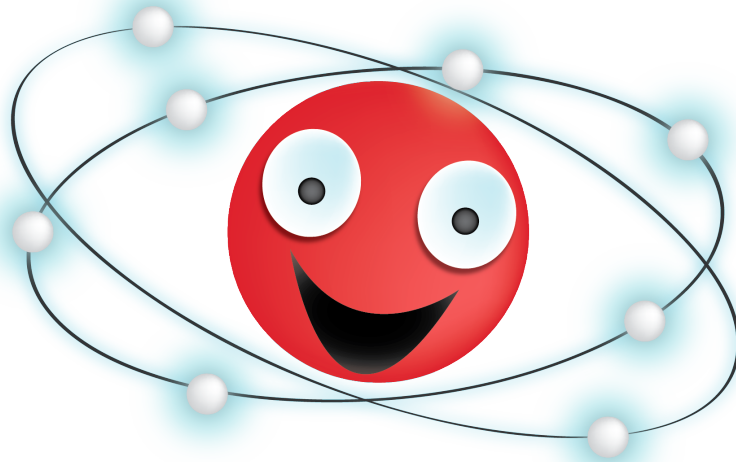


# CHEMICALL

RAPPORT DE PROJET  
SONS ET MUSIQUES

KEVIN BOLLINI, THOMAS HINSINGER, GEOFFREY MÉLIA



MASTER INFORMATIQUE  
ANNÉE UNIVERSITAIRE 2012/2013

# Table des matières

Remerciements . . . . .	5
<b>I Rapport de projet</b>	<b>6</b>
<b>1 Introduction</b>	<b>7</b>
<b>2 Le Serious Game</b>	<b>8</b>
2.1 Idée générale . . . . .	8
2.2 Contexte . . . . .	8
2.3 Gameplay . . . . .	9
2.3.1 Les contrôles . . . . .	9
2.3.2 Acteurs . . . . .	9
2.3.3 Les feedbacks . . . . .	9
<b>3 Approche Artistique</b>	<b>11</b>
3.1 Sons . . . . .	11
3.1.1 Interface . . . . .	11
3.1.2 Gameplay . . . . .	11
3.2 Musiques . . . . .	11
3.2.1 Contexte . . . . .	11
3.2.2 Démarche artistique de création . . . . .	12
3.2.3 Musicalité . . . . .	12
<b>4 Approche Technique</b>	<b>13</b>
4.1 Outils utilisés . . . . .	13
4.2 FMOD Designer . . . . .	14
4.2.1 Sons . . . . .	14
4.2.2 Musiques . . . . .	15
4.3 Sound Manager . . . . .	16
4.3.1 Rôle . . . . .	16
4.3.2 Fonctionnalités . . . . .	16
<b>5 Discussion</b>	<b>17</b>
5.1 Difficultés Rencontrées . . . . .	17
5.2 Apports personnels . . . . .	17
<b>6 Conclusion</b>	<b>19</b>
6.1 Conclusion . . . . .	19
6.2 Perspectives . . . . .	19
<b>7 Références</b>	<b>20</b>
<b>II Annexes</b>	<b>21</b>
<b>A Diagramme de Classes</b>	<b>22</b>



# Table des figures

2.1	Anion	9
2.2	Cation	9
2.3	Barre de vie	10
2.4	Info-bulle	10
4.1	Exemple de configuration de sound effect	14
4.2	Exemple de configuration d'événement musical	15
4.3	Aperçu de la classe SoundManager	16
A.1	Aperçu de l'application	22

## Remerciements

### **Pour la partie Sons et musiques :**

- Sylvain Daudé pour ses cours et conseils avisés en matière de Sound Design.
- Nastassia Jacquet pour son aide précieuse en matière de sound design et de bruitage.
- Frédéric Tari, pour avoir partagé ses connaissances et son affinité pour la musique.

### **Pour la partie Serious Game :**

- Damien Djaouti, pour ses conseils avisés en matière de Game Design.
- Abdelkader Gouaich pour ses cours et pipeline de développement.
- Camille Seilles, pour ses réalisations graphiques de notre avatar oxygène sous ses différentes formes.

**Première partie**

**Rapport de projet**

# Chapitre 1

## Introduction

Dans le cadre du Master 2 informatique, nous avons été amenés à concevoir une application en y intégrant musiques d'ambiances, bruitages et autres sons pour donner vie et donner plus de feedback à cette application. De plus et lors de ce même Master 2, il nous aussi été demandé de prototyper un Serious Game, jeu pour lequel nous avons choisi le thème de la chimie comme contexte.

Nous avons dès le début de ce Semestre 1 de Master 2 IMAGINA, décidé d'exploiter notre Serious Game pour y intégrer du Son et de la Musique pour donner plus de vie à notre Serious Games. Nos deux projets sont donc intimement lié et c'est pourquoi, avant même de parler des technologies utilisés, voir des musiques/bruitages créés nous allons dans un premier temps vous expliquer en quoi consiste ce Serious Game. Comme nous vous l'avons dit, dans un deuxième temps, nous expliquerons les technologies exploités pour intégrer à ce jeu des Sons et des Musiques. Enfin dans un dernier temps, nous développerons en détails les différents bruitages, et musiques intégrer à ce Serious Game.

## Chapitre 2

# Le Serious Game

### 2.1 Idée générale

Le joueur incarne un atome d'Oxygène qui doit délivrer d'autres atomes, lui permettant par là même d'augmenter ses capacités grâce aux interactions chimiques qu'il pourra alors réaliser. Dans sa quête, il rencontrera des entités ayant un électrons en moins ou en plus et qui lui feront perdre ou gagner des électrons : les ions. Il devra sortir du niveau en étant stable afin de gagner un nouvel atome. Dès lors, le joueur apprendra à utiliser ce nouvel atome avec lui-même afin de créer une molécule ayant des propriétés particulières. Il pourra alors appeler un de ces atomes gagnés, lors de prochains niveaux afin de modifier son état physique et franchir les obstacles.

