

# Projet de programmation

## Cahier des besoins - Mots croisés

Albanel Clément, Alves Nathanael, Auzi Emma, Bornes Yohan, Coudougnan Sylvain, Devidas  
Hugo

Février 2022

## Table des matières

<b>1</b>	<b>Objet de la demande</b>	<b>2</b>
1.1	Utilisateurs et parties prenantes du projet . . . . .	2
1.1.1	En interne . . . . .	2
1.1.2	En externe . . . . .	2
1.2	Description des objectifs généraux . . . . .	2
<b>2</b>	<b>Analyse de l'existant</b>	<b>3</b>
2.1	Introduction . . . . .	3
2.2	Outils en ligne . . . . .	3
2.3	Logiciel nécessitant une installation . . . . .	7
2.4	Synthèse . . . . .	8
<b>3</b>	<b>Besoins fonctionnels</b>	<b>8</b>
3.1	Avant-propos . . . . .	8
3.2	Diagramme des cas d'utilisation . . . . .	10
3.3	Besoins utilisateurs . . . . .	11
<b>4</b>	<b>Besoins non fonctionnels</b>	<b>29</b>
4.1	Besoins d'utilisabilité . . . . .	29
4.2	Besoins de performances . . . . .	29
4.3	Besoins de sécurité . . . . .	29

## Table des figures

1	Interface Application "The Teacher's Corner : puzzle and activities" . . . . .	3
2	Interface Application "Sport Cérébral : mots croisés" . . . . .	4
3	Interface Application "Education.com : Crossword Puzzle" . . . . .	5
4	Interface Application "Puzzel" . . . . .	6
5	Interface Application "Eclipse Crossword" . . . . .	7
6	Interface Application "Mots croisés pro" . . . . .	8
7	Diagramme de cas d'utilisation . . . . .	10
8	Maquette - Accueil . . . . .	11
9	Maquette - Menu d'importation . . . . .	12
10	Maquette - Grille précédemment remplie importée . . . . .	12
11	Maquette - Choix de la difficulté . . . . .	14
12	Maquette - Choix du thème . . . . .	15
13	Maquette - Paramétrage de la grille . . . . .	16
14	Maquette - Grille vide . . . . .	17
15	Maquette - Partie en cours . . . . .	19
16	Maquette - Menu fichier . . . . .	21
17	Maquette - Menu export . . . . .	22
18	Maquette - Aides . . . . .	24
19	Maquette - Solution . . . . .	26
20	Maquette - Grille remplie . . . . .	28
21	Maquette - Validation . . . . .	29

# 1 Objet de la demande

Le projet à réaliser consiste en la réalisation d'un programme capable de générer automatiquement des grilles de mots-croisés, dont les difficultés sont variables. Outre la génération, l'utilisateur doit pouvoir y jouer.

## 1.1 Utilisateurs et parties prenantes du projet

### 1.1.1 En interne

Le projet est réalisé dans le cadre de l'UE "Projet de Programmation" en Master 1 à l'Université de Bordeaux. Le commanditaire est Monsieur Philippe Narbel et le projet est suivi par Madame Wahiba Larbi. L'équipe de développement du projet est composée de Albanel Clément, Alves Nathanael, Auzi Emma, Bornes Yohan, Coudougnan Sylvain et Devidas Hugo.

### 1.1.2 En externe

Les clients concernés par l'application développée sont des enseignants chercheurs travaillant au Laboratoire Bordelais de Recherche pour l'Informatique.

## 1.2 Description des objectifs généraux

Le but premier de l'application à développer est de construire un programme capable de générer automatiquement des grilles de mots croisés. Il s'agira d'un jeu traditionnel de mots croisés et nous accorderons une grande importance aux techniques algorithmiques de génération de mots croisés. Ainsi, nous focaliserons nos recherches selon ces 4 références principales :

- L'article universitaire [8] publié à l'Université de Stanford traite des problèmes appartenant à la classe algorithmique nommée "Problème de satisfaction de contraintes". Il s'agit de problèmes mathématiques où l'on cherche des états qui satisfont un ensemble de critères ou de contraintes données. Le problème des mots croisés en est un. En effet, générer une grille de jeu implique évidemment qu'elle soit résoluble, et donc il s'agira de notre contrainte principale (bien sûr, elle implique de nombreuses sous-contraintes). Ce papier universitaire expose des algorithmes et techniques utilisées pour la résolution des problèmes de satisfaction de contraintes, appliquée à la génération de mots croisés.
- Le second article [2] issu de l'Université Nationale Australienne est directement lié au précédent, puisqu'il présente les mots croisés en tant que problème de satisfaction de contraintes. Plus précisément, à travers l'article, les 2 auteurs présentent des nouvelles techniques de génération de mots croisés, bien plus efficaces que les techniques utilisées jusqu'alors. Le modèle présenté repose sur deux points de vue différents : l'un contenant des variables de cellules, et l'autre contenant des variables d'emplacement de mots. Ce modèle hybride permet d'exploiter au maximum les avantages de chacune des 2 techniques, indépendamment.
- Par ailleurs, le papier universitaire [1] se base sur la recherche de solutions via les arbres de décisions. Cela permet, combiné à des heuristiques performantes d'aiguiller l'exploration de l'arbre et donc de trouver une solution plus rapidement à une grille donnée. Il a été montré que l'arbre de décisions influence largement l'efficacité de l'algorithme de recherche. L'idée est ici de montrer comment il est possible de construire l'arbre de décision, indépendamment de l'heuristique et ainsi d'obtenir de meilleures performances algorithmiques. Pour cela, le jeu des mots croisés est utilisé comme modèle.

- Enfin l'article de journal [7] écrit par Matthew L. Ginsberg aborde un solveur particulier, qui résout les grilles de mots croisés (variante Américaine). L'idée est de convertir le jeu des mots croisés en problème de satisfaction de contraintes et de ce fait de pouvoir utiliser des nouvelles techniques pour trouver des solutions.

Bilan des 4 articles : bien qu'il s'agisse essentiellement de problèmes abordants des solveurs, ils font néanmoins partie intégrante de la génération de grille, puisqu'une fois la grille générée, il s'agit de vérifier si cette dernière est résoluble.

## 2 Analyse de l'existant

### 2.1 Introduction

La plupart des outils que nous avons analysés sont des applications web et utilisent les technologies associées. Nous logiciel s'installera sur machine, cependant la plupart de ces outils gratuits et très facile d'utilisation restent une source d'inspiration conséquentes pour notre projet. Certains logiciel plus poussés sont également payants ou disposent d'options payantes. Ces outils sont plutôt similaires, mais des aspects comme la personnalisation de la grille ou l'interface utilisateur pourront être un peu plus poussés chez certains. Nous allons en présenter les principaux que nous avons retenus et ce que nous avons prévus de garder de ces outils.

### 2.2 Outils en ligne

- The Teacher's Corner : puzzle and activities [4]

Il permet de générer des grilles pré-faites sur différents thèmes comme les fêtes, la géographie, les langues etc.. Ou bien de générer soi-même sa propre grille en rentrant nous même nos propres mots et leurs définitions associées. Il permet de grandes options sur l'affichage de la grille, comme le choix des polices, changement de la taille et des couleurs des textes ou des cellules, ajouter du texte ou des images. Une fois que le projet est prêt on peut choisir de l'exporter avec ou sans la solution inscrite dans la grille. Bien que l'on puisse directement remplir les champs, l'usage de cet outil semble destiné à exporter la grille en image ou en pdf dans le but d'une éventuelle impression. Cet outil est à destination d'une cible anglophone si l'usage principal est d'utiliser les templates existants.

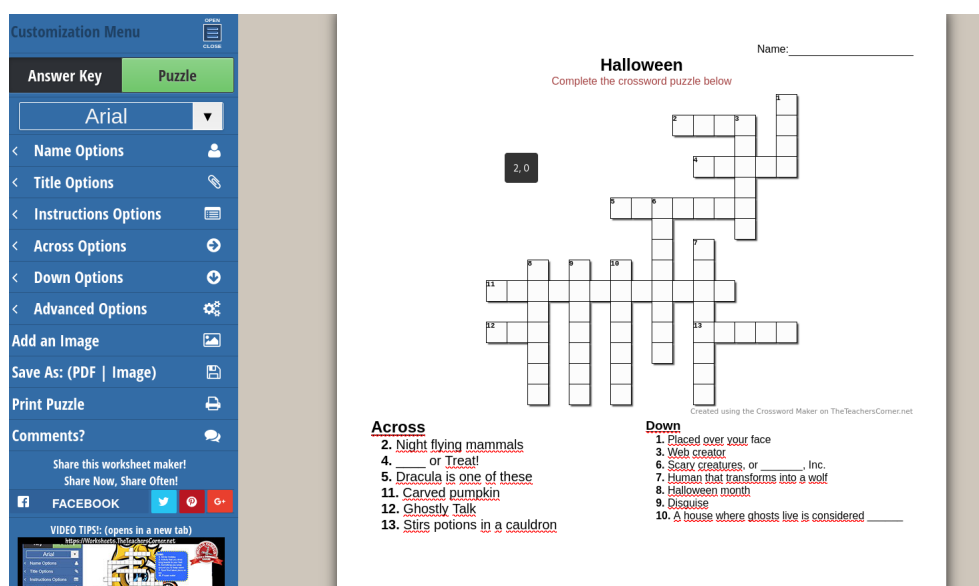


Figure 1: Interface Application "The Teacher's Corner : puzzle and activities"

- Sport Cérébral : mots croisés [9]

Cet outil permet de remplir une unique grille chaque jour dans le but de se divertir. Il ne permet aucune option de génération ou d'exportation. Cependant dans les fonctionnalités on a accès à des boutons indices et des options ergonomiques tels que le zoom, la possibilité d'effacer nos mots, etc pour enfin valider notre grille. D'un point de vue design et besoins utilisateurs, cette application est en adéquation avec le projet présenté.

