



## **RAPPORT DE PROJET**

### Jeu de puissance 4

<b>1) Présentation du projet</b>	
A) Cahier des charges	p.2
B) Le jeu portable	p.2
C) Le site Internet HTML/WML	p.3
D) Planning – diagramme de GANT	p.4
<b>2) Technologies Java</b>	
A) Etat de l'art	p.5
B) Doja et J2ME	p.7
<b>3) Technologies Internet</b>	
A) HTML	p.8
B) CSS	p.8
C) WML	p.11
<b>4) Conclusion</b>	
A) Difficultés rencontrées	p.15
B) Le résultat	p.15
<b>5) Annexes</b>	
A) Dossier de programmation	p.15
B) Dossier du site Web	p.15
C) Manuel de l'utilisateur	p.15

## 1) présentation du projet

### A) Cahier des charges

Enoncé du projet :

« L'objectif de ce projet est de réaliser un jeu de type puissance 4 en Java. La particularité de ce jeu est qu'il sera développé pour être utilisable sur les téléphones mobiles. Par conséquent, les étudiants devront faire un état de l'art sur les technologies mobiles Java et les possibilités offertes par ces technologies.

Les étudiants devront réaliser également un site internet sur le projet avec un nom de projet, un logo et mettre à disposition les sources. Ce site contiendra également des tutoriels expliquant le développement. »

Le cahier des charges est le suivant :

#### Le jeu Java :

- Le jeu doit être utilisable sur la majorité des téléphones portables (dimensions de l'écran, nombre de couleurs, touches et version de l'environnement Java)
- Le jeu doit rester léger (quelques kilo-octets seulement)
- Le jeu doit être attrayant : une interface colorée, quelques images
- Le jeu doit être facile à utiliser, intuitif
- Les deux joueurs jouent sur le même téléphone portable

#### Le site Internet :

- Le jeu doit être facilement téléchargeable depuis un téléphone portable à accès WAP
- Le site doit être correctement accessible depuis un navigateur Internet classique : mise en page uniforme et invariable
- Le site doit rester léger et rapide à charger
- Toutes les informations relatives au projet doivent être consultables depuis un navigateur Internet classique
- Les sources doivent être téléchargeables depuis un navigateur Internet classique (MS Internet Explorer, Netscape, Opera)

### B) Le jeu portable

L'objectif premier est de réaliser un jeu de puissance 4 pour téléphone portable.

Le puissance 4 est un jeu de société au tour par tour. Deux joueurs se partagent une grille de 42 cases (6 par 7), et doivent placer chacun à leur tour un jeton. Les deux joueurs disposent chacun de 21 jetons : 21 jetons jaunes et 21 jetons rouges.

Le but est d'aligner 4 jetons consécutifs, sans interruption par un jeton de couleur adverse.

Cet alignement peut se faire dans toutes les directions : verticale, horizontale, ou même en diagonale.

Le jeu s'arrête lorsqu'un joueur a aligné 4 de ses jetons, : il gagne, ou lorsque la grille est complètement remplie : match nul.

