)
    FIN SI
FIN SI
FIN

```

Modifier un exercice

```

VARIABLES
    Utilisateur : enseignant
    Système : systeme
    Fichier : exercice

DEBUT
    SI (enseignant.CliqueSur("Fichier")) ALORS
        SI (enseignant.CliqueSur("Ouvrir")) ALORS
            systeme.AfficheFenêtre(explorateurDeFichiers)
        TANT QUE (NON(enseignant.CliqueSur("Fermer")))
            SI (enseignant.SelectionneFichier()) ALORS
                systeme.ChargerFichierExercice(exercice)
            FIN SI
        FIN TANT QUE
        systeme.FermeFenêtre(explorateurDeFichiers)
        systeme.chargerScene("Edition")
    FIN SI
FIN SI
FIN

```

## Etudiant

Ouvrir un exercice

```

VARIABLES
    Utilisateur : etudiant
    Système : systeme
    Fenêtre : explorateurDeFichiers
    Fichier : exercice

DEBUT
    SI (etudiant.CliqueSur("Fichier")) ALORS
        SI (etudiant.CliqueSur("Ouvrir")) ALORS

```

```

        systeme.AfficheFenêtre(explorateurDeFichiers)
        TANT QUE (NON(etudiant.CliqueSur("Fermer"))))
            SI (etudiant.SelectionneFichier(exercice)) ALORS
                systeme.ChargeFichier(exercice)
            FIN SI
        FIN TANT QUE
        systeme.FermeFenêtre(explorateurDeFichiers)
    FIN SI
FIN SI
FIN

```

Voir l'aide

```

VARIABLES
    Utilisateur : etudiant
    Système : systeme
    Fenêtre : aide
DEBUT
    SI (etudiant.CliqueSur("Aide")) ALORS
        systeme.AfficheFenêtre(aide)
        SI (etudiant.CliqueSur("Fermer")) ALORS
            systeme.FermeFenêtre(aide)
        FIN SI
    FIN SI
FIN

```

Enregistrer l'exercice

```

VARIABLES
    Utilisateur : etudiant
    Système : systeme
    Fenêtre : explorateurDeFichiers
    Fichier : exercice
    Dossier : destination
    BOOLEEN : limiteTemps //si oui ou non l'exercice doit être fait dans une
limite de temps
    Compteur : tempsRestant
DEBUT
    SI (limiteTemps==vrai) ALORS
        TANT QUE (tempsRestant != 0)
            etudiant.RepondExercice(exercice) //l'étudiant
effectue l'exercice
        FIN TANT QUE
        systeme.AfficheFenêtre(explorateurDeFichiers)
        TANT QUE (NON(etudiant.CliqueSur("Fermer"))))

```

```

        SI (etudiant.SelectionneDossier(destination)) ALORS
            systeme.EnregistreFichier(exercice)
        FIN SI
    FIN TANT QUE
    systeme.FermeFenêtre(explorateurDeFichiers)
SINON
    TANT QUE (NON(etudiant.CliqueSur("Enregistrer"))
        etudiant.RepondExercice(exercice) //l'étudiant
effectue l'exercice
    FIN TANT QUE
    systeme.AfficheFenêtre(explorateurDeFichiers)
    TANT QUE (NON(etudiant.CliqueSur("Fermer"))))
        SI (etudiant.SelectionneDossier(destination)) ALORS
            systeme.EnregistreFichier(exercice)
        FIN SI
    FIN TANT QUE
    systeme.FermeFenêtre(explorateurDeFichiers)
FIN SI
FIN

```

cryptage

**VARIABLE**

ENTIER i

STRING texteCrypter

**FONCTION** STRING cryptage(STRING texteExercice)

**Début FONCTION**

POUR (i=0 ,i<taille(texteExercice),i++){

```

    SI (lireCharc(texteExercice,i)=[A~z]){
        ajouterCharc(texteCrypter, «#»);
    }
    SINON{
        ajouterCharc(texteCrypter,lireCharc(texteExercice,i));
    }
}

RETOURNER(texteCrypter);

```

**FIN FONCTION**

vérification

**VARIABLE**

ENTIER i

ENTIER j

STRING texteAffiché

STRING texteExercice

BOLLEAN correct

**PROCEDURE** vérification(STRING reponseDonnée)

**Début PROCEDURE**

```

POUR (i=0 ,i<(taille(texteExercice)-(taille(reponseDonnée)-1)),i++){
    correct=true ;
    POUR (j=0 ,j<taille(reponseDonnée),j++){
        SI (lireCharc(reponseDonnée,j) != lireCharc(texteExercice,i+j) ){
            correct=false ;
        }
    }
}

```



```
SI (correct=true){  
    POUR (j=0 ,j<taille(reponseDonnée),j++){  
        remplacerCharc(texteAffiché,i+j,lireCharc(reponseDonnée,j)) ;  
    }  
}
```

**Fin PROCEDURE**



# Glossaire

---

## Diagramme de classe

Schéma permettant de représenter les différentes classes au sein du code et leurs relations.

## Diagramme de cas d'utilisation

Schéma permettant de représenter les différentes utilisations possibles d'un système.

## IHM

Interface Homme-Machine, rassemble tous les aspect ergonomiques de l'application