Mémoire – Projet tuteuré

cHTeMeLe Surface

# Projet :

|  |  |
| --- | --- |
| Nom du projet | cHTeMeLe Surface |
| Tuteur | Adélaïde Albouy-Kissy |
| Relation avec une entrepise | Non |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Participants |  |  |  |  |
| Barbara Schiavi | Valentin Mourot | Clément Janisset | Pierre-Rémi Nouvet-Gire | Volodia Mandaud |

# Introduction

## Le projet cHTeMeLe Surface

Créé par Olivier Chambon (Créateur) et Pascal Mabille (Illustrateur) en 2012, cHTeMele est un jeu de société éducatif qui a pour ambition de faire apprendre les bases du langage HTML 5 aux néophytes. Pour rappel, le HTML5 (HyperText Markup Language) est un langage de présentation de données créé par le W3C (World Wide Web Consortium) en collaboration avec le WHATWG (Web Hypertext Application Technology Working Group) et est un standard utilisé dans la création de sites web.

Disponible depuis le 20 Septembre 2012 sous la licence "Creative Commons CC-BY-NC" (Téléchargement libre et gratuit pour une utilisation non commerciale), cHTeMeLe est un jeu de cartes dont le but est de créer une page HTML correcte. Les joueurs posent des cartes à tour de rôles afin de construire une page HTML et remporter le plus de point possible.

cHTeMeLe est téléchargeable sous la forme d'un jeu de 112 cartes représentant des balises HTML ainsi que 5 cartes de joueurs à l'image de 5 navigateurs web connus (Chrome, Firefox, Opera, Safari et Internet Explorer). Il est aussi vendu au prix de 20€ en version boîte (ou 15€ pour les étudiants ou professeur).

Le projet cHTeMeLe Surface a été créé dans le cadre des projets tuteurés de l’Université D’Auvergne. Le jeu cHTeMele étant sous licence Creative Commons, Adélaïde Albouy-Kissy a donc lancé un projet d'adaptation du jeu sur une table tactile. Le but était de créer une application fonctionnant sur la table Tactile Samsung SUR40 disponible à l'IUT, mais aussi de façon plus générale sur toutes les tables tactiles disposant de PixelSense et de la capacité à reconnaitre des Byte tags (Mais nous y reviendrons plus tard).

Le principal intérêt de porter le jeu cHTeMeLe sur une table tactile est de pouvoir avoir un aperçu en temps réel de la page HTML que les joueurs construisent. En effet, dans le jeu initial, les joueurs ne pouvait voir le résultat que si à la fin de la partie, ils rédigent le code dans un fichier HTML et le lancent sur leur navigateur (ce qui est fastidieux, mais qui peut aussi paraitre compliqué pour le public visé par le jeu : les néophytes). La table tactile apporte aussi un certain confort lors du jeu dans le sens où elle comptera automatiquement les points des joueurs, ainsi que la possibilité aux joueurs d'accéder directement à une documentation personnelle et spécialisée pour cHTeMeLe afin de pouvoir renseigner les joueurs lorsqu'ils le désirent.

## Utiliser la table tactile Surface SUR40

À compléter // Discours d’Adélaïde sur l’utilisation de la table pour les projets étudiants

# Présentation du sujet

Pour développer cette édition tactile de cHTeMeLe, nous étions libres de remodeler le jeu selon nos envies. Cependant nous avions certains critères à respecter dans l’application, que nous allons vous expliquer ci-dessous.

## Reconnaissance des cartes

Le jeu se devait de permettre aux joueurs de poser leurs cartes sur la table tactile et que celle-ci les reconnaisse pour interagir avec. *(Ex : Faire apparaitre des informations supplémentaires autour de la carte, pré-compléter le code HTML en surbrillance, etc)*

Pour cela nous devions utiliser la reconnaissance de Tags que propose la table tactile.

## Choix de feuilles de style

Cette option relève du gadget mais fait partie du sujet. L’idée est de proposer aux joueurs plusieurs manières de visualiser ce que donne leur code HTML dans un navigateur. Les sites web utilisent des feuilles de style écrites en CSS (Cascading Style Sheet) associées à leur page HTML pour rendre la page visuellement plus attrayante. cHTeMeLe Surface devait proposer aux joueurs la possibilité de changer à tout moment la feuille de style CSS parmi plusieurs proposées pour changer le style visuel de leur page HTML.

## Règles syntaxiques

cHTeMeLe étant un jeu à but éducatif, il fallait qu’il détecte les erreurs dans le code créé par les joueurs ou qu’il les empêche de les faire.

## Indentation à respecter

Pour garder un code propre le jeu doit faire en sorte que l’indentation du code soit respectée.

