# 

**Sciences Informatiques et Mathématiques**

# **PROJET D’INTÉGRATION- ITÉRATION (1)**

***Super Billard version 2***

## **Professeure** : Niar Wafaa

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

## **Fait par** : LACHANCE, Marc-Antoine, PASSUELLO-DUSSAULT, aaaaaaaaaaaaaaVittorio, RAFFIS, Édouard, RONDEAU, Victor

## **Session** **: Hiver 2017**

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

Le but de ce rapport est d’expliquer comment nous allons concevoir notre projet intégrateur, fait dans le cadre du cours Projet Intégrateur du programme de Sciences, Informatiques et Mathématiques. Le rapport est destiné à la professeure de ce cours, Wafaa Niar et à toute l’équipe qui travaille sur le projet Super Billard.

* 1. **Portée du produit logiciel**

Le logiciel se nommeSuperBillard. 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 que les gens vont pouvoir prendre connaissance 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**

Rien à signaler

* 1. **Documents de référence**

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**

Document MS Project. (En annexe)

Chemin : Dans la remise sur Moodle. Dans le dossier Remise-SuperBillard

MANQUE NOMS DES ÉTUDIANTS ASSOCIÉS À LEUR RÔLE

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

Dans cette première version, l’utilisateur sera capable de choisir l’angle et la force de frappe d’un tir et d’effectuer le tir. Il sera aussi possible de frapper les balles dans les trous pour les faire disparaître. Dans les prochaines versions les options de masses et frottement seront disponible. Il y aura aussi l’ajout des règlements complets du billard ainsi que l’addition d’une intelligence artificielle pour pouvoir jouer des parties de 8ball pool complètes.

Dans cette deuxième version, l’utilisateur sera capable de choisir l’angle et la force de frappe d’un tir et d’effectuer le tir. De plus, les balles (15) auront des collisions entre elles. Il sera aussi possible, dans la prochaine version, de frapper les balles dans les trous pour les faire disparaître. Dans les prochaines versions les options de masses et frottement seront disponible. Il y aura aussi l’ajout des règlements complets du billard ainsi que l’addition d’une intelligence artificielle pour pouvoir jouer des parties de 8 ball pool complètes.

Dans cette troisième version, il y a eu des changements sur les collisions ainsi que sur le graphisme. L’utilisateur sera donc encore capable de choisir l’angle et la force de ses tirs. De plus, lorsque le joueur rentre une balle dans un trou cette dernière vient se placer dans un conteneur. Il est maintenant possible de régler le niveau de la musique. Dans la prochaine version il y aura l’ajout d’options pour modifier la masse des balles et le frottement de la table ainsi que l’ajout d’une gestion pour effectuer des parties.

* 1. **Caractéristiques des utilisateurs**

Ce programme est destiné aux utilisateurs de tous les niveaux. Tant bien aux jeunes gens qu’aux adeptes de billard. Les utilisateurs n’auront pas besoin d’avoir de formation et n’importe qui devrait être en mesure de jouer à Super Billard. Pour avoir une expérience de jeu plaisante il est recommandé de connaître les bases de l’ordinateur, de la souris et du clavier.

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



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

***Cas d’utilisation***: Joueurau jeu

***Acteurs*** : Joueur

***But***: Démarrer et jouer une partie de billard

***Description***:

***Cours typique d’évènements :***

|  |  |
| --- | --- |
| **Actions Acteur** | **Réponses système** |
| 1. L’utilisateur initie en cliquant sur le bouton Jouer. | 1. Le système répond en affichant la table de billard et les champs de paramètres. (Vitesse, Angle) |
| 1. L’utilisateur entre les données de force et vitesses et clique sur le bouton lancer. | 1. Le système met la balle en mouvement. |
| 5- La balle cesse de rouler. | 5- Le système demande à l’utilisateur d’inscrire de nouvelles données. |

Cours alternatifs :

Ligne 4 : un des champs est vide : le système indique une erreur.

