**DOCUMENTATION DU SYSTÈME**

1. **DESCRIPTION BRÈVE DE LA PROBLÉMATIQUE**

Programmation d’un jeu \*Bonhomme pendue\*, être capable d’utiliser les notions sur les fichiers apprises en classe.

1. **CONCEPTION DE LA SOLUTION**
   1. **DESCRIPTION DES DONNÉES : Résultats à obtenir**

*Description des résultats à obtenir. Identificateur, type, description, exemple si nécessaire*

|  |  |  |  |
| --- | --- | --- | --- |
| **Identificateur** | **Type** | **Description** | **Exemple** |
| chosenWord | String | Mot choisis aléatoirement | « cow » |
| letterInput | Char | Lettre entrer par utilisateur | « e » |
| wordArray | Char[] | Opérations arithmétique | {‘c’, ‘o’, ‘w’} |

* 1. **DESCRIPTION DES DONNÉES : Existantes**

**Constantes:**

|  |  |  |  |
| --- | --- | --- | --- |
| 4 difficultés | Int | Switch case(menu contextual) | /0/ |
| 1 mot à trouver | String | 1 mot par partie à trouver | 2<7 |

**Données d’entrée:**

|  |  |  |  |
| --- | --- | --- | --- |
| Input / Scanner | String / Char | Peut-être un string ou un char  Dépend du besoin du prog | /0/ |
| myFile | File | Path du fichier en interaciton | File Path |

**Données intermédiaires:**

|  |  |  |  |
| --- | --- | --- | --- |
| CountX | Int | Utiliser pour tenir le compte des erreurs d’entré | /0/ |
| String[] hP | String[] | Print hangman | {‘’…’’} |

**JEU DE TESTS**

Tous les Menu (Ils ont tous le même code):

-Entrer un String/Char = pas de Crash, Déclare l’erreur.  
-Entrer un Integer qui n’est pas une option = pas de Crash, Déclare l’erreur.  
-Rien entrer = pas de Crash, ne déclare pas l’erreur.

Options 1 – New game.

-Entrer un Integer/boolean/float/double = pas de Crash, ne déclare pas l’erreur, retire un essai  
-Rien entrer = pas de Crash, ne déclare pas l’erreur.

Options 2 et 3 – Add Word / Delete Word.

-Entrer une valeur autre que ceux attendu = pas de Crash, déclare l’erreur et retour au menu + Ajoute le mot au répertoire des mots non-valider.

Options 4 – See Word

-Agis comme les autres menu contextuel.