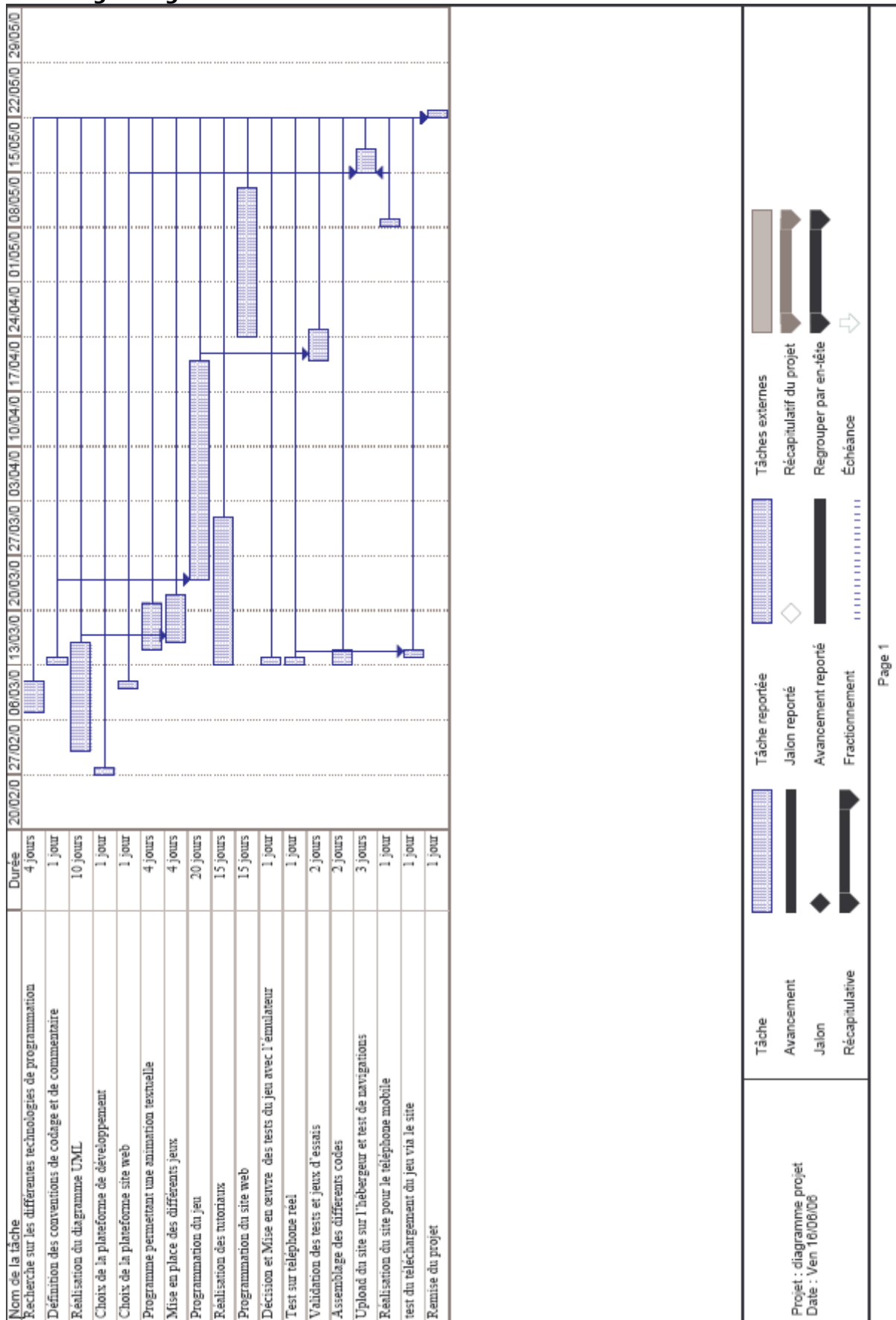
Ainsi, nous avons réalisé un jeu de puissance 4 pour téléphone portable, les deux joueurs se partageant le même appareil.

### **C) Le site Internet HTML/WML**

Le rôle du site Internet est :

- de faire connaître notre réalisation, la promouvoir
- mettre à disposition du public les sources du projet, ainsi que le téléchargement du jeu de puissance 4 via un navigateur classique ou un téléphone portable
- proposer une initiation aux technologies utilisées dans l'intégralité du projet, à savoir la conception du site : HTML, CSS et WML, ainsi que le jeu de puissance : état de l'art sur la mobilité avec Java, le concept de Java, et l'explication du code source via une documentation de type javaDoc
- diffuser nos rapports de projet

**D) Planning – diagramme de GANT**



## 2) Technologies Java

### A) Etat de l'art

#### Java

Java est un langage de programmation dit « orienté objet », aussi appelé POO, mis au point et maintenu par l'entreprise Sun Microsystems depuis 1995.

Java appartient, comme C# et beaucoup d'autres langages orientés objet, à la nouvelle génération de langage de programmation, par opposition à la programmation communément appelée « classique », visible avec des langages tels que le C.