### 2.2 Contexte

Ce jeu a été pensé et conçu pour les enfants âgés de 10 à 14 ans dont la scolarité amène à apprendre les premières notions de chimie. Le choix de ces notions dans le contenu pédagogique de notre jeu a donc été réalisé en fonction des connaissances chimiques attendues d'un enfant de cet âge.

Nous avons dans un premier temps étudié les notions importantes à mettre en valeur dans notre jeu puis, dans un second temps, la manière dont ces notions pouvaient être intégrées au jeu afin de suivre une visée pédagogique.

#### Éléments de chimie

L'atome est un élément essentiel de la chimie et en compose la base : nous avons choisi de l'utiliser comme l'élément central du jeu. Notre objectif étant de faire apprendre :

- les notions d'interactions entre atomes ainsi que l'état physique résultant.
- Les relations entre électrons et atomes
- Les différents états de l'atome en fonction de son nombre d'électrons.

Au vue des programmes scolaires de chimie, le premier élément complexe, ou molécule, appris est l'eau. Nous avons donc penser utiliser les atomes le composant : l'oxygène ou l'hydrogène. Étant donné que l'hydrogène n'a que deux électrons et que pour notre idée de jeu, ceci était trop faible, nous avons décidé de garder l'atome oxygène.

De plus, l'oxygène est un élément simple qui s'associe facilement avec d'autres atomes, ce qui était intéressant pour notre gameplay. Nous pouvons facilement créer les molécules simples comme le dioxygène et l'eau.

#### Intégration

Après avoir réfléchi aux éléments et aux notions de chimie que nous souhaitions intégrer, nous nous sommes intéressés à la manière la plus adéquate pour fonder dans le jeu ces différents éléments.

L'atome étant l'élément central du serious game, nous avons décidé de l'utiliser comme personnage principal. Afin de proposer un challenge au joueur, nous voulions intégrer des personnages ennemis ou alliés dans le jeu. Étant donné que nous souhaitions introduire la notion d'électrons, l'idée d'anion et de cations nous apparut rapidement comme la solution la plus adéquate.

Pour l'association avec d'autres atomes, nous avons pensé à plusieurs possibilités telles que les intégrer directement dans le niveau comme un pouvoir disponible par exemple. Mais ceci ne nous apportait pas de solutions permettant de proposer

un challenge intéressant pour finir le niveau. Nous avons donc décidé de l'intégrer à la fin de la réussite d'un niveau et de pouvoir expliquer l'intérêt d'utiliser cet atome.

## 2.3 Gameplay

Il est important de préciser que notre Serious Game est un Puzzle/Plateforme Game dans un environnement 2D. Nous allons ici vous présenter comment se joue ce jeu.

### 2.3.1 Les contrôles

Il existe un certain nombre d'actions que le joueur peut faire durant une partie. Précisons tout de même que le jeu se joue exclusivement au clavier. Voici la liste des actions que le joueur pendant le jeu :

- Se déplacer (gauche, droite)
- Sauter
- Utiliser une habilité
- Changer d'habilité
- Ouvrir le menu

Il faut noter cependant que lors de la création de ce Serious Games, il était demandé de créer un prototype de celui-ci et non un jeu entièrement terminé. Dans ce prototype, il n'y a pas encore la transformation en molécule et donc l'utilisation du système d'habilités.

### 2.3.2 Acteurs

Il existe dans ce jeu différents acteurs pouvant interagir avec le personnage que le joueur déplacera tout au long de son aventure. Comme expliqué dans la partie précédente, il existe dans son aventure des ions qui peuvent donner ou perdre des électrons influençant sur la vie du personnage contrôlé par le joueur. Ils sont au nombre de deux :

**Anions :** entité possédant un électron en trop. Si le joueur entre en contact avec cette entité il se verra ôté d'un électron. Voici à quoi il ressemble dans le jeu :



FIGURE 2.1 – Anion

**Cations :** entité possédant un électron en moins. Si le joueur entre en contact avec cette entité, il se verra augmenter d'un électron.

Voici à quoi il ressemble dans le jeu :



FIGURE 2.2 – Cation

### 2.3.3 Les feedbacks

Les feedbacks éléments essentiels dans un jeu-vidéo et d'autant plus dans un Serious Game permettent au joueur de comprendre ce qui se passe et d'avoir un réel ressenti au cours de son aventure. Nous allons ici en lister quelques uns :

**La barre de vie :** cette barre de vie, située en haut à gauche de l'écran, indique au joueur l'état actuel de sa vie. S'il possède 6 électrons, cette barre est "verte" et indique au joueur qu'il est stable et que tout va bien. Cependant lorsque qu'il rentrera en contact avec un cation ou un anion, celle-ci se mettra à jour et le joueur saura ainsi toujours où il en est.



FIGURE 2.3 – Barre de vie

**Les info-bulles :** ces info-bulles permettent d'expliquer au joueur ce qui lui arrive. En effet, rappelons que ce jeu est destiné à des enfants qui débutent dans la chimie, il est intéressant de leur apprendre chimiquement ce qu'il se passe lorsque leur avatar d'atome entre en contact avec un anion ou un cation.



FIGURE 2.4 – Info-bulle

**Les sons et les musiques :** évidemment ces feedbacks sont très importants, et c'est un point que nous approfondirons dans le chapitre 3.

# Chapitre 3

## Approche Artistique

### 3.1 Sons

#### 3.1.1 Interface

Le premier aspect visible d'un jeu vidéo est son interface, qu'elle soit inhérente au jeu lui-même ou aux éléments qui lui sont rattachés comme les menus. Cependant, aussi important soit son caractère visuel, l'interface ne devient vraiment parlante que lorsqu'elle intègre un aspect sonore. Ce dernier renforce de manière décisive le feedback de l'utilisateur. Notre premier travail fut donc d'identifier tous les moments clés durant lesquels un bruitage se révélait nécessaire. On retrouve ainsi la navigation parmi les différents éléments d'un menu ou la sélection de l'un d'entre eux, les messages d'erreur ou encore l'apparition d'une info-bulle.