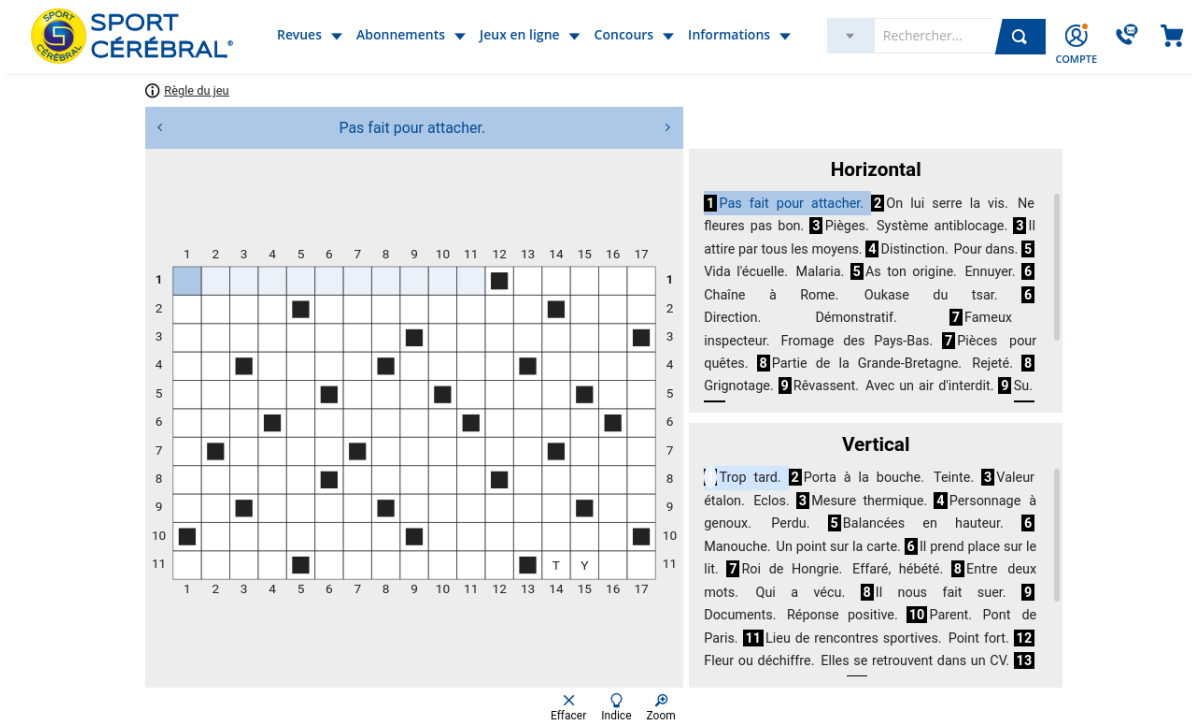


Figure 2: Interface Application "Sport Cérébral : mots croisés"

- Education.com : Crossword Puzzle [6]

Outil permettant de générer ses propres mots croisés, avec une très légère option de personnalisation du design (bordures) et possibilité d'export/import de nos propres grilles.

## Crossword Puzzle

### Reading worksheet generator

Options
Theme

#### Enter crossword puzzle content

See instructions below. Under 50 words is best. More help: [help](#)

Title

Titre test

Words

alligator, a reptile that looks a little like a crocodile  
tiger, a big orange cat with black stripes  
spider, an eight-legged bug that spins webs

+ Create

#### Preview Worksheet

WORKSHEET
ANSWER KEY

Titre test

Down:

- an eight-legged bug that spins webs
- a reptile that looks a little like a crocodile

Across:

- a big orange cat with black stripes

Education.com

Scramble
Download Worksheet
Download Answer Key

Save worksheet
Crossword Puzzle
Save

Figure 3: Interface Application "Education.com : Crossword Puzzle"

- Puzzel [10]

Outil à l'interface soignée, beaucoup d'options ne sont malheureusement accessibles qu'avec une participation financière.

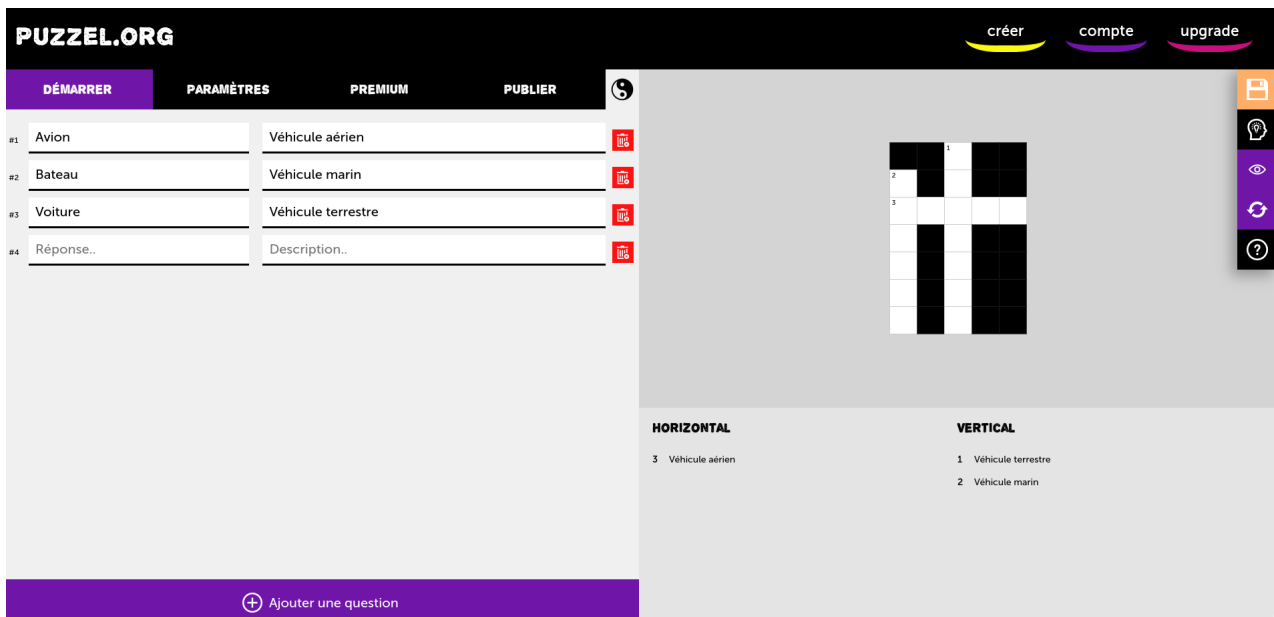


Figure 4: Interface Application "Puzzel"

## 2.3 Logiciel nécessitant une installation

- EclipseCrossword [5]

Disponible sur le catalogue d'applications Microsoft. Ce logiciel est gratuit et disponible uniquement sur Windows.

Logiciel complet de création de grille. Possibilité d'importer une liste de mots/indices. Possibilité d'enregistrer sa grille dans le format du logiciel (.ecw) et sa "word list", d'imprimer avec plusieurs options (présence des réponses dans la grille, afficher les indices, etc...) et partager sa grille en ligne. On trouve également des options plus avancées comme des exports html, rtf, wmf, eps et "across lite text format".