Ce qui le différencie de la majorité de ses concurrents est sans aucun doute son incroyable portabilité, en plus de son haut niveau d'abstraction. En effet, on retrouve JAVA sur tout type de support, que ce soit des machines classiques telles que des ordinateurs de bureau, des serveurs Web, en passant par les téléphones mobiles, les assistants personnels et même les télescopes orbitaux !

Enfin, on trouve deux grandes familles de programmes Java :

- Les applications : ce sont les programmes que vous utilisez sans doute le plus souvent, logiciels tels qu'un traitement de texte, des outils, etc...
- Les appliquettes : en anglais « applets », ce sont des programmes chargés par votre navigateur Internet, et ensuite lancé sur votre machine. On retrouve beaucoup d'applets sur le Web, ne serait-ce que sous forme de jeux ou de lecteurs multimédias (mp3, etc).

#### JRE

Java Runtime Environment : Environnement d'exécution Java, plus connu sous le nom de « machine virtuelle Java ».

Comme le langage C# de .NET, Java n'est pas un langage directement compilé en code machine : tout comme .NET passant par du code « assembly », un programme Java est compilé en « pseudo code », ce dernier étant ensuite interprété et/ou compilé par la machine virtuelle (les parties les plus lourdes et les plus répétitives sont directement compilé en langage machine par la JRE).

Parce qu'il existe des JRE pour la majorité des systèmes d'exploitations, Java est extrêmement portable. Il suffit d'avoir une JRE.

Une machine virtuelle Java est comparable à un framework .NET.

#### J2ME

J2ME : Java2 Micro Edition, pourrait être vulgairement considérée comme une version bridée du J2SE (version standard du kit de développement Java).

Plus précisément, J2ME permet une prise en charge de Java sur les appareils électroniques et les systèmes embarqués, tels que les téléphones portables ou bien les tablet-PCs.

C'est justement cette plate-forme J2ME que nous avons choisit pour développer le jeu de puissance 4 sur téléphones portables, les deux autres plate-formes ne fournissant pas les outils nécessaires.

Concrètement, nous avons opté pour l'EDI<sup>[1]</sup> NetBeans 5.0, avec le « Mobility Pack 5.0 ». C'est à travers ce dernier que nous bénéficions des fonctionnalités du J2ME, en plus d'un simulateur intégré de téléphones portables et d'un éditeur d'interfaces graphiques.

**[1]** : EDI, Environnement de Développement Intégré, en anglais IDE, Integrated Development Environment. C'est une interface graphique d'aide au développement de logiciels.

## Configuration

Une configuration est pour ainsi dire la base de J2ME : il s'agit d'une machine virtuelle, ainsi que des bibliothèques de « bas niveau ». Une bibliothèque est en quelque sorte une caisse à outils, dans laquelle le programmeur va piocher pour concevoir son logiciel. Les bibliothèques de bas niveau sont les composants vitaux d'un programme Java sur votre mobile.

Parce que tous les matériels mobiles ne sont pas égaux devant la puissance : un téléphone portable ne dispose pas d'autant de ressources qu'un assistant personnel ou qu'un Pocket PC. Ainsi, on va choisir parmi plusieurs configurations.

A l'heure actuelle, il en existe deux :

- La configuration CDC (Connected Device Configuration) : on l'utilise sur les appareils mobiles performants, c'est à dire des matériels disposant d'au moins 2 MB de mémoire vive (RAM), d'un processeur 32 bits, et de 2,5 MB de mémoire morte (ROM). Bref, CDC convient aux appareils puissants, haut de gamme tels que les Pocket PC, les décodeurs, et même les télévisions.
- La configuration CLDC (Connected Limited Device Configuration) : on l'utilise sur des environnements assez limités, tels que les téléphones portables. Bref, CLDC convient aux appareils peu puissants.

## Profils

Un profil permet de spécifier quel type d'API J2ME on va utiliser.

Concrètement, il correspond à une interface homme machine (IHM) d'un type particulier, et un ensemble de méthodes d'accès aux entrées/sorties (le clavier, l'écran, la minicaméra, etc) du matériel mobile.

MIDP se décline sous deux versions : 1.0, et 2.0 (version actuelle).