#### 3.1.2 Gameplay

Pour le gameplay également le retour sonore est très important. Il indique au joueur que son action a eu un impact sur le jeu, ou qu'un événement a eu lieu. Cette information est d'autant plus importante si le joueur n'a pas vu l'information visuelle associée (parce qu'il regardait ailleurs ou parce que trop d'informations sont affichées par exemple). Les événements associés incluent donc le classique saut de l'avatar, le contact avec les différents PNJ (Personnages Non Joueurs) du jeu, l'échec ou la réussite d'un niveau, l'interaction avec un élément activable du décor ou encore l'utilisation d'une compétence ou habileté.

#### Autre :

Bien que nous n'ayons pas traité cet aspect là, bien d'autres éléments peuvent être associés à l'utilisation de sons. On peut ainsi imaginer que des éléments du décor émettent aussi un son, une voix ou une musique, que le joueur serait en mesure d'entendre lorsque son personnage s'approcherait de la source sonore. Cela permet d'ajouter une dimension d'immersion supplémentaire au jeu.

### 3.2 Musiques

Le projet Chemicall est certes un Serious Game et a donc pour objectif de faire passer un message sérieux, il n'en reste pas moins en premier lieu un jeu vidéo, et se doit donc d'être ludique et agréable à jouer. Dans cette optique, les musiques jouent un rôle important. C'est un composant essentiel du Jeu Vidéo que l'on retrouve dès ses prémices dans des formats 8 puis 16 bits. Ces sonorités nous ont d'ailleurs quelque peu inspirés, puisque, comme nous le verrons plus tard, l'on retrouve dans nos compositions cet aspect "minimaliste".

Avant de parler de l'aspect technique musical, ciblons d'abord un peu plus le contexte de création de ces musiques, ainsi que la démarche créative que nous avons suivie pour leur réalisation.

#### 3.2.1 Contexte

Comme expliqué dans la partie 2.1, Chemicall est un jeu de plateformes divisé en plusieurs niveaux. A chacun de ces niveaux correspond un monde particulier (dans sa conception tout du moins, la version présentée n'étant qu'un prototype) et un atome à libérer pour notre personnage. Afin de renforcer la distinction entre chacun de ces niveaux, de ces mondes, nous voulions créer des ambiances propres à chacun d'entre eux : graphiquement et musicalement. Un autre élément important

était le menu principal du jeu, passage incontournable pour le joueur. Nous nous sommes particulièrement concentrés sur l'aspect sonore en créant quatre thèmes musicaux : un pour le menu et trois pour les niveaux. L'utilisation d'un monde microscopique et d'atomes nous laissent assez libres quand aux thèmes et ambiances sonores. Nous avons cependant gardé une certaine cohérence musicale entre les thèmes des différents niveaux, avec une musicalité "électronique" commune, rappelant le thème de notre jeu.

Ambiances :

- Caverneuse : pour le menu Jouer le son (Adobe Reader)
- Froide et apaisante (thème 'glacial') Jouer le son (Adobe Reader)
- Spatial Jouer le son (Adobe Reader)
- Dynamique et "vintage" futuriste Jouer le son (Adobe Reader)

### 3.2.2 Démarche artistique de création

Si la musique est une facette incontournable du jeu vidéo, celle-ci n'est pas notre spécialité et ce fut pour nous une découverte que de réaliser des compositions musicales. Nous nous sommes donc inspirés des expériences dont a pu nous faire part M. Tari (cf 7), en nous adaptant bien sûr à nos propres modes de fonctionnement et de ressenti. Pour mieux comprendre ce processus, nous allons tenter de vous le détailler.

Tout d'abord, comme indiqué précédemment, il faut savoir que nous n'étions initialement que peu limités dans le choix de nos ambiances musicales. En effet, si ce n'est le thème général du jeu, aucun "Level" de jeu n'était alors défini et nous pouvions donc concevoir et moduler le LevelDesign selon notre gré, et donc le SoundDesign par la même occasion.

La première étape est une phase d'inspiration qui consiste à "s'imprégner" de différentes sonorités et ambiances musicales. Pour cela, nous avons écouté les différents éléments présents dans des soundpools, ou écouté la radio par exemple. Une autre source d'inspiration était bien sûr nos expériences de joueurs : certains thèmes sont récurrents dans le monde du Jeu : on retrouve fréquemment les "mondes" du feu, de l'eau, la glace, l'espace, un monde aérien ou un autre sous terrain etc.

Une fois un thème choisi et les bases sonores correspondantes mises en place, l'idée est alors de venir enrichir la musicalité par incrémentation de sonorités - éléments perturbateurs, renforcement irrégulier du thème- et d'effets sur ces sons : filtres fréquentiels, ajustement du pitch, réglage du son etc.

Tout au long du processus de création, il est nécessaire de faire la correspondance (mentalement, puis musicalement) entre le son ou l'effet que l'on souhaite ajouter, l'ambiance déjà établie dans le morceau, et l'émotion que l'on cherche à transmettre dans le niveau de jeu. Cela implique d'arriver à prendre une certaine distance sur son travail. Cela peut se faire en faisant écouter le travail à une personne tierce, ou simplement en retravaillant plus tard le morceau, une fois les petits détails de création oubliés.

On notera quelques particularités des créations musicales ici présentes. Celles-ci doivent pouvoir tourner en boucle sans discontinuité musicale afin de ne pas perturber le joueur. Elles ne doivent pas non plus devenir trop présentes : la musique est ici une pièce seulement du puzzle, et non pas la finalité de l'application.

### 3.2.3 Musicalité

Comme indiqué en 3.2.1, nous voulions donner à Chemicall une certaine ambiance électronique, jouant ainsi sur la métaphore de l'électron, élément important dans le jeu. Cela se traduit donc par un emploi récurrent de musiques et sons électroniques dans les différents thèmes des niveaux. Comme la musique est ici un accompagnement sonore pour le joueur, elle ne doit pas être agressive ni surcharger l'attention auditive du joueur. L'utilisation notamment de sonorités issues de la Minimale ou de la Tech House nous ont ainsi permis "d'habiller" sobrement nos niveaux. Une autre caractéristique que l'on retrouve, est la simplicité de la plupart des sons, une certaine référence aux anciennes musiques de jeux vidéo, assez dépouillées. L'on retrouve ainsi très peu de cordes ou de vents, se concentrant sur les sons électroniques et les percussions.