***Cas d’utilisation***: Obtenir de l’aide

***Acteurs*** : Joueur

***But***: Apprendre comment jouer

***Description***:

***Cours typique d’évènements :***

|  |  |
| --- | --- |
| **Actions Acteur** | **Réponses système** |
| 1. L’utilisateur initie en cliquant sur le bouton Aide | 1. Le système répond en affichant la fenêtre d’aide (Règles et explication des paramètres) |
| 3- L’utilisateur clique sur le lien des règlements complet | 4- Le système ouvre le site dans une fenêtre d’Internet |
| 5- L’utilisateur clique sur le lien du site de super billard. | 5- Le système ouvre le site dans une fenêtre d’Internet |
| 5- L’utilisateur clique sur le bouton menu | 6- Le système quitte le menu aide et retourne au menu principal |

***Cas d’utilisation***: Configurer le jeu

***Acteurs*** : Joueur

***But***: Changer les options

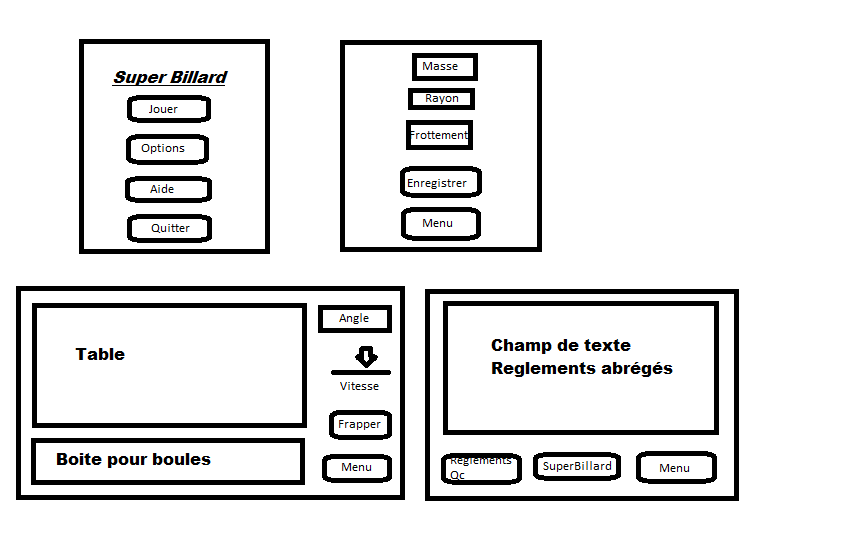
***Description***:

***Cours typique d’évènements :***

|  |  |
| --- | --- |
| **Actions Acteur** | **Réponses système** |
| 1. L’utilisateur initie en cliquant sur le bouton Options | 1. Le système répond en affichant la fenêtre d’options (Frottement, musique, masse) |
| 3- L'utilisateur entre les données dans les champs de texte (Rayon, Masse, Frottement) et appuie sur le bouton | 4- Le système change les paramètres du jeu et affiche un message disant que l’action a été complétée. (Version 2 : Affiche les paramètres mais ne les changes pas.) |
| 5- L’utilisateur clique sur le bouton menu. | 6-Le système quitte le menu options et retourne au menu principal |

À noter : Les changements de paramètre ne sera pas implémenté dans la version trois, les paramètres seront seulement affichés dans la console.

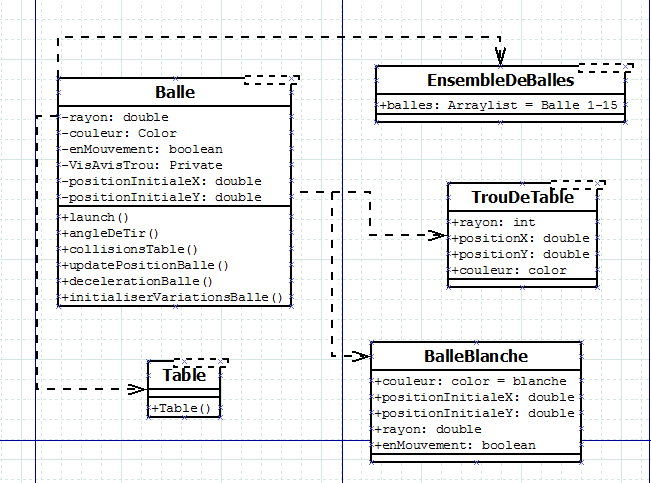
* + 1. **Prototypes d’interfaces :**



* 1. **Le modèle structural :**

**

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



METTRE TOUTES LES CLASSES DANS CE DIAGRAMME

Classe menus :