## Midlet

Un profil MIDP ne s'utilise pas tout seul, c'est avant tout un support, une sorte de norme que l'on va utiliser.

Pour utiliser un profil MIDP, on va développer ce qu'on appelle des MIDlets.

Un MIDlet est une application conforme à son profil MIDP.

Concrètement, au niveau du code Java, un MIDlet est ni plus ni moins qu'une classe abstraite (on va obligatoirement la dériver) disposant de trois méthodes bien particulières :

- Une méthode **startApp** : on y place le code qui sera exécuté lors de l'appel du MIDlet, on peut parler de « constructeur » du MIDlet.
- Une méthode **pauseApp** : lorsque le MIDlet est mis en pause.
- Une méthode **destroyApp** : lorsque l'on arrête un MIDlet, on peut parler de « destructeur » du MIDlet.

## **B) Doja et J2ME**

Doja et J2ME sont deux technologies utilisées sur les matériels portables, dont les téléphones mobiles.

- Doja (DoCoMo Java) de NTT DoCoMo, est utilisé sur des périphériques « i-mode ». Doja est l'équivalent de MIDP développé par Sun Microsystems et présent sur les téléphones mobiles WAP.  
On retrouve principalement Doja en Asie, et dans ses version 1.5 et 2.5 en Europe.  
L'avantage de Doja sur J2ME est sa constance sur les différents mobiles i-mode.
- J2ME (Java2 Micro Edition), avec MIDP (Mobile Information Device Profile) de Sun, est utilisé sur les périphériques bénéficiant du WAP.  
L'avantage de J2ME réside dans sa généricité, contrairement à Doja qui est réservé aux portables i-mode. L'inconvénient est que son interprétation par les différentes machines virtuelles peut différer suivant le périphérique. C'est le prix de cette grande portabilité.

### 3) Technologies Internet

#### A) HTML

HTML : HyperText Markup Language : langage hypertexte à balises.  
C'est le langage de base utilisé pour créer des pages Web statiques. Ce langage se résume à un jeu de balises dont leur rôle est de mettre en forme la page Web : jouer sur l'aspect du contenu, et charger divers éléments.

HTML est issu du langage SGML, de dernier ayant aussi donné naissance à XML. SGML et XML sont des méta-langages : ils servent à créer d'autres langages, tels que :

- HTML
- WML
- MATHML
- RSS
- etc...

Un organisme se charge de mettre au point et de maintenir les normes liées à ces langages : le W3C (World Wide Web Consortium).

#### B) CSS

Voici les différents point de CSS abordés :

- Les unités
- Les polices de caractères
- Les marges
- Les évènements
- Le positionnement absolu
- Les unités

Il existe plusieurs unités de mesure :

- px : le pixel,
- em : une unité extensible selon la configuration de l'écran,
- cm : le centimètre,
- mm : le millimètre,
- % : une portion en pourcentage d'une dimension de la page.
- Etc.

px, cm et mm sont des unités dites fixes. Quel que soit la résolution de la page, leur taille ne changera pas.

em et % sont des unités adaptables. Elles varient dynamiquement suivant la résolution de la page.

Pour ces raisons de lisibilité et d'accessibilité, on préférera les unités adaptables de par leur compatibilité avec les petites résolutions, mais aussi avec les plus grandes.

### Les polices de caractères :

Voici les principaux styles applicables sur les caractères :

- font-family : la police d'impression : Bitsream Vera Sans, Arial, serif, etc.
- font-size : la taille des caractères, avec les unités vues précédemment (en 2.A).
- font-style : le style des caractères : normal (« normal »), en italique (« italic »), ou en oblique (« oblique »).
- font-weight : le poids des caractères : normal (« normal »), en gras (« bold »), plus gras (« bolder », on augmente de 1 le poids), moins gras (« lighter », on diminue de 1 le poids), ou une valeur : 100, 200, etc.
- letter-spacing : l'espace horizontal entre les caractères, avec les unités vues précédemment.
- line-height : la hauteur des lignes, avec les unités vues précédemment.

### Les marges :

Avec CSS, on va manipuler deux sortes de marges : « margin » et « padding ». On les utilise avec les unités de mesure vues précédemment.