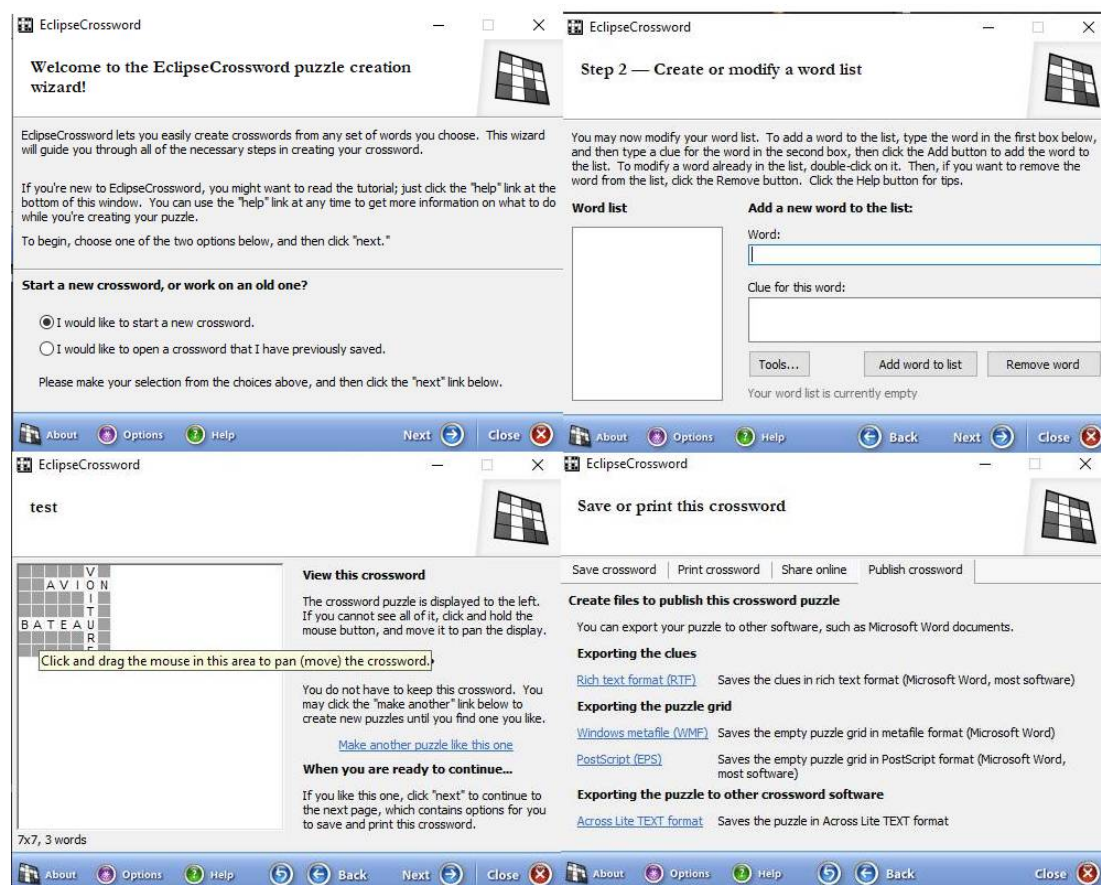


Figure 5: Interface Application "Eclipse Crossword"

- Mots-croisés-Pro [3]

Logiciel payant disponible uniquement sur Windows au prix de 59€. Une version gratuite d'essai est disponible.

Ce logiciel permet la création de grilles de mots croisés ou fléchés jusqu'à 400 cases. Il permet de créer des grilles de manière automatique, semi-automatique ou manuelle. Le grand avantage est de pouvoir se servir des bases de données existantes pour nos premiers tests. Outil extrêmement complet, mais malheureusement limité dans sa version gratuite.



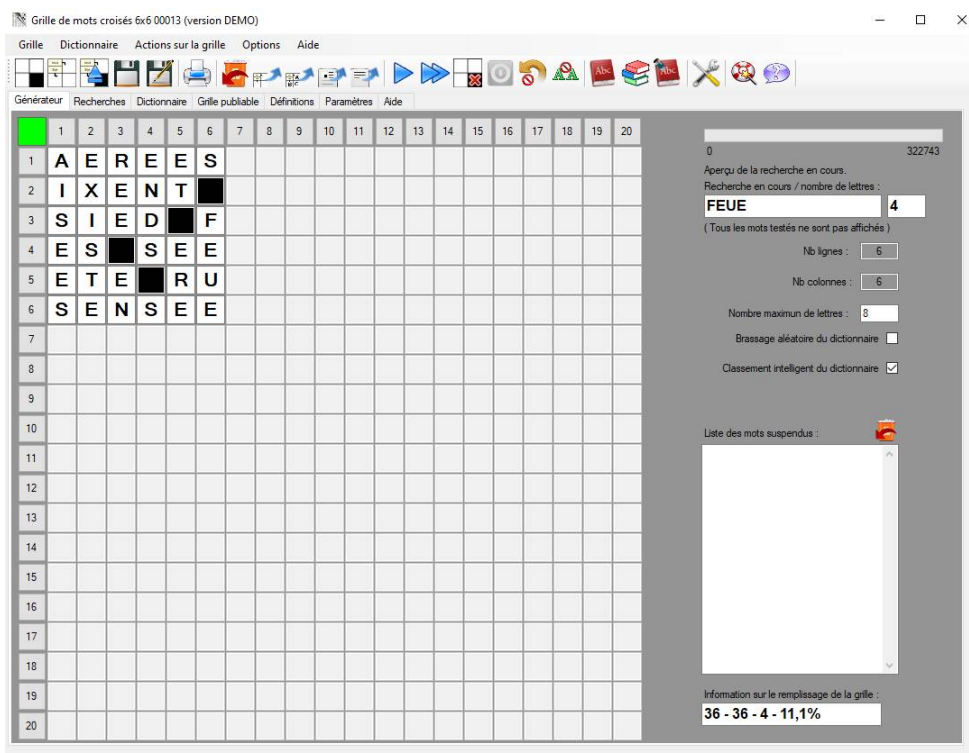


Figure 6: Interface Application "Mots croisés pro"

## 2.4 Synthèse

La plupart des générateurs facilement trouvables en ligne permettent de créer des grilles très simples, la cible de ces générateurs sont probablement des enfants ou un public non expert des mots-croisés, l'utilisation de tels outils semble adaptée à un but éducatif. En revanche aucun des ces outils ne semble proposer de solutions plus avancées telles que l'on pourrait trouver dans des journaux ou magazines spécialisées. En effet si l'on cherche à construire une grille complexe avec peu de blocs noirs (absence de lettres) et des normes précises comme une taille de grille particulière ou le respect de la symétrie (voir figure plus bas), il faut nous pencher vers des logiciels plus avancés et souvent payant, comme le fait Mots-croisés-Pro. De plus ces logiciels sont davantage une aide à la construction manuelle de grille, qu'une solution toute faite automatisée.

Dans notre application, les options de personnalisation poussées ne seront pas prioritaires, elles viendront enrichir notre logiciel que dans un second temps. Nous voulons nous inspirer des options d'aide à la résolution de grille pour l'utilisateur comme ceux présents dans Sport Cérébral par exemple. Nous ne prévoyons pas d'utiliser le web pour notre application (pour ne pas nous préoccuper des conventions du web telles que les requêtes HTTP), ainsi l'interface utilisateur devrait être bien différente des générateurs en ligne que l'on a pu étudier ici. Le partage et la publication de grilles ne nous concerne pas particulièrement, les options d'import/export à l'usage d'un seul utilisateur nous suffisent.

## 3 Besoins fonctionnels

### 3.1 Avant-propos

Nous utiliserons les codes de correspondances suivants au sein des besoins :

- **(E)** : Essentiel → le logiciel ne sera pas acceptable sans que ce besoin soit réalisé.
- **(C)** : Conditionnel → besoin qui étend et améliore le logiciel, sans que ce besoin soit nécessaire pour rendre le logiciel acceptable.

- **(O)** : Optionnel → besoin dont la valeur n'est pas encore assurée.

Pour chacun des besoins présentés, nous écrirons ici 3 différents tests :

- test **positif** (ou test d'affirmation) qui vérifient le bon fonctionnement du besoin dans des conditions d'utilisations correctes (qui ne "stressent" pas les cas d'utilisation du logiciel).
- test **négatif** qui vérifient le comportement hors des cas d'utilisation définis par les besoins du logiciel.
- test **aux frontières**, ils s'appliquent aux limites/frontières de fonctionnement normal. Ils sont donc situés à la frontière entre les cas d'utilisation entraînant un test positif, et ceux entraînant un test négatif. A noter que ce test ne peut pas s'appliquer à tous les besoins, nous ne le mettrons donc pas toujours (comme parfois les tests négatifs).

Pour chacun des tests énoncés ci-après, nous énoncerons d'abord le nom du besoin tel qu'exprimé précisément dans le cahier des besoins, puis écrirons les 3 tests selon l'ordre expliqué en introduction. Nous séparerons chacun des 3 tests selon la charge graphique suivante : fond bleu = test positif / fond rouge = test négatif / fond gris = test aux frontières.