# Chapitre 4

## Approche Technique

L'autre grande étape de ce projet à été l'intégration à proprement parler de nos réalisations dans le jeu. Nous présenterons d'abord les outils que nous avons utilisés pour cette intégration. Ensuite, nous expliquerons la conception du projet avec FMOD Designer, puis l'utilisation de ce projet dans le jeu via l'implémentation d'une classe "SoundManager" en-capsulant FMOD EX.

### 4.1 Outils utilisés

**Langage** Le projet sur lequel nous avons décidé de faire la sonorisation étant réalisé en C#, l'intégration sonore l'est aussi.

**Framework** Pour l'ajout des sons nous avons choisi d'utiliser la librairie vue en cours, FMOD EX, disponible initialement pour le langage C. Le projet étant réalisé en C#, nous avons utilisé le binding C# officiel présent dans L'API FMOD EX.

**Software** Pour la création de la banque de son et des événements nous avons utilisé FMOD Designer.

## 4.2 FMOD Designer

Avant de pouvoir insérer les assets sonores au jeu, nous avons décidé de les regrouper et de les paramétrer via FMOD Designer.

### 4.2.1 Sons

Ce soft permet de générer des projets ensuite utilisables par la Librairie FMOD Ex. L'intérêt d'utiliser cela plutôt que de directement utiliser les fichiers sons est qu'ici on crée des événements sonores. La particularité d'un événement sonore est qu'il permet d'ajouter à un son des effets, éventuellement paramétrables ensuite à l'exécution. Ces paramètres sont variés : Volume, Pitch, Fade in, Filtres etc.

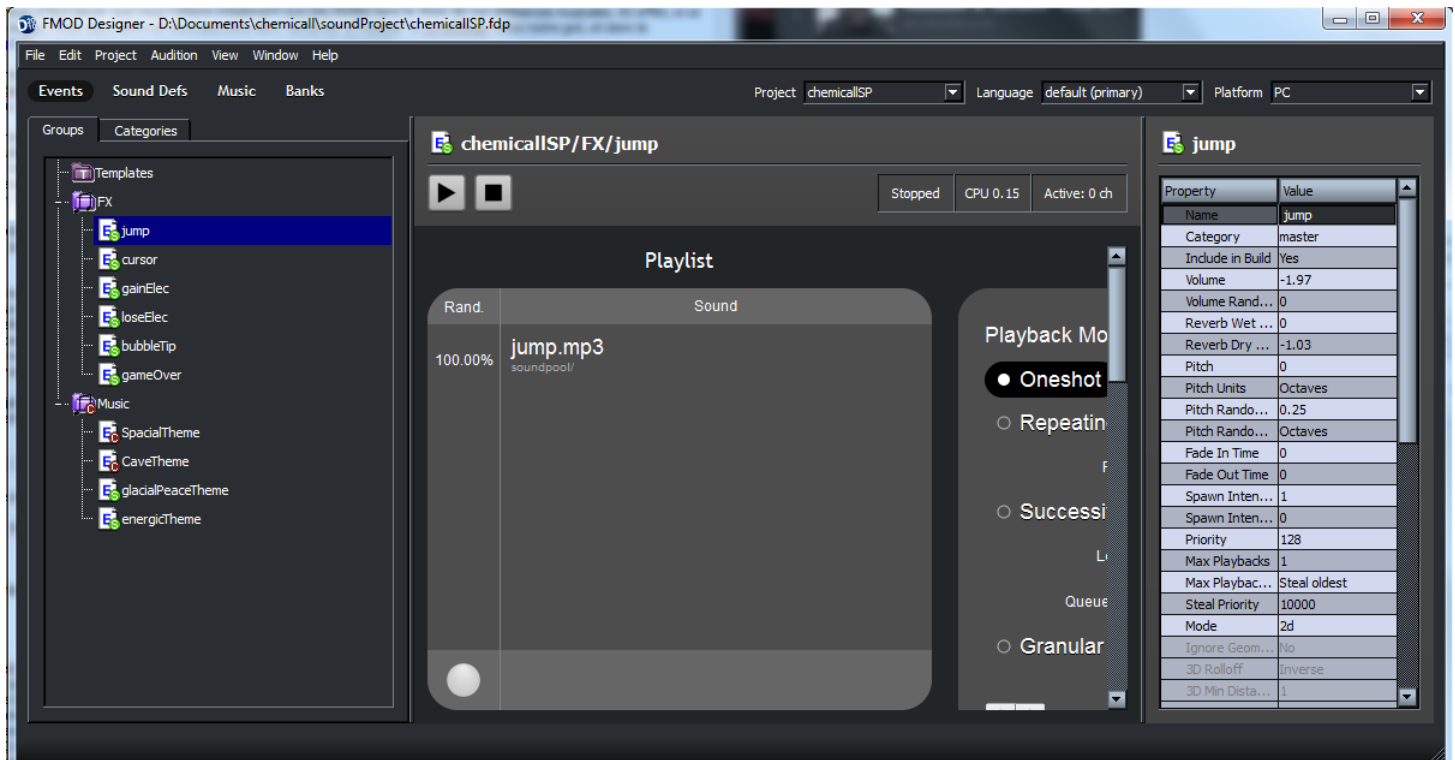


FIGURE 4.1 – Exemple de configuration de sound effect

Dans le cadre du projet, nous avons choisi d'appliquer à tous les effets sonores une réduction du volume pour éviter de saturer la bande son avec des sons à pleine puissance et d'appliquer un pitch aléatoire de manière à éviter la monotonie des effets. On peut voir sur la figure ci-dessus un exemple de déclaration d'événement.