<html>

<head>

<title>**Ceci est le titre de la page**</title>

</head>

<body>

<h1>**Ceci est une balise sur une ligne.**</h1>

<p>

**Ceci est une balise sur plusieurs lignes.**<br/>

**Il contient des balises qui n'engendrent pas de**

**retours à ligne** <em>**comme celle-ci**</em>**.**

</p>

<div>

<h2>**Exemple**</h2>

</div>

</body>

</html>

Exemple d’indentation de code HTML

Globalement une bonne indentation de code HTML distingue trois types de balises :

* Les éléments multi-lignes : les deux balises ouvrante et fermante occupent chacune une ligne entière et ce qui se trouve entre les deux est indenté au niveau inférieur. *(Ex : <p>, <div>, <body>, etc.)*
* Les éléments mono-ligne : l’élément entier occupe une ligne entière. *(Ex : <h1>, <h2>, <title>, etc.)*
* Les éléments inline : l’élément est situé dans un autre élément non-inline. Il n’engendre aucun retour à la ligne.  
  *(Ex : <em>, <br>, <strong>, etc.)*

*Un élément html est composé d’une ou deux balises (une ouvrante, et souvent une seconde fermante), ainsi que d’un contenu qui peut être du texte et/ou d’autres éléments html.*

# Présentation du projet

## Réécriture des règles

À partir de cet énoncé nous avons imaginé notre version du jeu cHTeMeLe. Nous avons réécrit les règles pour qu’elles soient le plus simple possible à prendre en main.

De plus, dans le jeu original sont présentes de nombreuses cartes « Action » permettant d’entraver la progression de ses adversaires ou de faciliter la nôtre. Beaucoup d’entre elles étaient trop complexes ou trop puissantes. Par exemple l’une d’entre elles demandait aux joueurs de recommencer la partie en conservant le score actuel des joueurs.

Nous avons donc refait toutes ces cartes « action » en diminuant leur nombre et en rendant leur puissance plus homogène.

Par ailleurs, le jeu original demande de débuter la partie en posant la carte **<html>** et propose à n’importe quel joueur de terminer la partie prématurément en posant, s’il l’a, la carte **</html>**, lui permettant de remporter un grand nombre de points, ce qui le plus souvent faisait gagner ce joueur.

Nous avons décidé de retirer cette carte du jeu et rendre le système de score moins chaotique. En effet quand un joueur posait une carte Balise fermante (</balise>), il était gratifié d’un nombre de points en fonction du contenu de la balise. Or l’élément **<html>** englobe toute la page web. Le joueur qui posait la carte **</html>** gagnait un score égal au total des points qu’octroyaient chaque carte posée durant la partie.

Le nouveau système de score ne récompense plus la fermeture d’un élément html que par un nombre de points fixe, égal au nombre de points qu’octroie la pose de la balise ouvrante de l’élément html.

Du coup, il fallait réfléchir à de nouvelles conditions de fin de partie. Les règles originales stipulent que la 10e ligne de code est également la dernière. Nous avons trouvé cela absurde et avons décidé de fixer un nombre de tours limité au début de par partie (10 par défaut, mais modifiable). Cela vaut mieux car souvent les joueurs n’ont pas les cartes nécessaires pour bien clôturer le code qu’ils posent. Le jeu original sanctionnait ce fait par une perte de points. Nous avons décidé de ne pas limiter le nombre de lignes et de ne pas sanctionner le code incorrect en fin de partie.

Nous avons légèrement changé la liste des balises html que le jeu proposait. Cela dit certains éléments html ne peuvent être posés qu’à l’intérieur de certains autres et les rendent très difficiles à jouer. C’est pour cette raison que le jeu ne propose pas de créer des listes et des tableaux. Cela dit, certaines cartes restent plus difficiles à jouer, notamment certaines cartes attributs *(src, href, title)*, car ils ne peuvent pas aller dans tous les éléments html.  
Ces cartes auront plus de valeur que les autres en termes de score.

Enfin, nous avons préféré augmenter la quantité de code que poseront les joueurs au fil du jeu. Nous leur donnons non plus 6 mais 10 cartes au début de chaque tour et la possibilité de poser autant de cartes html qu’ils souhaitent. Le code créé par les joueurs sera plus vaste et le jeu plus dynamique.

## Utilisation de l’interface tactile

La table tactile Samsung SUR40 peut reconnaitre simultanément un nombre limité de tags, et donc de cartes.  
Pour des raisons d’espace et techniques, nous avons donc décidé de ne pas laisser les cartes posées sur la table tout le long de la partie. Un joueur posera sa carte puis la défaussera une fois que sa pose sera validée (appui sur un bouton sur l’écran de la table). Cela permet de ne pas limiter la taille du code, au vu de la taille des cartes et des limites techniques de la table.