### 3.2 Diagramme des cas d'utilisation

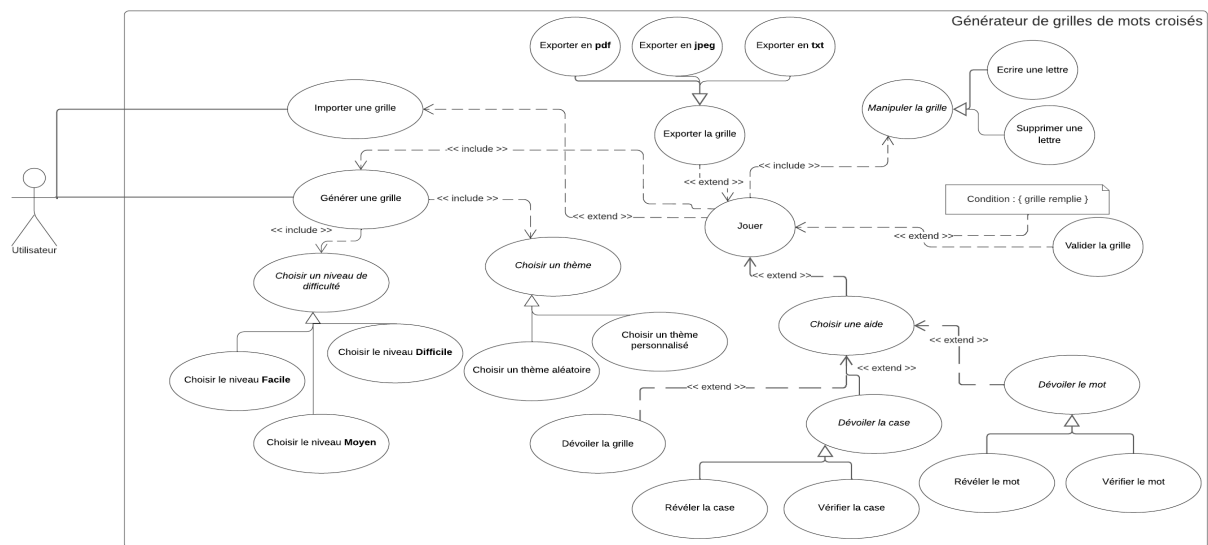


Figure 7: Diagramme de cas d'utilisation

### 3.3 Besoins utilisateurs

#### (O) Importer une grille

Acteur : Utilisateur.

Résumé : L'utilisateur souhaite importer une grille spécifique sur l'application pour y jouer.

Préconditions : L'application est correctement lancée.

Post conditions : L'utilisateur peut jouer sur la grille qu'il a importée.

Enchaînement nominal :

Étapes	Utilisateur	Système	Maquette correspondante
1	Clique sur "File/Import"		Maquette - Accueil
2	Sélectionne le fichier correspondant		Maquette - Menu d'importation
3		Vérifie la validité du fichier (cf. codage d'une grille)	/
4		Affiche la grille ou une erreur d'importation	Maquette - Grille précédemment remplie importée

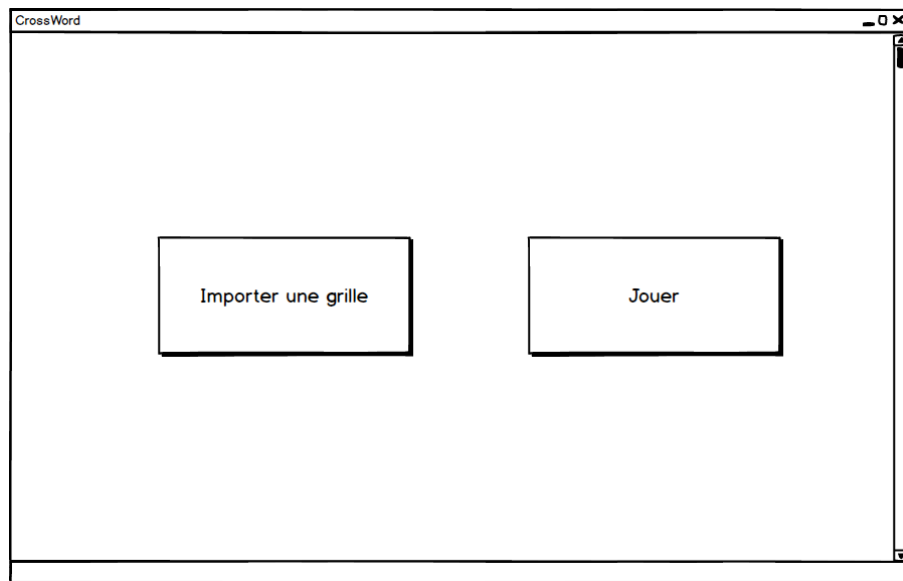


Figure 8: Maquette - Accueil

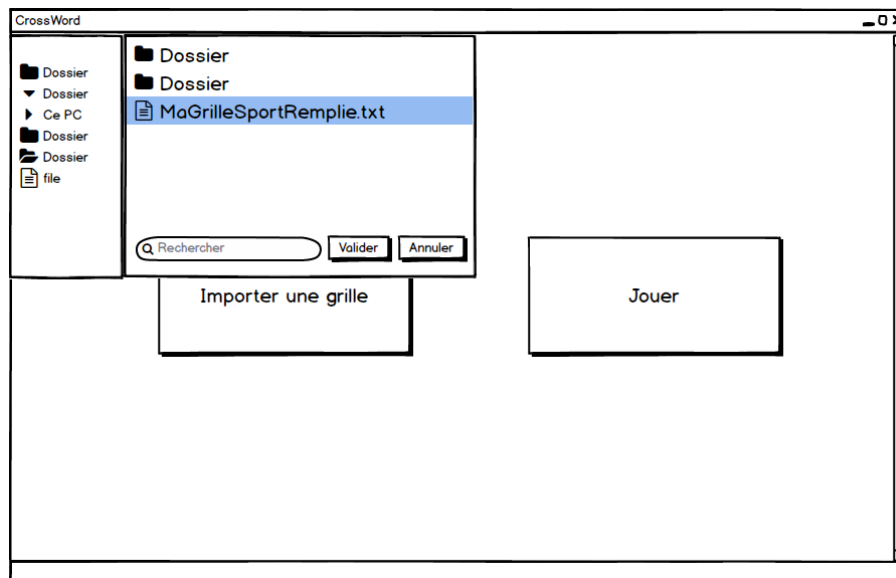


Figure 9: Maquette - Menu d'importation

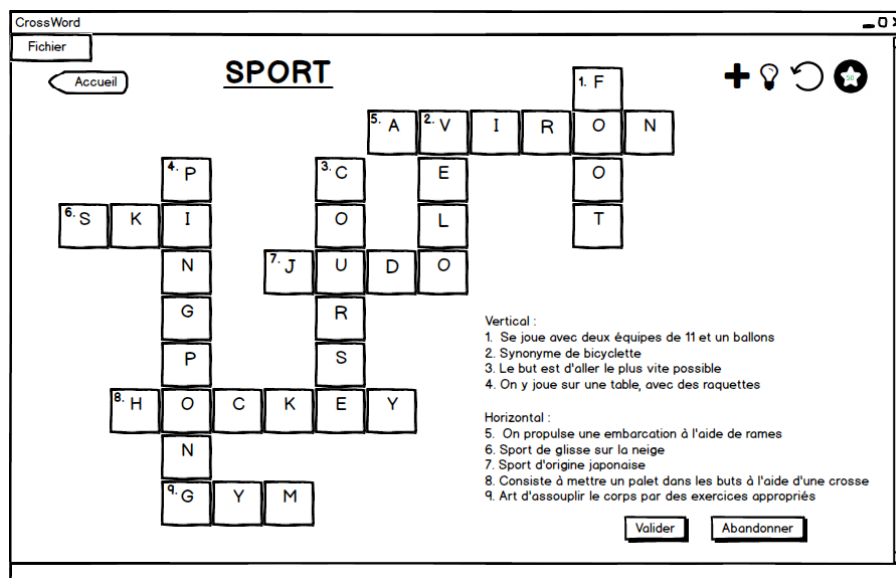


Figure 10: Maquette - Grille préemment remplie importée

Codage d'une grille : La grille sera codée dans un fichier .txt, nous devons stocker les différents mots et définitions pour que le système arrive à régénérer correctement la grille. Le format précis du codage sera précisé lors des spécifications techniques, si le fichier .txt ne respecte pas le codage, le système retournera une erreur lors de l'importation.

Tests associés :

1. *Description et but du test* : Importation d'une sauvegarde valide d'une grille.
2. *Cas de test* : On utilise la sauvegarde d'une grille "témoin" valide. On importe cette grille sauvegardée et on vérifie que l'importation de cette dernière a bien été effectuée.
3. *Déroulement et scénario du test* : On sauvegarde une grille sur notre machine, on l'importe et on vérifie que la grille importée est la même que la grille fraîchement sauvegardée.
4. *Analyse du test* : Si la grille importée est bien celle attendue alors le test est valide, invalid sinon.

1. *Description et but du test* : Importation d'une sauvegarde non valide d'une grille (format de fichier non conforme).
2. *Cas de test* : Un fichier de sauvegarde invalide d'une grille. Il est attendu que le logiciel ne se charge pas et qu'un message d'erreur s'affiche.
3. *Déroulement et scénario du test* : On importe la sauvegarde invalide dans notre logiciel avec l'aide du bouton "Importer une grille".
4. *Analyse du test* : Si un message d'erreur indiquant que le fichier est invalide s'affiche alors le test est valide, sinon il est échoué.