## 4.2.2 Musiques

D'autre part, nous avons aussi déclaré les musiques comme événements sonores. Chaque musique créée pour les phases de gameplay possèdent un même paramètre, ici appelé "charge", qui permet de modifier le Pitch de la piste sonore. Ce paramètre permet en jeu de pouvoir accélérer ou ralentir la musique en fonction de l'état de notre atome. Aussi, les musiques de "gameplay" subissent une réduction de volume de 6 Db, de manière à ne pas surpasser les feedbacks apportés par les effets sonores, la musique étant ici surtout une musique d'ambiance.

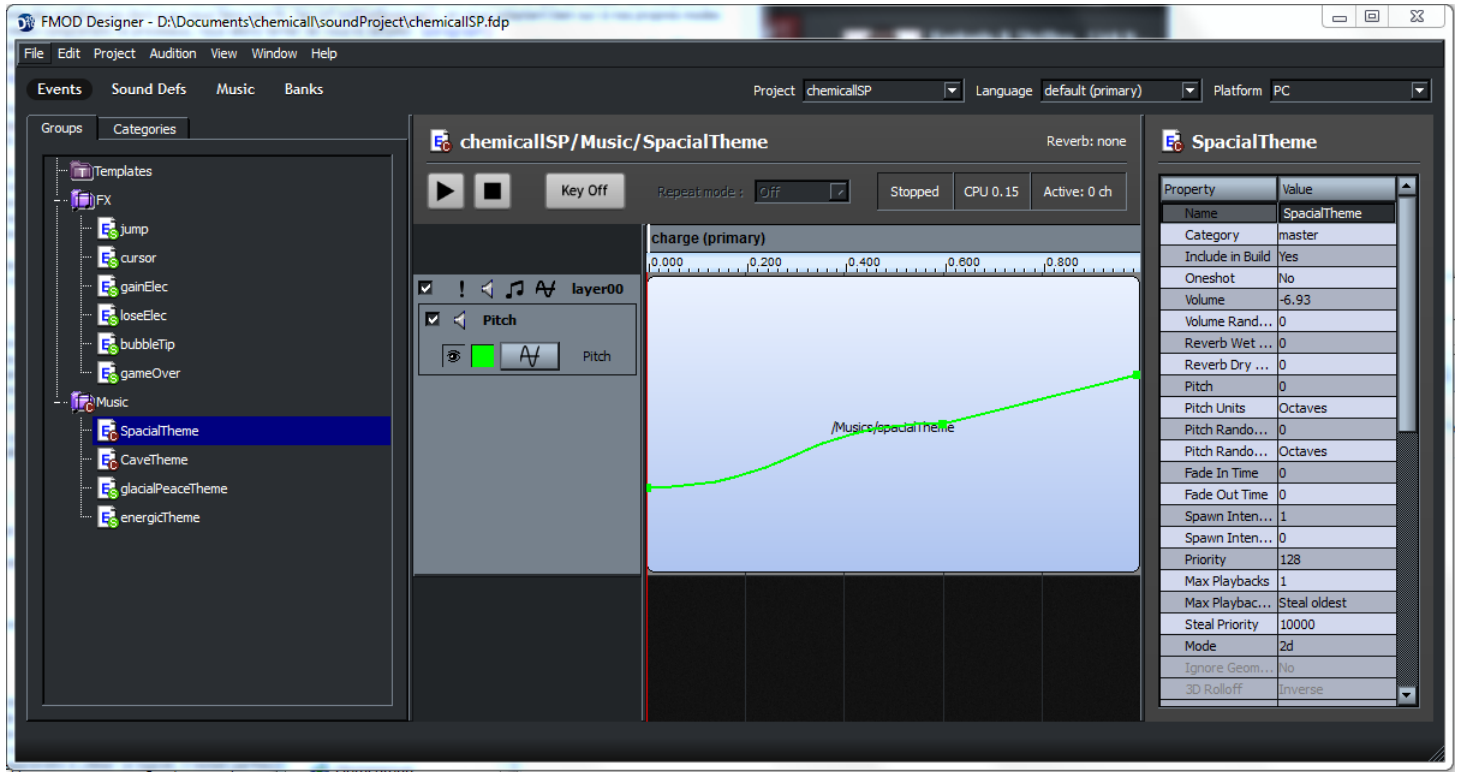


FIGURE 4.2 – Exemple de configuration d'événement musical

On peut voir sur la figure ci-dessus comment un événement multi-piste fonctionne. La courbe verte représente ici la valeur du pitch en fonction du paramètre charge, on constate que la courbe est à zéro à une charge de 0.600, valeur représentant l'état neutre de notre atome, à savoir 6 électrons.

A la fin de cette étape, on obtient alors un fichier projet .fev et sa sound bank maintenant exploitable par FMOD EX.

## 4.3 Sound Manager

Nous avons décidé de développer une classe "SoundManager"

### 4.3.1 Rôle

Le rôle de cette classe est de gérer l'utilisation des sons dans le jeu. Elle en-capsule la librairie FMOD Ex, permettant ainsi de cacher au niveau du code les détails de l'initialisation du moteur de son. Cette classe est un singleton car le moteur de son doit être unique. D'autre part, ce choix d'implémentation permet un accès plus simple de toutes les composantes au SoundManager, ceci permettant de faciliter grandement la liaison entre les événements de jeu et les événements sonores.

### 4.3.2 Fonctionnalités

Cette classe est basée sur quatre fonctionnalités principales : L'initialisation, le lancement, l'arrêt et la modification de la valeur d'un paramètre d'un événement. Comme expliqué ci-dessus, SoundManager ne possède qu'une instance, présente dans la classe elle-même. Cette instance est publique, ceci permettant l'accès au SoundManager à toute les entités du jeu.

```
namespace chemicall
{
    public class SoundManager
    {
        private static SoundManager instance;|
        Dictionary<int, FMOD.Event> soundList;
        FMOD.EventSystem eventsystem = null;
        FMOD.RESULT result;

        public static SoundManager getInstance()...

        private SoundManager()...

        public void initEvent(int id, string path)...

        public void startEvent(int soundId)...

        public void stopEvent(int soundId)...

        public void setParameterValue(int soundId, string paramName, float value)...
```

FIGURE 4.3 – Aperçu de la classe SoundManager

On peut voir dans l'aperçu ci-dessus que la classe possède comme attribut un dictionnaire associant une int ID à un FMOD.Event. Étant donné que lors de l'exécution on va forcément jouer plusieurs fois le même événement, il semble plus simple en début d'exécution, de charger ce dictionnaire, permettant alors de pouvoir accéder aux événements par leurs identifiants, déclarés comme constantes, plutôt que d'effectuer une requête sur le l'EventSystem à chaque fois pour retrouver le bon événement.