Voici à quoi elles peuvent ressembler :



« margin » est la marge dite extérieur, tandis que « padding » est la marge intérieure. Ici, la bordure discontinue simule le cadre de la zone dans laquelle se trouve le texte.

Notez que les mages peuvent être générales : elles ont la même valeur pour les écart de gauche, de droite, en haut et en bas. Elles peuvent aussi être découpées en 4 sous marges : gauche, droite, haut et bas.

Deux exemples avec « margin » :

```
p{
    margin: 10px;
}
```

Ici, le margin est général et de 10 pixels.

```
p{
    margin-left: 10px;
    margin-right: 10px;
    margin-top: 10px;
    margin-bottom: 10px;
}
```

Ici, on spécifie manuellement chaque sous marge de margin, de 10 pixels. Notez que les deux exemples donneront exactement le même résultat.

On pourrait donner les mêmes exemples avec padding.

Attention : veillez à toujours donner au moins une spécification des marges dans la classe « body ». La majorité des navigateurs Web donnent une valeur par défaut à ces marges si elles ne sont pas spécifiées. Sans cette manipulation, votre site Web risquerait fortement de provoquer des affichages différents suivants les navigateurs.

### Les évènements :

Nous avons vu comment décrire les éléments de page Web de façon statique : une fois la page chargée le style ne varie pas tant que l'on n'a pas changé de feuille CSS ou de page Web.

C'est là que les évènements entrent en jeu.

On distingue plusieurs évènements :

- hover : lorsque l'on passe le curseur sur la zone en question.
- visited : lorsque le lien hypertexte à déjà été cliqué au moins une fois.
- link : si la zone est un lien hypertexte.

Un évènement est précédé du la classe à laquelle il s'applique.

De cette manière, on peut personnaliser les liens, et tout autre élément d'une page Web. Voici un exemple de personnalisation de la classe « type1 » et des liens hypertextes :

```
type1:hover {
    color:#000;
}
a:link {
    color:#484;
}
a:visited {
    color:#484;
}
a:hover {
    color:#000;
}
```

### Le positionnement absolu :

Une manière couramment utilisée sur le Web pour découper une page consiste à disposer des calques via des positions absolues.

On utilisera l'attribut position: absolute.

De cette manière, il sera possible de spécifier les coordonnées du coin supérieur gauche d'un calque, l'origine de la page étant le coin supérieur gauche de la zone d'affichage du navigateur Web.

```
p.type1 {
    position: absolute;
    left: 10px;
    top: 9px;
}
p.type2 {
    position: absolute;
    right: 100px;
}
```

Dans cet exemple, la classe p.type1 sera positionnée à 10 pixels du bord gauche du navigateur, et à 9 pixels du haut.

La classe p.type2 sera positionnée) 100 pixels du bord droit du navigateur, la coordonnée verticale n'étant pas spécifiée.

Notez qu'avec de tels attributs, si deux calques se trouvent l'un sur l'autre, c'est le dernier utilisé dans le code HTML qui apparaîtra sur l'autre. On peut ainsi superposer différents calques sans aucune surprise.

## **C) WML**

Création d'une page WML afin de télécharger un < application java

Le tutoriel se découpera en deux parties :

- Création de la page WML
- Création du fichier htaccess

Création de la page WML

Voici l'architecture d'une page WML vierge :

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN" "
    http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
<card id="card1" title="Titre de la card">
<p align= "center">
<big><b><u>Titre de la page </u></b></big><br/>
<small>
    Corps de la page
</small>
</p>
</card>
</wml>
```

Les trois premières lignes correspondent à l'entête de la page. Je vous renvoie donc au paragraphe se trouvant dans la section technologie.

La balise <card > correspond a un écran du téléphone. Dans un fichier nous pouvons mettre plusieurs card a la suite. Ceci fera donc un site par niveau. Mais une card n'est pas une page Web. Donc pour notre projet, nous avons besoin simplement d'un seul écran. Maintenant nous allons modifier la page afin de pouvoir télécharger le jeu.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN" "
        http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
<card id="card1" title="JeuxPuissance4">
<p align= "center">
<big><b><u>Jeu du puissance4 </u></b></big><br/>
<small>
Ce site conçu pour WAP vous permet de télécharger notre jeu.<br/>
Pour ce faire, il suffit de cliquer sur
le lien se trouvant en bas de la page.<br/>
Voici le lien <br/>
<a href="puissance4.jad">puissance4</a><br/>
</small>
</p>
</card>
</wml>
```