## (E) Choisir niveau de difficulté

Acteur : Utilisateur.

Résumé : L'utilisateur peut choisir une difficulté pour la grille de mots croisés (il se rend dans "play/select difficulty/difficulty wanted).

Préconditions : L'application est correctement lancée et l'utilisateur a choisi de commencer à jouer.

Post conditions : L'utilisateur a la liberté de jouer à différentes difficultés.

Maquette :

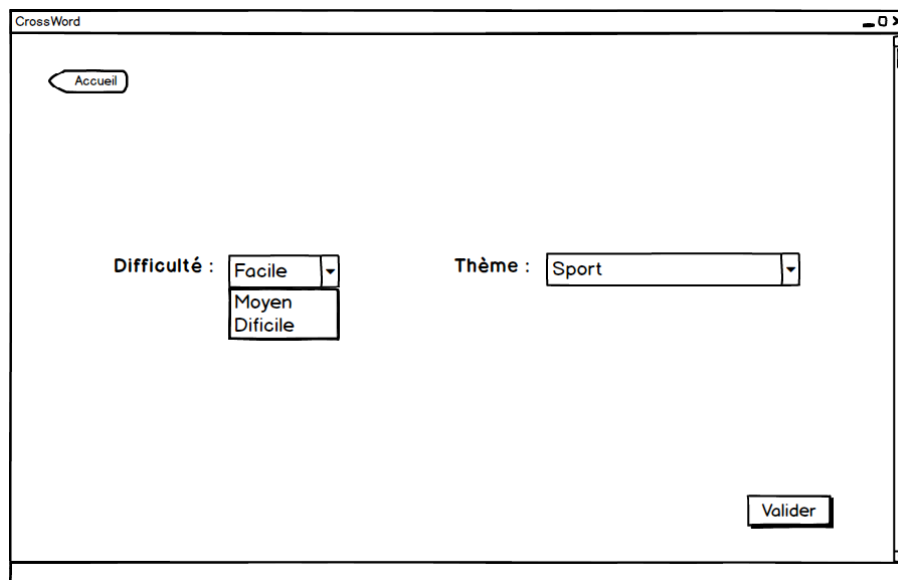


Figure 11: Maquette - Choix de la difficulté

Les différentes difficultés : La difficulté d'une grille sera définie par la taille de la grille. Nous aurons 3 niveaux de difficultés : Facile (grille 6x6), Moyen (grille 8x8) et Difficile (grille 10x10).

Test associé :

1. *Description et but du test* : Vérifier que la génération des grilles est bien modifiée en fonction du niveau de difficulté.
2. *Cas de test* : Utilisation du logiciel témoin (sans condition de stress). Le résultat attendu est que la grille soit générée avec la méthode qui représente la difficulté sélectionnée.
3. *Déroulement et scénario du test* : On génère la grille avec chaque difficulté possible, à chaque génération on vérifie que la grille sélectionnée est générée avec la méthode reliée correspondant à sa difficulté.
4. *Analyse du test* : Si chaque grille a été générée avec la méthode correspondant à son niveau de difficulté alors le test est validé sinon il a échoué.

## (O) Choisir un thème pour la grille

Acteur : Utilisateur.

Résumé : L'utilisateur peut choisir un thème pour la grille de mots croisés.

Préconditions : L'application est correctement lancée et l'utilisateur a choisi de commencer à jouer.

Post conditions : L'utilisateur a la liberté de jouer sur des grilles de différents thèmes.

Maquette :

La maquette représente une fenêtre d'application intitulée "CrossWord". Dans le coin supérieur gauche, il y a un bouton "Accueil" avec une flèche pointant à gauche. Au centre, il y a deux menus déroulants. Le premier est étiqueté "Difficulté" et est actuellement réglé sur "Facile". Le second est étiqueté "Thème" et est actuellement réglé sur "Sport". Le menu "Thème" est ouvert, montrant une liste de thèmes : "Cinéma", "Animaux", "Noel", "Informatique" et "Pas de thème". En bas à droite de la fenêtre, il y a un bouton "Valider".

Figure 12: Maquette - Choix du thème

Les différents thèmes : Les thèmes influenceront les mots qui composent la grille, par exemple un thème de sport donnera une grille avec des mots du lexique du sport. Les thèmes auxquels nous avons pensé sont : Sport, Cinéma, Animaux, Noël, Informatique.

Test associé :

1. *Description et but du test* : Vérifier que les mots utilisés pour créer la grille viennent de la liste de mots correspondant au thème.
2. *Cas de test* : Utilisation du logiciel témoin (sans condition de stress) et des listes de mots des différents thèmes. On s'attendait à ce que les grilles générées avec un thème n'utilisent que des mots de la liste de son thème.
3. *Déroulement et scénario du test* : On génère la grille avec chaque thème et pour chaque grille on vérifie que seuls les mots de leurs thèmes ne composent la grille.
4. *Analyse du test* : Si chaque grille n'est composée que des mots correspondant à son thème alors le test est validé, sinon le test échoue.



## (C) Changer de grille

Acteur : Utilisateur.

Résumé : L'utilisateur n'est pas satisfait d'une grille et veut donc changer de grille.

Préconditions : L'application est correctement lancée, l'utilisateur est en train de jouer sur une grille.

Post conditions : La nouvelle grille est affichée, l'utilisateur peut maintenant jouer.

Enchaînement nominal :

Étapes	Utilisateur	Système	Maquette correspondante
1	Clique sur "Changer de grille"		(Bouton + de l'écran de jeu)
2		Affiche les différentes options	Maquette - Paramétrage de la grille
3	Choisi une difficulté (cf. "Choisir une difficulté")		(cf. "Choisir une difficulté")
4	Choisi un thème (cf. "Choisir un thème")		(cf. "Choisir un thème")
5	Valide sa sélection		Maquette - Paramétrage de la grille
6		Génère la grille	/
7		Affiche la grille	Maquette - Grille vide

CrossWord

Accueil

Difficulté : Facile

Thème : Sport

Valider

Figure 13: Maquette - Paramétrage de la grille

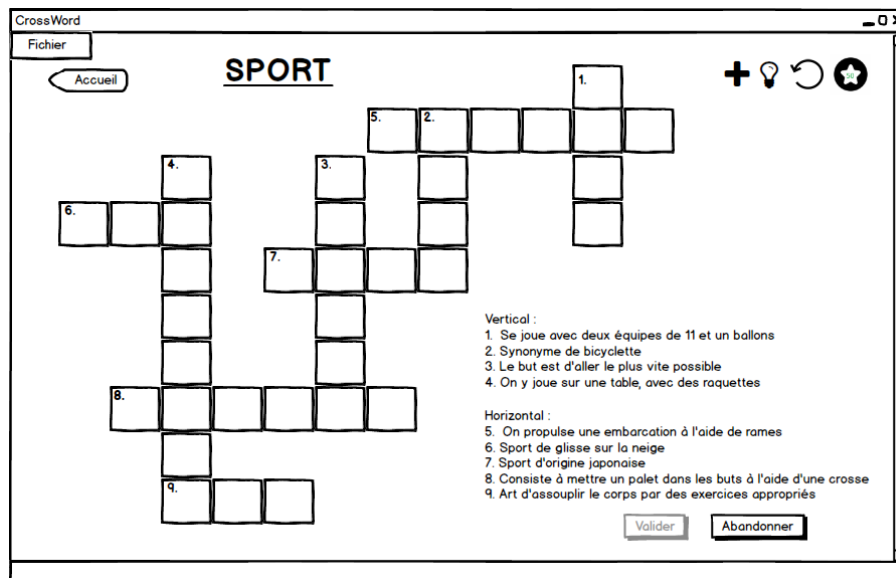


Figure 14: Maquette - Grille vide

#### Tests associés :

1. *Description et but du test* : Vérification que le bouton changer de grille génère bien une nouvelle grille.
2. *Cas de test* : Utilisation du logiciel témoin (sans condition de stress) . On s'attend a ce que le logiciel génère une nouvelle grille.
3. *Déroulement et scénario du test* : On clique sur le bouton "Changer de grille".
4. *Analyse du test* : On génère une nouvelle grille. On vérifie que cette dernière n'est composée que de mots appartenant au thème de l'ancienne grille.

1. *Description et but du test* : Vérification que le bouton changer de grille n'est pas disponible si aucune grille n'est généré.
2. *Cas de test* : Utilisation du logiciel témoin (sans condition de stress) . On s'attend a ce que le bouton de changement de grille ne soit pas affiché.
3. *Déroulement et scénario du test* : On lance l'application et on vérifie que le bouton n'est pas présent.
4. *Analyse du test* : Le test est un succès si le bouton n'est pas affiché.