# Chapitre 5

## Discussion

### 5.1 Difficultés Rencontrées

Lors de la création de ce projet nous avons rencontré un certain nombre de difficultés que nous allons partager dans cette partie.

**Utilisation de Samplitude :** en effet, il est certain que ce logiciel est très performant et efficace, cependant pour pouvoir l'utiliser à son potentiel maximum (ce dont nous ne sommes pas encore capables), il faut un minimum d'expérience en la matière et quelques heures "d'entraînements". Même si nous avons eu quelques TPs pour nous apprendre à utiliser ce logiciel, il restait parfois difficile d'arriver à créer rapidement ce que l'on voulait.

**Utiliser FMOD sous C# :** FMOD une fois implémenté dans notre Serious Game est un véritable bonheur, tout se fait bien et qui plus est rapidement. Cependant, cette librairie n'est disponible que sous C++ et vu que notre application ne tournait que sous C# il a fallu trouver le binding nécessaire pour pouvoir l'utiliser avec notre Serious Game. Même s'il existe un binding officiel pour C#, il fut assez compliqué à mettre en place convenablement.

**Création de musiques :** nous ne sommes en effet pas musiciens et ce n'est absolument pas trivial pour nous d'arriver à créer une bande sonore complète et en phase avec l'esprit de notre Serious Game. Il fallait de plus ne pas surcharger nos musiques, et c'est quelque chose même si cela paraît un peu évident et pourtant pas si facile à faire. En effet, lorsque des morceaux de musiques nous plaisaient nous voulions absolument les mettre dans nos musiques et pourtant quelque fois ce n'est pas forcément la meilleure chose à faire. Il vaut mieux faire quelque chose de simple mais efficace. Cela est d'autant plus vrai que ce jeu a pour vocation d'être expérimenté par des enfants et mettre des musiques trop complexes pourrait aggraver leur ressenti au cours de leurs parties.

**Le loop :** nos musiques, que ce soit à l'intérieur des niveaux ou lors du menu, tournent en boucle. Il fallait donc que la fin de nos musiques ne soit pas trop rude et en relation avec le début de ces mêmes musiques pour qu'elle puissent s'enchaîner correctement.

**Prendre de la distance :** il est difficile parfois de prendre de la distance sur le travail fait surtout pour nous informaticien. En effet lorsque que l'on nous demande de faire une application avec certains critères il est "assez" facile de dire à la fin de la réalisation de cette application de dire qu'elle est réussit et complète. Cependant avec les musiques, on peut penser que nous avons réussi à faire ce que nous voulions donné parce que nous les avons créés nous-même. Et il est donc difficile de prendre de la distance pour se dire si réellement elles sont réussies ou non.

### 5.2 Apports personnels

**Samplitude :** ce projet nous a permis de nous familiariser avec des outils utilisés par des professionnels et de comprendre tout ces petits détails appliqués aux musiques et aux sons qui nous permettaient sans s'en rendre compte d'apprécier un jeux-vidéo, outre le gameplay, par la musique et l'ambiance qu'elle engendre.

**L'approche artistique :** on ne s'était jamais intéressés au côté artistique d'un jeux-vidéo. En effet nous sommes concepteurs, et ce côté là nous ne le connaissions pas. Ce projet nous a permis de comprendre comment les Sound Designer essayent d'incorporer des ambiances au travers de musiques ou de bruitages pour donner des sensations ou des émotions aux joueurs. Et il faut le dire, c'est assez complexe. Mais c'est un bon point pour nous concepteurs de pouvoir se mettre à leur place et comprendre cette difficulté.

# Chapitre 6

## Conclusion

### 6.1 Conclusion

Pour conclure nous dirons que ce projet nous a beaucoup apporté. Tant sur le point de la connaissance de nouveaux logiciels comme Samplitude, FMOD event, et autres, que sur le travail que doivent fournir les Sound Designer, comme la recherche de bruitages, l'incorporation de musiques cohérentes et variés. De plus nous sommes assez satisfaits du travail fourni tout au long de la création de ce projet et du résultat obtenu. Nous avons réussi à créer les ambiances qui nous semblaient convenir avec notre Serious Game et l'ensemble des musiques et bruitages nous semble cohérent et suffisamment varié.

### 6.2 Perspectives

Il y a d'autres choses que nous aurions aimé faire pour améliorer notre projet de Sons et Musiques en lien direct avec notre Serious Game. En effet, notre Serious Game n'est à l'heure d'aujourd'hui qu'un simple prototype, nous aurions aimé que celui-ci soit réellement fini pour pouvoir y tester toutes nos musiques et bruitages et voir la cohérence et la variété de ceux-ci à travers l'utilisation complète de notre Serious Game. De plus il y a d'autres effets que nous aurions aimé créer, comme un effet de réverbération lorsque notre atome se transforme en molécule d'eau par exemple, que ce soit sur le thème du niveau en question ou les différents bruitages joués à ce moment là. Il y a aussi quelque chose que nous aurions aimé faire et ceci est plus sur le point personnel : la création de bruitage par nos propres moyens. Comme nous l'a montré durant son intervention Mme Nastassia Jacquet avec pour exemple la création d'un bruitage de Shotgun. Cela aurait été très intéressant sur le plan personnel de voir comment on peut parvenir à créer nos propres bruitages.

