# 

**Sciences Informatiques et Mathématiques**

# **PROJET D’INTÉGRATION- ITÉRATION (I)**

## ***Titre du travail (Projet-version-i)***

## **Professeure** : Niar Wafaa

## **Cours** : 420-204-RE

## **Fait par** : *noms, prénoms*

## **Session** **: Hiver ……**

1. **INTRODUCTION :**
   1. **Objectif du document**

Le but de ce rapport est de pouvoir expliquer comment nous allons concevoir notre projet intégrateur pour sa première version. Le rapport est destiné à un public qui s’intéresse à l’informatique et plus précisément à la programmation de logiciels qui utilisent des concepts scientifiques.

* 1. **Portée du produit logiciel**

Le logiciel se nomme *Super Billard*. Il permet de jouer au billard en ajoutant la possibilité de modifier certains paramètres scientifiques. Ces paramètres pourront être sauvegardés même lorsque l’on quitte le jeu. Il ne permet pas de jouer à plusieurs et de modifier la taille de la table de billard. Ce produit s’intègre dans un contexte où les jeux utilisent des concepts scientifiques, surtout physiques. Le logiciel a pour objectif de permettre à ses utilisateurs de pouvoir voir concrètement les effets d’un certain concept scientifique. Le principal bénéfice de cette application est qu’il s’agit d’une représentation concrète et graphique de plusieurs concepts scientifiques. Les retombées du logiciel sont les gens vont pouvoir voir que les concepts scientifiques s’appliquent dans des activités de tous les jours. Le programme va permettre de modifier le frottement de la table, la masse des balles, la vitesse des balles et l’angle de tir. L’application utilise les concepts de la cinétique et du mouvement.

* 1. **Définitions, acronymes et abréviations**

*Définir tous les termes non courants, les acronymes, et les abréviations nécessaires à la compréhension de ce document.*

* 1. **Documents de référence**

*Citer tous les documents utilisés et référencés dans celui-ci. Préciser les titres, noms d’auteurs, etc.., et également toute autre source d’information.*

*Mentionner ICI les documents de références pour les normes ergonomiques, les normes de programmation, etc.*

ORACLE ET SUN MICROSYSTEMS. API Java SE 8. [En ligne]. https://docs.oracle.com/javase/8/docs/api/ [1er mars 2017].

ORACLE ET SUN MICROSYSTEMS. API JavaFX 8. [En ligne]. https://docs.oracle.com/javase/8/javafx/api/toc.htm [1er mars 2017].

SPOLSKY, Joel et ATWOOD, Jeff. StackOverflow. [En ligne]. http://stackoverflow.com/ [1er mars 2017].

* 1. **Planification des tâches**

*Mettre à jour le plan de projet MSProject :écrire l’organisation et le partage des tâches entre les membres de l’équipe dans chaque étape de développement, en mentionnant les dates, la durée et les intervenants (membres de l’équipe) avec le rôle de chacun.*

*Nommer et décrire les sous-tâches pour chaque étape, par exemple si certaines parties de la tâche ont été faites par différents membres de l’équipe., rajouter colonnes au besoin.*

*OU RENVOYER VERS LE DOCUMENT « PLAN PROJET » S’IL Y A LIEU*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tâches** | **Date début** | **Date fin** | **Durée (nbre heures)** | **Intervenants et rôles** | **Livrables (documents, code,..)** |
| 1. **Analyse**   **Sous-tâches :**  **-**  **-**  **-** |  |  |  |  | Cas d’utilisations |
| 1. **Conception**   **Sous-tâches :**  **-**  **-**  **-** |  |  |  |  | Diagramme de classes |
| 1. **Implémentation**   **Sous-tâches :**  **-**  **-**  **-** |  |  |  |  | Codes sources |
| 1. **Tests, mise au point et gestion exceptions**   **Sous-tâches :**  **-**  **-**  **-** |  |  |  |  |  |
| 1. **Intégration**   **Sous-tâches :**  **-**  **-**  **-** |  |  |  |  |  |
| 1. **Rédaction du rapport**   **Sous-tâches :**  **-**  **-**  **-** |  |  |  |  |  |

1. **ANALYSE :**
   1. **Perspective du produit**

*Présenter une description détaillée des fonctionnalités du logiciel à développer et de son environnement.*

***Important : Expliquer ici ce qui va être réalisé dans cette version, et un apercu de ce qui reste à réaliser dans les prochaines versions.***

* 1. **Caractéristiques des utilisateurs**

*Décrire les caractéristiques générales des utilisateurs potentiels du logiciel, en précisant leur niveau de formation, d’expertise et d’expérience.*

* 1. **Modèle environnemental : les cas d’utilisation**
     1. **Diagramme de cas d’utilisation :**

*Dresser le diagramme de cas d’utilisation en visualisant : les acteurs, la limite du système et les cas d’utilisation. IMPORTANT : NE DÉCRIRE QUE CE QUI EST FAIT DANS CETTE VERSION DE L’APPLICATION*