## (E) Lancer la génération d'une grille

Acteur : Utilisateur.

Résumé : L'utilisateur lance la génération d'une grille de mots croisés en vue d'y jouer (détaillé sur le PDF Algorithmes, choix techniques, outils utilisés).

Préconditions : L'application est correctement lancée.

Post conditions : La grille est affichée, l'utilisateur peut maintenant jouer.

Enchaînement nominal :

Étapes	Utilisateur	Système	Maquette correspondante
1	Clique sur " Jouer"		(cf. Maquette - Accueil)
2		Affiche les différentes options	(cf. Maquette - Paramétrage de la grille)
3	Choisi une difficulté (cf. "Choisir une difficulté")		(cf. "Choisir une difficulté")
4	Choisi un thème (cf. "Choisir un thème")		(cf. "Choisir un thème")
5	Valide sa sélection		(cf. Maquette - Paramétrage de la grille)
6		Génère la grille	/
7		Affiche la grille	(cf. Maquette - Grille vide)

Test associé :

1. *Description et but du test* : Vérifier que la génération de la grille fonctionne.
2. *Cas de test* : Génération de la grille en fonction du thème et de la difficulté choisi par l'utilisateur.
3. *Déroulement et scénario du test* : Sélection d'un thème et d'une difficulté, valider la sélection.
4. *Analyse du test* : Vérifie que la grille est bien solvable par l'utilisation de notre solveur. Vérification de la bonne appartenance des mots de la grille au thème et à la difficulté sélectionnée préalablement.

## (E) Jouer (remplir les cases)

Acteur : Utilisateur.

Résumé : L'utilisateur veut jouer à une grille de mots croisés.

Préconditions : L'application est correctement lancée.

Post conditions : La grille est affichée, l'utilisateur peut maintenant jouer.

Enchaînement nominal :

Étapes	Utilisateur	Système	Maquette correspondante
1	Sélectionne une case		(cf. Maquette - Grille vide)
2		Prend en compte la case sélectionnée	(cf. Maquette - Grille vide)
3	Rentre une nouvelle valeur (au clavier)		/
4		Affiche la nouvelle valeur de la case	Maquette - Partie en cours

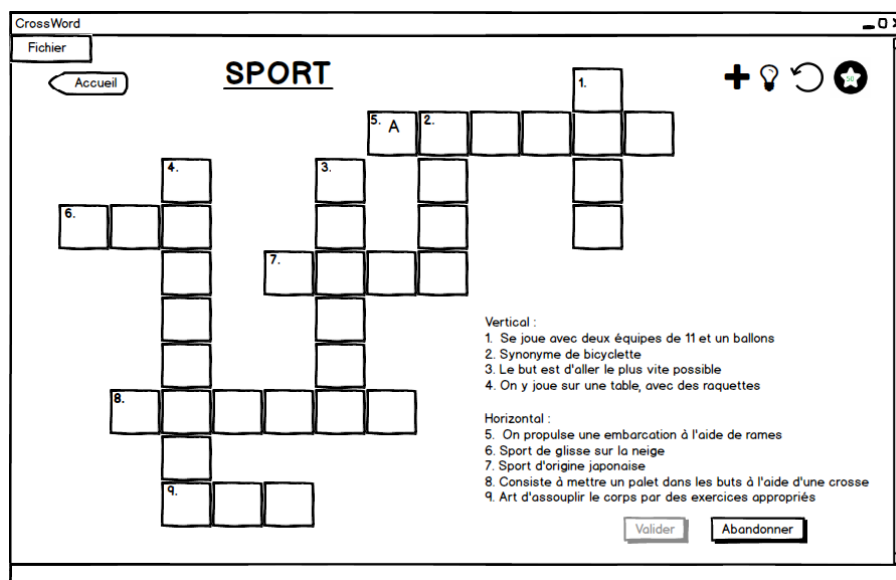


Figure 15: Maquette - Partie en cours

Test associé :

1. *Description et but du test* : L'utilisateur doit pouvoir remplir les cases de la grille par des lettres  $\in [a - z]$ .
2. *Cas de test* : Ecrire dans une case avec différents caractères.
3. *Déroulement et scénario du test* : A partir d'une nouvelle grille, essayer de remplir des cases avec des caractères acceptés par le programme et par des caractères spéciaux ou majuscule (non acceptés)
4. *Analyse du test* : Vérifie que les caractères spéciaux et majuscule ne sont pas entrés dans la case sélectionnée. Seul les caractères  $\in [a - z]$  seront acceptés.

## (O) Exporter la grille

Acteur : Utilisateur.

Résumé : L'utilisateur veut exporter une grille de mots croisés.

Préconditions : L'application est correctement lancée, une partie est lancée, la grille est correctement affichée.

Post conditions : L'utilisateur possède la grille sous un format externe (pdf, jpeg ou txt).

Enchaînement nominal :

Étapes	Utilisateur	Système	Maquette correspondante
1	Selectionne "Export" dans le menu File		Maquette - Menu fichier
2	Choisi le format (pdf, jpeg ou txt)		Maquette - Menu export
3		Génère le fichier au format demandé (cf. ci-dessous explication des différent formats)	/
4	Enregistrer le fichier		/

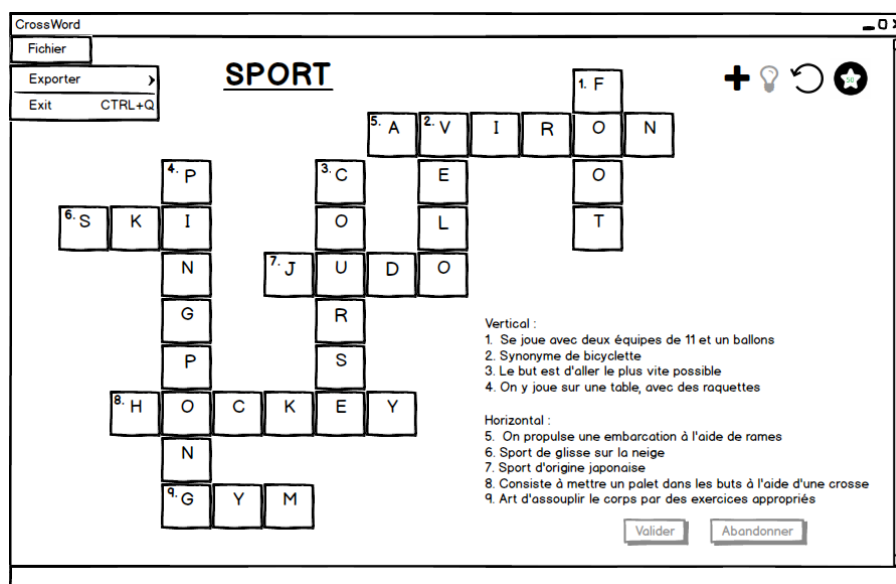


Figure 16: Maquette - Menu fichier

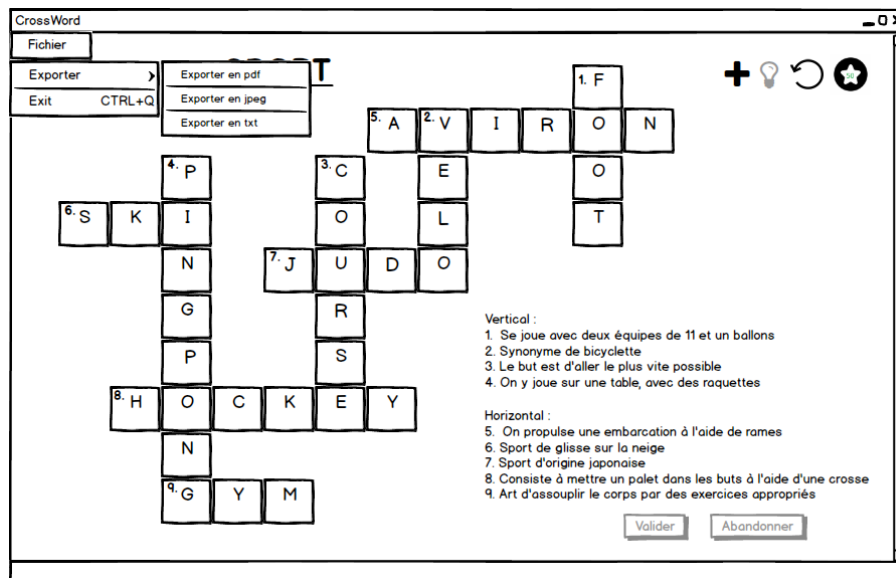


Figure 17: Maquette - Menu export

Format pdf : Le format pdf est mis à disposition en vue d'une impression, la grille est affichée de manière classique tel qu'elle était au moment de l'importation.

Format jpeg : Le format jpeg est mis à disposition en vue d'un partage, la grille est affichée de manière classique tel qu'elle était au moment de l'importation.

Format txt : Le format txt est mis à disposition en vue d'une importation future de la grille. La grille est codée (cf. "Importer une grille" pour la description du codage).