# Chapitre 7

## Références

### Personnes nous ayant apporté leur enseignement et compétences :

- Frederic Tari : Compositeur, créateur sonore, Enseignant et directeur artistique (<http://frederic.tari.free.fr/>)
- Nastassia Jacquet : Sound designer, technicien du son et compositeur (<http://fr.linkedin.com/pub/nastassia-jacquet/50/46/a51>)

### Sources web

- <http://www.education.gouv.fr/>
- <https://fr.wikibooks.org>
- <http://www.demonixis.net>
- <http://fr.wikipedia.org>

**Deuxième partie**

**Annexes**

# Annexe A

## Diagramme de Classes

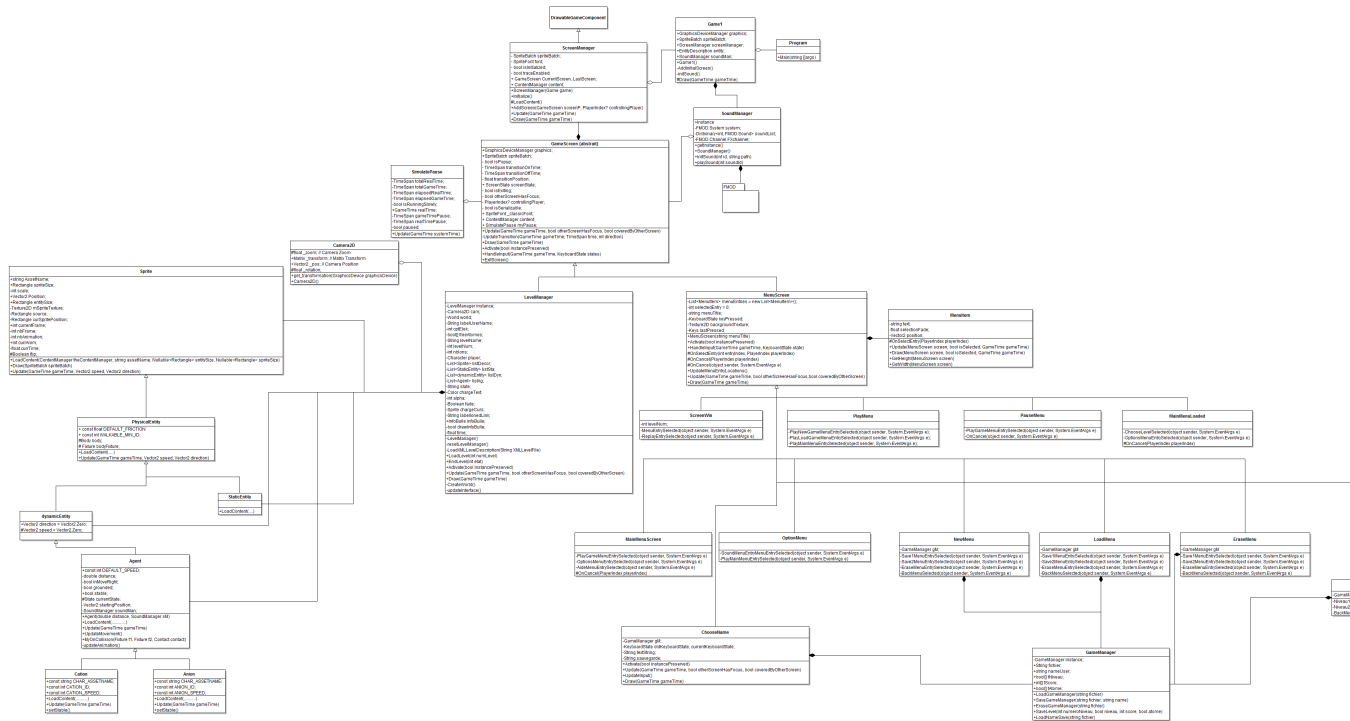


FIGURE A.1 – Aperçu de l'application

**Annexe B**

# **Game Design Document**

SERIOUS GAME

---

# Game Design Document

## Chemicall

---



Nathalie GLAD  
Thomas HINSINGER  
Geoffrey MELIA

Ailin MORENO  
Kevin BOLLINI

5 Décembre 2012

# Table des matières

<b>1</b>	<b>Pitch du jeu</b>	<b>2</b>
<b>2</b>	<b>Coeur(s) de gameplay</b>	<b>2</b>
2.1	Mono-Core . . . . .	2
<b>3</b>	<b>Contrôles</b>	<b>2</b>
<b>4</b>	<b>Camera</b>	<b>2</b>
<b>5</b>	<b>Acteurs</b>	<b>2</b>
5.1	Joueur . . . . .	2
5.2	Ennemis . . . . .	3
5.3	Éléments . . . . .	3
<b>6</b>	<b>Feedback</b>	<b>4</b>
6.1	ingame . . . . .	4
6.2	Menu . . . . .	4
6.3	Sons . . . . .	4
<b>7</b>	<b>Assistance</b>	<b>4</b>
<b>8</b>	<b>Boucle OCR</b>	<b>5</b>
8.1	Boucle Principale . . . . .	5
8.2	Boucle "Quête-Annexe" . . . . .	5
8.2.1	Quête-Annexe 1 . . . . .	5
8.2.2	Quête-Annexe 2 . . . . .	5
<b>9</b>	<b>Règle du jeu</b>	<b>5</b>
<b>10</b>	<b>Menu</b>	<b>5</b>

# 1 Pitch du jeu

Notre petit ami l'atome d'Oxygène se retrouve bloqué dans un monde imaginaire. Vous devez l'aider à se sortir de ce monde pour retrouver ses amis atomes. Mais plein d'obstacles se dressent sur son chemin. Il devra franchir ces obstacles à l'aide de ses amis Hydrogène, Carbone (et les autres!). Mais pour se faire il devra au préalable les libérer au cours de son épopée fantastique. Libérez les tous et affrontez le boss final pour enfin le sortir de sa prison.