Pour des raisons de confort, nous devons limiter le jeu à 4 joueurs (un à chaque côté de la table) et non 5 comme dans le jeu original, et encore moins 6 comme il était proposé dans le sujet.

Nous proposons aux joueurs de rentrer eux-mêmes le contenu textuel des éléments html qu’ils posent sur la table, et donc nous retirons du jeu les cartes **Lorem** et **Ipsum** qui servaient de cartes texte. Écrire du texte ne rapporte donc plus aucun point.

Certains contenus ne sont pas textuels : l’attribut **src** d’un élément **<img>** doit contenir un chemin vers un fichier image. Le jeu proposera une petite banque d’image et permettra au joueur de choisir visuellement l’image qu’il veut insérer.

Certains contenus textuels ne devraient pas être totalement libres, sinon ils seraient inutiles. C’est le cas de l’attribut **class** : les classes sont définies dans le code HTML et son utilisées dans le code CSS. Or le fichier CSS n’est pas accessible aux joueurs. Le jeu proposera donc pour la carte **class** une liste de valeurs possibles qui seront toutes implémentées dans les feuilles de style proposées par cHTeMeLe.

Certaines cartes « action » sont appelées cartes « attaque » car elles ciblent un adversaire et lui donne un désavantage. Le jeu proposera au joueur de choisir sa cible via des boutons sur la table tactile, et l’application gèrera les conséquences de l’utilisation des cartes « action » automatiquement si possible (sinon, elle affichera une bulle d’aide aux joueurs pour leur indiquer la marche à suivre).

Au fil des tours, l’interface du jeu tournera pour se mettre dans la direction du joueur actuel, la zone où se trouve le code sera toujours de son côté et la visualisation dur navigateur en face. Le joueur dont c’est le tour aura à sa disposition un bouton pour passer la main au joueur suivant.

Chaque joueur aura à sa disposition une petite interface servant de documentation sur les cartes du jeu et un lien vers les règles. Se trouvera à côté sa « carte joueur », une interface affichant son nom, son score total, les effets des cartes « action » qui sont actives sur lui et des informations sur son tour précédent (nommé « dernière combinaison »).

L’application offrira aux joueurs la possibilité de sauvegarder leur partie pour la continuer plus tard, enregistrant dans un fichier l’était des joueurs, la progression du jeu et le code généré. À la fin d’une partie, le jeu proposera de créer un mémoire de la partie contenant le code, le visuel de la page et le tableau des scores.

# Technologies utilisées

## La table tactile Samsung SUR40

### Caractéristiques techniques

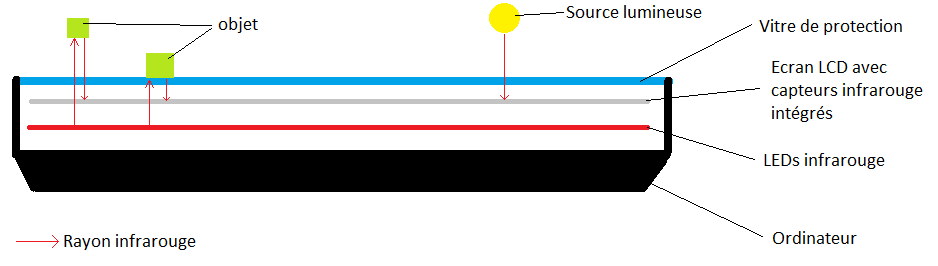
|  |  |
| --- | --- |
| Taille de l’écran | 40 pouces |
| Processeur | Athlon II X2 Dual-Core |
| Mémoire interne | RAM 4Go  HDD 320Go |
| Processeur graphique | AMD Radeon HD 6750 |
| Interfaces USB | 4 |
| Système d’exploitation | Windows 7 professionnel |

L’interface tactile de la table est recouverte d’un verre Gorilla Glass, un verre chimiquement renforcé. La technologie utilisée pour capter les entrées tactiles est basée sur un système infra-rouge. Elle peut détecter principalement les touchers, les survols et les objets réels.

### Reconnaissance des entrées

La table utilise la technologie PixelSense pour récupérer et reconnaitre les points de contact. Cette technologie consiste à envoyer des rayons infrarouge grâce à des LEDs situées en dessous de l'écran et récupère les rayons infrarouge renvoyés par des objets proches de la surface. Plus un objet est proche, plus il renverra de rayon infrarouge. Ce système dispose de plusieurs avantage: il permet d'une part de reconnaitre jusqu'à 52 points de contact simultanément (qu'ils soient en contact avec la surface ou qu'ils survolent simplement l'écran), mais aussi de reconnaitre leur forme, leur taille ainsi que leur orientation.