***Exemple :***

Gestion dépanneur

Client commis

Fournisseur gérant

*(Acteurs)*

* + 1. **Description narrative des cas d’utilisation**

*Détailler les cas d’utilisation ici, sous la forme suivante :*

***Cas d’utilisation***: *nom (exemple : Gérer stock)*

***Acteurs*** : *préciser les acteurs impliqués*

***But***: *but du cas d’utilisation*

***Description***: *description du cas d’utilisation sous forme de tableau comme suit :*

***Cours typique d’évènements :***

|  |  |
| --- | --- |
| **Actions Acteur** | **Réponses système** |
| 1. *L’utilisateur initie en demandant* … | 1. *Le système répond en…..* |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

***Cours alternatifs :*** *préciser les options alternatives et leur numéro de ligne du tableau*

*Ligne xx : ……….*

* + 1. **Prototypes d’interfaces :**

*Insérer des dessins d’interfaces graphiques, qui constituent une base sur laquelle ceux-ci seront conçus. Donner une légende à chaque dessin d’écran.*

* 1. **Le modèle structural :**

*Insérer le diagramme de classes du domaine. Attention, ce n’est pas le diagramme des classes logicielles. Il est moins détaillé : juste des noms de classes et les liens, aucune classe technique provenant de Java par exemple, juste les classes identifiées dans l’énoncé du problème, concrètes ou abstraites.*

Si le diagramme est fait sous VISIO, vous pouvez l’insérer dans ce document même avec le menu : Insertion/Objet (dans le groupe Texte) dans la boite affichée, choisir l’onglet A partir du fichier, puis parcourir pour choisir le fichier Visio qui contient votre dessin.

1. **CONCEPTION**
   1. **Diagramme des classes logicielles**

*Insérer le diagramme de classes logicielles. Ce diagramme est plus détaillé car il fait apparaitre toutes les classes de l’application ainsi que les relations. IMPORTANT : NE DÉCRIRE QUE CE QUI EST FAIT DANS CETTE VERSION DE L’APPLICATION*

* 1. **Classes modèle**

Nos classes modèles sont Balle, Table et TrouDeTable*.*

1. **IMPLÉMENTATION ET TESTS**

**6.1. Choix technologiques**

Plateforme utilisée : Microsoft Windows 7 Professionnel, Microsoft Windows 10 Famille et Ubuntu 16.04.2 LTS

Langage de programmation utilisé: JavaFX 2.2.75, Java SE 1.7.0, HTML 5, CSS 3 et JavaScript (Pour l’aide en ligne)

IDE utilisé : Netbeans 8.1

Autres outils : MS Project, Dia, Github, StackOverflow, API Java SE 8, API JavaFX 8

**6.2. Codage**

*Préciser ici où se trouvent les fichiers de l’application (nommer le chemin et dossier de remise dans lequel le projet sera déposé.)*

À MODIFIER JUSTE AVANT LA REMISE

**6.2. Tests**

Tests effectués :

-Test de la force de la balle (Données utilisées : 1, 2 et 3)

-Test de la vitesse de la balle (Données utilisées : 1 à 10)

-Test de l’angle de la balle (Données utilisées : 0 à 360)

-Test de la position de la table (Données utilisées : 0 à 20)

-Test de la grandeur de la table (Données utilisées : 500, 1000, 325, 650)

-Test de la position des trous sur la table (Données utilisées : 0 à 20(puis en fonction de la grandeur de la table))

-Test de la grandeur des trous (Données utilisées : 10 à 30)

Critères de tests :

-Le programme doit rester fluide

-Le programme doit être simple d’utilisation

-Le programme doit faire ce que l’on attend de lui

-Tous les éléments graphiques doivent être présents et cohérents (Pas plein de couleurs différentes)

Bogues rencontrés :

-Lorsque la balle touche un coin, elle se déplace uniquement de gauche à droite ou de haut en bas

-La balle disparait et se divise en deux à chaque secondes

-La balle accélère lorsqu’elle devrait décélérer

-Les trous de la table n’apparaissent pas

-Lorsqu’on entre une vitesse trop grande la balle ne bouge pas

Grille d’évaluation

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Équipe no :  Noms |  | | | | |
|  | Très satisfaisant | Satisfaisant | Assez satisfaisant | Peu satisfaisant | Insatisfaisant |
| Rapports :  Planification réaliste  Diagrammes précis  Respect normes UML | 25 | 20 | 15 | 5 | 0 |
| Codage :  Concepts orienté objet (encapsulation,héritage,polymorphisme)  Organisation classes (MVC)  Réutilisation | 25 | 20 | 15 | 5 | 0 |
| Convivialité :  Normes ergonomiques  Aide en ligne\*  Fichier jar\* | 15 | 12 | 8 | 4 | 0 |
| Validations :  Absence des buggs  Gestion exceptions  Traitement erreurs | 15 | 12 | 8 | 4 | 0 |
| Qualité du code :  Normes programmation Documentation javadoc | 15 | 12 | 8 | 4 | 0 |
| Qualité français :  Rapport  Interfaces graphiques | 5 | 4 | 3 | 1 | 0 |
| TOTAL | % | | | | |

\* : *pourraient ne pas être exigées dans les 1ères versions.*