## 2 Coeur(s) de gameplay

### 2.1 Mono-Core

- Jeu de plateforme 2D/casse tête.
- Traverser un niveau tout en évitant les obstacles (trou, mur, etc...), et les ennemis (anions, cations) pour finir le niveau en restant stable.
- Si le joueur termine le niveau en restant stable, il gagne l'atome correspondant au niveau. Il pourra par la suite utiliser cet atome pour se transformer en molécule. On appelle cela une compétence : si l'atome d'Oxygène gagne l'atome d'Hydrogène, il peut alors se transformer en molécule H<sub>2</sub>O et se transformer en eau, pour traverser certains obstacles (une grille par exemple).
- Le joueur doit donc combiner toutes ses compétences pour terminer le niveau tout en restant stable.

## 3 Contrôles

- Déplacer (Gauche et Droite)
- Sauter
- Utiliser abilité
- Changer abilité

## 4 Camera

- Scrolling Horizontal et Vertical

## 5 Acteurs

### 5.1 Joueur

- Le joueur représentant l'atome initial, l'Oxygène.  
Voici à quoi il pourrait ressembler :



FIGURE 1 – Atome d'Oxygène stable



FIGURE 2 – Atome d'Oxygène en surcharge



FIGURE 3 – Atome d'Oxygène en sous charge

## 5.2 Ennemis

- Cations : possèdent un électron en moins.



FIGURE 4 – Un Cation stable



FIGURE 5 – Un Cation en sous-charge

- Anions : possèdent un électron en trop.



FIGURE 6 – Un Anion stable



FIGURE 7 – Un Anion en sur-charge

## 5.3 Éléments

- Interrupteurs : Altèrent la température du niveau.
- Atomes : peuvent être appelés par le joueur pour utiliser leurs abilities.
- Pièges : trous que le joueur doit éviter, obstacles à éviter, etc...

## 6 Feedback

### 6.1 ingame

Affichage du nombre d'électrons possédant le joueur avec un code couleur

- < 6 : orange->rouge (0 = niveau perdu)
- = 6 : vert (0 = niveau perdu)
- > 6 : orange->rouge (0 = niveau perdu)



FIGURE 8 – La barre de vie du joueur lorsqu'il est stable

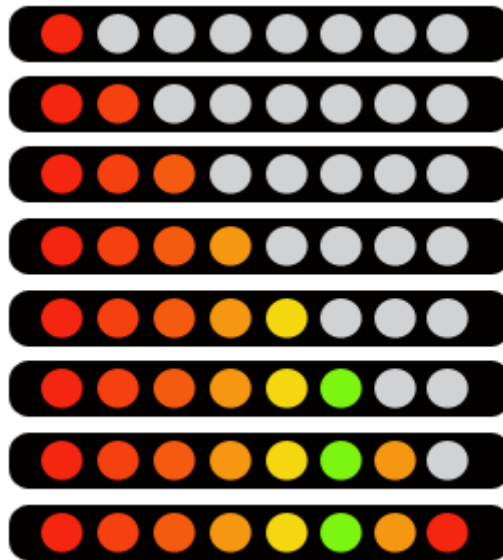


FIGURE 9 – Toutes les barres de vie possible pour le joueur

### 6.2 Menu

Affichage des niveaux réalisés et des atomes gagnés.

Possibilités à tout moment du jeu, d'afficher les atomes possédés par le joueur ainsi que l'explication de leurs associations (2 parties : une théorique pour l'aspect éducatif, l'autre pratique pour le gameplay)

### 6.3 Sons

- Musique d'ambiance à l'intérieur du niveau changeant en fonction du nombre d'électrons possédés par l'atome d'Oxygène
- Bruitage lorsque l'atome d'Oxygène touche un anion ou un cation

## 7 Assistance

- Affichage du nombre d'électrons que possède le joueur
- Affichage lors de la réussite d'un niveau, de l'atome gagné
- Explication de l'association entre l'atome gagné et celui d'oxygène
- Explication de la molécule pouvant être formée, ainsi que son état et son intérêt

## 8 Boucle OCR

### 8.1 Boucle Principale

- Objectif : Réussir le niveau sans "mourir" (en gardant au minimum 1 électron).
- Challenge : Finir le niveau avec exactement six électrons.
- Récompense : Nouvel atome gagné.

### 8.2 Boucle "Quête-Annexe"

#### 8.2.1 Quête-Annexe 1

- Objectif : Équilibrer tous les atomes, cations et anions, en leur retirant ou ajoutant des électrons.
- Challenge : Stabiliser chaque atome en évitant de se déséquilibrer fatalement soi-même.
- Récompense : Badge Chimiste accompli.

#### 8.2.2 Quête-Annexe 2

- Objectif : Gagner tous les atomes.
- Challenge : Finir chaque niveau avec exactement six électrons.
- Récompense : Badge Tableau périodique des éléments.

## 9 Règle du jeu

Vous incarnez un atome d'Oxygène. Vous possédez un nombre précis d'électrons à votre départ. Vous pouvez perdre ou gagner des électrons. Le but du jeu est de finir le niveau avec au moins six électrons. Pour cela, vous avez des alliés, nommé Anions, ils possèdent un électron en trop, ils pourront alors vous le donner. Cependant une fois l'échange effectué, l'anion devient alors stable, et il ne peut vous en donner d'autres. Vous aurez aussi des ennemis, des cations, il leur manque un électron et donc à leur contact, il vous prendront un électron sur la totalité que vous possédez (game over si ce total tombe à 0). Lorsque que vous démarrez le jeu, vous avez la possibilité de choisir parmi les premiers niveaux disponibles. Chaque niveau indique l'atome pour lequel vous jouez. Si vous finissez le niveau avec au moins six électrons, vous gagnerez cette atome. Une explication vous sera donnée sur le fonctionnement de cet atome avec le votre (Oxygène O) et les possibilités qui s'offrent à vous Dans les prochains niveaux, vous pouvez alors appeler le (ou les) atome(s) gagné(s) afin de vous transformer en une molécule avec un état physique particulier. Il vous aidera à passer les obstacles ou à vous défendre.

## 10 Menu

- Jouer
- Règles du jeu
- Options (Son, Contrôles)
- Récapitulatif de toutes les propriétés physiques/atomiques découvertes
- Quitter le jeu