Test associé :

1. *Description et but du test* : Sauvegarde de la grille en cours avec une possibilité de choix de trois extensions différentes : .PDF, .JPG, .TXT .
2. *Cas de test* : Avec l'utilisation d'une grille "témoin", cette dernière doit être sauvegardée sous l'extension choisie par l'utilisateur, à l'emplacement choisi par l'utilisateur grâce à l'ouverture de l'explorer. Cette grille doit pouvoir être rechargée par notre application afin que l'utilisateur puisse poursuivre sa partie.
3. *Déroulement et scénario du test* : L'utilisateur clique sur le "Fichier" puis "Exporter" et il choisi l'extension ".TXT" parmi les trois proposées. L'explorateur de fichier s'ouvre afin que l'utilisateur puisse sélectionner son emplacement de sauvegarde. Une fois la grille sauvegardée, il devra pouvoir la recharger(cf. voir besoin "Importer une grille").
4. *Analyse du test* : Une comparaison de fichier entre une grille témoin sauvegardée en format .TXT par l'utilisateur à un moment défini X du jeu et une grille témoin (en guise de test Oracle) défini préalablement. Les deux fichiers comparés ne doivent montrer aucune différence.

### (C) Choisir une aide

Acteur : Utilisateur.

Résumé : L'utilisateur choisit d'être aidé pour se débloquer.

Préconditions : L'application est correctement (affichage du menu de sélection des paramètres), une partie est commencée et la grille est correctement affichée.

Post conditions : Une aide a été affichée dans la grille.

Enchaînement nominal "Révéler la case" :

Étapes	Utilisateur	Système	Maquette correspondante
1	Selectionne une case		/
2		Prend en compte la case sélectionnée	/
3	Clique sur "Choisir une aide"		/
4		Affiche les différentes aides	Maquette - Aides
	Choisi "Révéler la case"		/
6		Affiche la lettre de la case correspondante	/
7		Actualise les points d'aide de l'utilisateur (cf. explication des points d'aide)	/

Enchaînement nominal "Révéler le mot" :

Étapes	Utilisateur	Système	Maquette correspondante
1	Selectionne une définition		/
2		Prend en compte la définition sélectionnée	/
3	Clique sur "Choisir une aide"		/
4		Affiche les différentes aides	Maquette - Aides
5	Choisi "Révéler le mot"		/
6		Affiche le mot correspondant dans la grille	/
7		Actualise les points d'aide de l'utilisateur (cf. explication des points d'aide)	/



Enchaînement nominal "Vérifier la case" :

Étapes	Utilisateur	Système	Maquette correspondante
1	Sélectionne une case		/
2		Prend en compte la case sélectionnée	/
3	Clique sur "Choisir une aide"		/
4		Affiche les différentes aides	Maquette - Aides
5	Choisi "Vérifier la case"		/
6		Affiche si la lettre est bonne ou mauvaise	/
7		Actualise les points d'aide de l'utilisateur (cf. explication des points d'aide)	/

Enchaînement nominal "Vérifier le mot" :

Étapes	Utilisateur	Système	Maquette correspondante
1	Sélectionne une définition		/
2		Prend en compte la définition sélectionnée	/
3	Clique sur "Choisir une aide"		/
4		Affiche les différentes aides	Maquette - Aides
5	Choisi "Vérifier le mot"		/
6		Affiche si le mot est bon ou mauvais	/
7		Actualise les points d'aide de l'utilisateur (cf. explication des points d'aide)	/

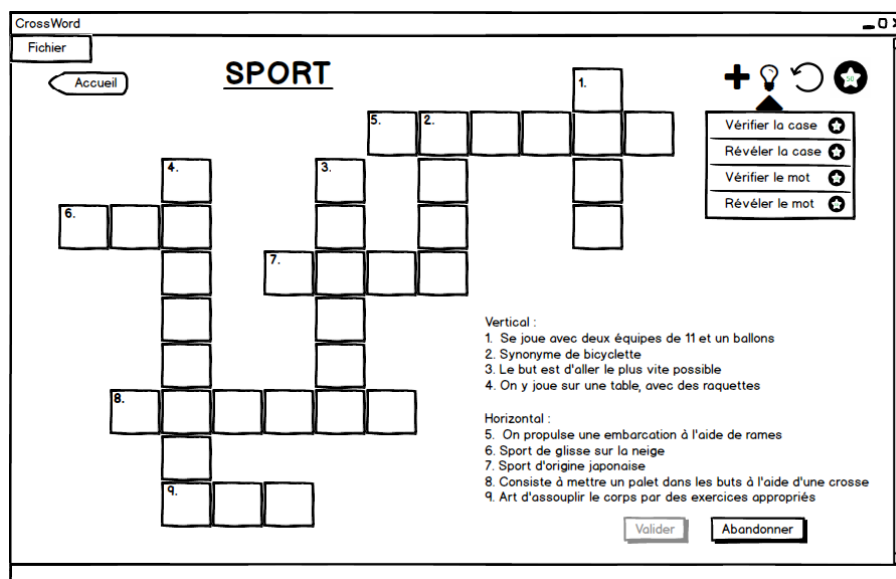


Figure 18: Maquette - Aides

Explication des points d'aide : Lors d'une partie, l'utilisateur aura tant de points (ce nombre de points sera fixé par la difficulté de la grille). A chaque utilisation d'une aide, l'utilisateur perdra des points, selon la "force" de l'aide plus ou moins de points seront enlevés : Vérifier la case < Révéler la case < Vérifier le mot < Révéler le mot. Si l'utilisateur n'a plus de points, il ne peut plus utiliser d'aides.

Tests associés :

1. *Description et but du test* : Bon fonctionnement des options d'aide.
2. *Cas de test* : A partir d'une grille non finie, l'utilisateur doit pouvoir sélectionner les différentes aides proposé par notre programme si son nombre de point le permet.
3. *Déroulement et scénario du test* : L'utilisateur choisi d'utiliser l'aide "révéler le mot", "révéler la case", "vérifier la case", "vérifier le mot".
4. *Analyse du test* : Vérifie que l'aide sélectionnée fonctionne correctement et que les points associés à l'utilisation de cette aide sont bien décomptés aux points restant à l'utilisateur.

1. *Description et but du test* : Utilisation d'une aide avec un nombre de point insuffisant.
2. *Cas de test* : L'utilisateur souhaite utiliser une aide mais n'a pas assez de point. Son utilisation doit être refusée
3. *Déroulement et scénario du test* : Cliquer sur l'utilisation d'une aide avec un nombre de point inferieur à celui demandé pour utiliser l'aide
4. *Analyse du test* : Vérifie que l'aide n'a pas été exécutée.

(C) Dévoiler la grille - abandonner

Acteur : Utilisateur.  
Résumé : L'utilisateur veut voir la solution complète, même s'il n'a pas commencé à remplir la grille.  
Préconditions : L'application est correctement lancée, la grille a été générée et affichée.  
Post conditions : La solution de la grille est affichée, et l'utilisateur n'aura plus la possibilité de valider.

Enchaînement nominal :

Étapes	Utilisateur	Système	Maquette correspondante
1	Clique sur le bouton "Abandonner"		(cf. Maquette - Grille vide, Bouton "Abandonner")
2		Affiche la grille remplie	Maquette - Solution
3		Bloque la validation de la grille	

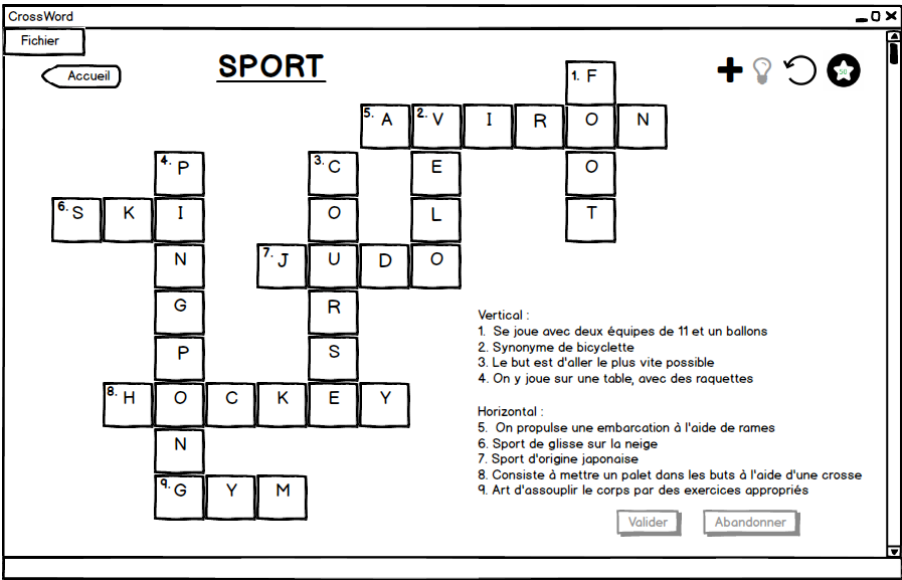


Figure 19: Maquette - Solution

Tests associés :

1. *Description et but du test* : Dévoiler toutes les cases restantes de la grille .
2. *Cas de test* : A partir d'une grille non finie, l'utilisateur doit pouvoir dévoiler l'entièreté de la grille. Chaque case non dévoilée sera remplacée par la lettre associée.
3. *Déroulement et scénario du test* : L'utilisateur clique sur le bouton "Abandonner". Chaque case est remplacée par la lettre associée.
4. *Analyse du test* : Vérifie que toutes les cases de la grille ont bien une lettre associée. Comparaison de la grille dévoilée "temoin" et une grille construite préalablement (test Oracle).