Une fois la page terminée, il suffit de sauvegarder cette dernière avec l'extension .wml Passons maintenant au fichier htaccess

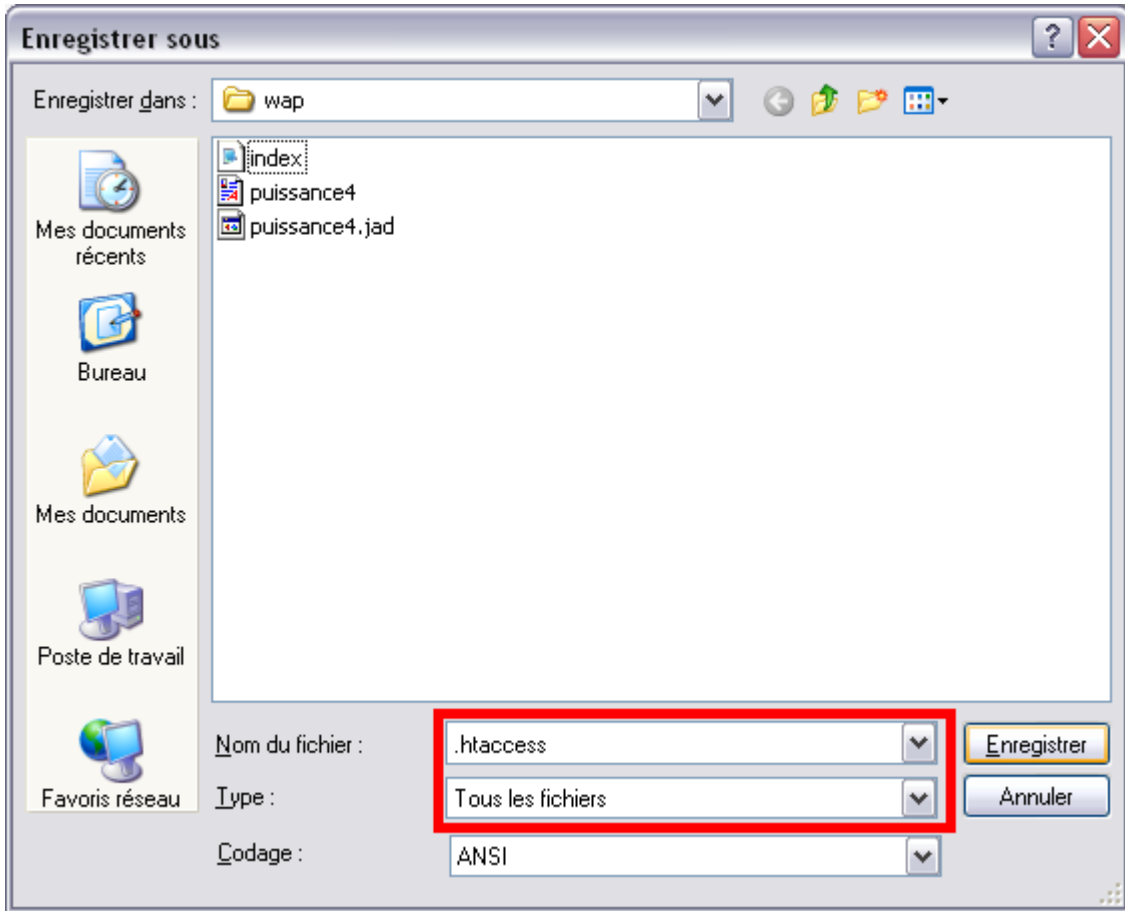
### Création du fichier htaccess

Les fichiers .htaccess sont des fichiers de configuration des serveurs Web Apache. Ils peuvent être placés dans n'importe quel répertoire du site Web (la configuration s'applique au répertoire et à tous les répertoires qu'il contient) et peuvent être modifiés alors que le serveur est en cours d'exécution.