Mais en contrepartie, la technologie PixelSense dispose d'un problème assez important: la reconnaissance se faisant uniquement sur l'acquisition de rayon infrarouge, la table devient donc extrêmement sensible à la lumière qui peut elle aussi emmètre des rayons infrarouges. Or la table ne possède aucun moyen de vérifier si la source de rayon infrarouge est externe ou non, par conséquent la table prendra également en compte les sources lumineuses externes (comme une ampoule ou la lumière du soleil) et les interprétera comme des points de contact.

******

Schema de fonctionnement de la technologie PixelSense

Une fois les rayons récupérés, la table construit une image en niveau de gris en fonction des données acquit et interprète cette dernière pour déterminer les points de contact (type, taille, orientation).

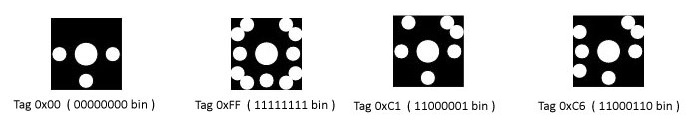
La technologie Pixelsense classe les points de contact en 3 types distincts:

### Les fingers

Les fingers représente une interaction sur la table avec un doigt (ou une petite surface tel qu'un stylo). Ils remplace en général les interactions avec la souris et permettent donc d'interagir et de naviguer sous windows tout au long de son utilisation.

### Les tags

Les tags sont des petites images carrées en noir et blanc stockant de l'information. Ces petites vignettes peuvent être interprétées, une fois posées, par la table tactile qui est alors en mesure de restituer l'information qu'elles contiennent. Il existe toute sorte de tags permettant de stocker plus ou moins de données dessus. Les tags qui sont reconnu nativement par la technologie PixelSense sont les Bytes Tags ou les Identity Tag pouvant stocker une seul valeur (allant respectivement de 0 à 255 et de 0 à 65535).



Exemple de tags (source: msdn.microsoft.com)

### Les blobs

Les blobs représentent toutes les autres interactions que PixelSense ne peut pas classer dans les 2 types précédant, c'est à dire les gros points de contact (paume de la main, coude, feuille, etc...).

## Microsoft Pixelsense

### À compléter SDK PixelSense (Surface 2.0)

PixelSense, plus connu et commercialisé sous le nom de Surface permet à un ou plusieurs utilisateurs d’avoir des interactions avec la surface tactile de la table.  
Le kit de développement pour ce type d’appareil a été mis de côté suite à la sortie des tablettes tactiles sous Windows 8, la communauté déjà pas très grande due au prix excessif de la table a fait qu’il y a eu très peu d’informations autres que la documentation faite par Microsoft.

## Pixelsense Input Simulator

### À compléter Simulateur d’entrées

Le logiciel Input Simulator, permet à n’importe quel PC fonctionnant sous Windows Vista et 7 de simuler tout type d’entrée, les fingers, les tags, et les blobs. Ce simulateur permet grâce à un clavier et une souris de positionner un ou plusieurs contacts sur votre écran comme s’il s’agissait d’un écran tactile. Bien que ce simulateur étant très complet de par ses fonctionnalités, tout est faisable comme sur la table avec un peu de patience bien sûr, nous avons rencontré des problèmes plus ou moins fréquents de plantage, ça et d’autres petits problèmes.

## Langages : C# - Xaml – HTML – CSS

### À compléter Présentation des langages utilisés et dans quel contexte

Le projet a deux parties bien distinctes, le jeu se basant sur l’interprétation du langage HTML en C# - XAML, et sa documentation faite en HTML – CSS utilisé dans un navigateur implémenté pour que le joueur ai accès directement aux informations et règles du jeu.  
  
La programmation de logiciel WPF, « Windows Presentation Foudation », se divise en trois parties, la partie Visualisation et Contrôleur qui utilisent le langage et des fichiers de type XAML, et la partie Modèle ou dit plus simplement le cerveau de l’application qui exécute toute les tâches qui s’écrit en C#.  
  
Pour une modification du contenu de la documentation plus simple et rapide nous avons décidé de l’écrire en HTML – CSS ce qui permet donc à n’importe qui ou presque de pouvoir modifier cette documentation et l’utiliser facilement en jeu par l’affichage via un navigateur.

# Mémoire du développement

La plus longue partie. C’est ici qu’on dira au fil du temps tout ce qui nous est arrivé. L’histoire d’une vie. Non j’en fais trop, là.

## Chapitre 1 : Découverte du SDK

## Chapitre 2 :

# II\_ Etapes de développement et difficultés rencontrées

### ..1.1)

### ..1.2)

### ……… 1.2) a)

### ……… 1.2) b)

### ..2.1)

# III\_ Résultats

### ..1.1)

### ..1.2)

### ……… 1.2) a)

### ……… 1.2) b)

### ..2.1)

# Conclusion