1. *Description et but du test* : Dévoiler la grille sur une grille finie.
2. *Cas de test* : A partir d'une grille finie et correctement complétée, l'utilisateur a toujours accès au bouton "Abandonner".
3. *Déroulement et scénario du test* : L'utilisateur a correctement complété la grille mais le bouton "Abandonner" est toujours visible.
4. *Analyse du test* : Vérifier que le bouton "Abandonner" n'est plus disponible lorsqu'une grille est finie et correctement remplie.

## (E) Valider la grille

Acteur : Utilisateur.

Résumé : L'utilisateur valide sa solution après avoir rempli la grille en entier. Il saura alors combien de mots il a correctement devinés et le nombre d'erreurs effectuées.

Préconditions : L'application est correctement lancée, la grille a été générée, affichée et l'utilisateur l'a remplie.

Post conditions: L'utilisateur a fini, les mots correctement devinés et le nombre d'erreurs sont affichés.

Enchaînement nominal :

Étapes	Utilisateur	Système	Maquette correspondante
1	Clique sur le bouton "Valider"		Maquette - Grille remplie
2		Affiche les mots correctement devinés	Maquette - Validation
3		Affiche le nombre d'erreurs	
4		Met à jour le score	

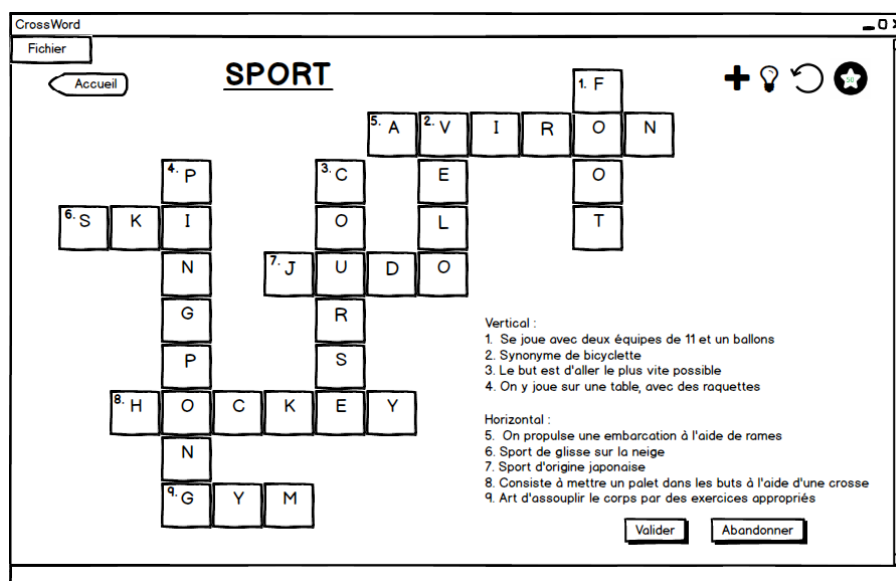


Figure 20: Maquette - Grille remplie

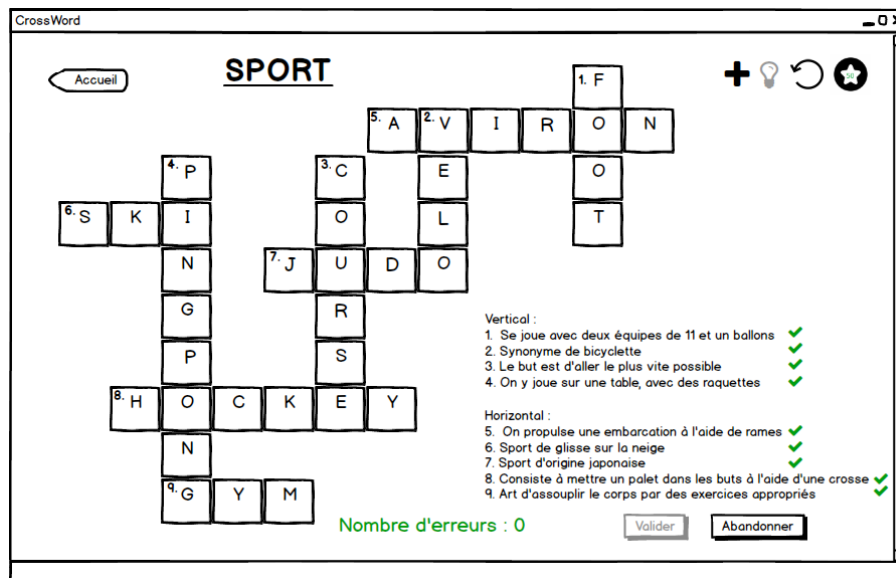


Figure 21: Maquette - Validation

## 4 Besoins non fonctionnels

### 4.1 Besoins d'utilisabilité

#### (E) Interface graphique

Notre grille de mots croisés sera jouable sur une simple interface graphique, disposant de la grille et d'un menu. Cette interface devra être simple d'utilisation.

#### (E) Manuel d'utilisation

Notre programme disposera d'un court manuel d'utilisation afin de préciser les différents points sur l'utilisation de notre grille de mots croisés et sur l'utilisation des différents menus de notre application.

#### (E) Langue du programme

Tout le programme (interface, mots de la grille et menu) devra être écrit en Français.

### 4.2 Besoins de performances

#### (E) Temps de génération de la grille de mots croisés

Toute génération de grille de mots croisés (quelque soit les paramètres choisis par l'utilisateur) doit se faire en moins de quatre secondes. Une attention particulière sur la complexité de l'algorithme de génération ainsi que sur les tests de génération de la grille devra être apportée.

### 4.3 Besoins de sécurité

#### (E) Sécurité

L'utilisation de notre programme ne requiert aucune connexion internet, de ce fait aucune donnée personnelle propre à l'utilisateur ne sera sauvegardée ou utilisée.

## References

- [1] Xinguang Chen Adam Beacham and al. “Crossword Puzzles as a Constraint Problem”. In: 2008. URL: <https://dept-info.labri.fr/~narbel/PdP/Subjects21-22/Crosswords/CP08.pdf>.
- [2] Ámbulagan and Adi Botea. “Crossword Puzzles as a Constraint Problem”. In: 2008. URL: <https://dept-info.labri.fr/~narbel/PdP/Subjects21-22/Crosswords/CP08.pdf>.
- [3] Aragon-Technologies. *Mot croises online*. <https://www.mots-croises-online.com/fr/motscroises/logiciel.php>. Accessed: 11-03-2022. Last update in 2012.
- [4] The Teacher’s Corner. *Crossword Puzzle Maker*. <https://worksheets.theteacherscorner.net/make-your-own/crossword/lang-fr/>. Accessed: 11-03-2022. Last update in 2022.
- [5] Green Eclipse. *EclipseCrossword*. <https://www.microsoft.com/en-us/p/eclipsecrossword/9wzdnrfj9kp#activetab=pivot:overviewtab>. Accessed: 11-03-2022. Last update in 2016.
- [6] Education.com. *Crossword Puzzle*. <https://www.education.com/worksheet-generator/reading/crossword-puzzle/>. Accessed: 11-03-2022. Last update in 2022.
- [7] Matthew L. Ginsberg. “Crosswords and an Implemented Solver for Singly Weighted CSPs”. In: *Journal of Artificial Intelligence Research* 42 (2011), pp. 851–886. URL: <https://dept-info.labri.fr/~narbel/PdP/Subjects21-22/Crosswords/1401.4597.pdf>.
- [8] John Duchi James Connor and Bruce Lo. “Crossword Puzzles and Constraint Satisfaction”. In: 2005. URL: [https://dept-info.labri.fr/~narbel/PdP/Subjects21-22/Crosswords/crossword\\_writeup.pdf](https://dept-info.labri.fr/~narbel/PdP/Subjects21-22/Crosswords/crossword_writeup.pdf).
- [9] *Mots croisés - Sport Cérébral*. <https://www.sportcerebral.com/mots-croises-gratuit>. Accessed: 03-02-2022. Last update in 2022.
- [10] Puzzel.org. *Crossword Create*. <https://puzzel.org/fr/crossword/create>. Accessed: 11-03-2022. Last update in 2022.