-menuJouer

-menuOptions

-menuAide

-menuPrincipal

Classes modèles :

-Table

-EnsembleDeBalles

-TrouDeTable

-Balle

-BalleBlanche

-BalleHuit

1. **IMPLÉMENTATION ET TESTS**

**4.1. 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

**4.2. Codage**

Les fichiers code sources sont dans l’archive Remise-SuperBillarv2, dans le dossier Codes Sources. Le tout se trouve dans la remise sur Moodle.

ATTENTION AUX LIGNES VIDES

**4.3 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

*OK*

*ENLEVER LES REMARQUES PROF DE LA VERSION 1!*

*~~REMARQUES POUS CODES SOURCES :~~*

* *~~ATTENTION , ÉVITER LES CARACTERES ACCENTUÉS (COMME LE NOM DU PACKAGE NETBEANS)~~*
* *~~NE PAS UTILISER DE CLASSE VUE DANS UNE CLASSE MODÈLE, PAR EXEMPLE VOUS UTILISEZ LA CLASSE MENUJOUER DANS LA MÉTHODE LANCH() DE LA CLASSE BALLE, , CE N’EST PAS PERMIS DANS ARCHITECTURE MVC, IL FAUT TROUVER UN MOYEN DE DÉFINIR UNE CLASSE CONTRÔLEUR QUI FERAIT LE LIEN ENTRE LES CLASSES MODÈLE ET LES CLASSES VUE.~~*
* *~~ENCAPSULATION DOIT ÊTRE COMPLÈTE DANS LES CLASSES MODÈLE~~*
* *~~INTÉGRER HÉRITAGE~~*
* *~~IL EST PRÉFÉRABLE D’UTILISER UN TABLEAU OU UNE COLLECTION DANS LA CLASSE BALLE AU LIEU DES ATTRIBUTS BALLE1 , BALLE2, ,…. BALLE15~~*
* *~~IL MANQUE DES COMMENTAIRES, EXPLIQUER CHAQUE CLASSE UTILISÉE ET CHAQUE MÉTHODE~~*

**Grille d’évaluation**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Équipe no : 2  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 | 87 % | | | | |

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

REMARQUES VERSION 2 :

* ANGLE : PRÉCISER LES UNITÉS OU METTRE UN SLIDER ENTRE UN MIN ET UN MAX
* VITESSE : PRÉCISER UNITÉ ET VALEURS MIN ET MAX (PAS BESOIN D,ALLER DANS L’AIDE POUR AVOIR CETTE PRÉCISION
* AJOUTER UNE OPTION POUR RÉINITIALISER LE JEU ?
* MENU PRINCIPAL, LE BOUTON « OPTION » PORTE À CONFUSION, L’APPELER PLUTÔT « CONFIGURER JEU » OU « PARAMÈTRES D’AFFICHAGE »
* NE PAS UTILISER DE MOTS TECHNIQUES DANS L’AIDE(COMME SLIDER) ET FAIRE ATTENTION AUX FAUTES DE FRANÇAIS ET AUX ACCENTS ABSENTS
* CLASSE BALLE, LA MÉTHODE SUIVANTE PEUT ÊTRE REMPLACÉE SIMPLEMENT PAR UNE CONSTANTE : FINAL RAYON\_BALLE\_AMERICAINE=0.0286

public double rayonBalleAmericaine(){

return 0.0286;

}

SI BIEN SUR CA NE CHANGERA JAMAIS

* EN PRINCIPE ON NE DEVRAIT PAS AVOIR DE CLASSES GRAPHIQUES DANS LES CLASSES MODÈLES (TABLE ET TROUDETABLE), ON PEUT SOIT TROUVER UN MOYEN DE SÉPARER LE CODE DU GRAPHISME, SI ON NE PEUT PAS, ALORS IL FAUT CONSIDÉRER LA CLASSE COMME CONTRÔLEUR ET NON MODÈLE.