Ce fichier est très important. Il va permettre d'autoriser les téléchargements du type fichier jar depuis votre serveur. Le fichier se nommera .htaccess voici ce qu'il faut écrire à l'intérieur :

```
AddType text/vnd.wap.wml wml
AddType text/vnd.wap.wmlscript wmls
AddType application/vnd.wap.wmlc wmlc
AddType application/vnd.wap.wmlscriptc wmlscriptc
AddType image/vnd.wap.wbmp wbmp
AddType application/java-archive jar
AddType text/vnd.sun.j2me.app-descriptor jad
```

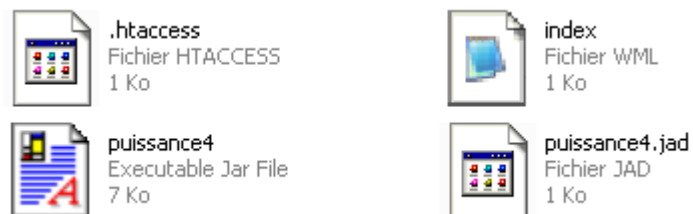
Attention: Pour enregistrer votre fichier. Il faut effectuer la démarche suivante :



Cette méthode est la seule qui fonctionne. En effet, sous Windows, il est impossible de renommer un fichier en .htaccess. Le système d'exploitation croit que le fichier ne possède pas de nom et celui-ci refuse tout fichier sans nom.

Résultat en images :

Voici ce que doit contenir votre dossier Wap :



Cholez Lionel, Kubicki Arnaud, Lermilage Jonathan   Projet JAVA / J2ME : Puissance4 2005/2006

Connectons nous sur l'adresse afin de récupérer le jeu du puissance 4. <http://www.bastion-empire.info/puissance4/wap/>

ou

<http://wap.bastion-empire.info/>



L

## 4) Conclusion

### A) Difficultés rencontrées

Durant le projet, les phases qui nous ont paru le plus difficiles sont :

- l'affichage d'images avec J2ME :  
==> il est impossible de charger et d'afficher des images dont le poids dépasse une certaine limite (quelques kilo-octets)
- le redimensionnement des images avec le J2ME :  
==> à l'instar de J2SE, J2ME ne propose aucune méthode permettant de redimensionner une image, nous avons donc dû en concevoir une
- le cas particulier des téléphones utilisant la technologie IMode :  
==> ce type de téléphone ne supporte ni l'application générée via J2ME, ni le site WAP classique. N'ayant ni le temps ni les ressources nécessaires pour concevoir une seconde version du projet entier pour IMode, nous sommes restés sous J2ME

### B) Le résultat

Nous avons finalement réalisé un jeu de puissance 4 pour téléphones portable, un jeu disposant de sa propre interface graphique et d'une bonne gestion des joueurs.

Nous avons aussi permis diverses améliorations futures :

- la gestion des joueurs permet une implémentation aisée d'une intelligence artificielle, celle-ci étant considérée comme un simple joueur par le jeu.
- la gestion de l'interface graphique peut être améliorée : plutôt que de reconstruire la grille et les jetons à chaque tour effectué, on pourrait – grâce à deux attributs (*i\_dernier\_x* et *i\_dernier\_y* de la classe *Jeu*) – seulement rajouter le dernier jeton joué et actualiser l'interface. On gagnerait ainsi en charge processeur, ce qui n'est pas négligeable pour un appareil aussi léger qu'un téléphone portable.
- Nous nous sommes retrouvés face à deux technologies concurrentes dans le domaine de la téléphonie mobile : IMode et les autres. IMode étant un cas particulier, nous nous sommes concentrés sur des technologies moins exotiques et davantage utilisées en Europe.

## 5) Annexes

### A) Dossier de programmation

*voir le dossier correspondant*

### B) Dossier du site Web

*voir le dossier correspondant*

### C) Manuel de l'utilisateur

*voir le dossier correspondant*