

IHM 2

PHIMP4



EIA-FR - Section informatique
Jonathan Sifringer
Luis Domingues
Fabien Yerly
Tiago De Deus
11.01.13

Table des matières

1	Introduction	1
2	But	1
3	Spécification.....	1
4	Règles du jeu.....	1
5	Mockups MyBalsamiq.....	1
5.1	MainView	2
5.2	Options	2
5.3	Network Multiplayer	3
5.4	SinglePlayer	3
5.5	GameView	3
5.6	Avis sur MyBalsamiq.....	4
6	Avis sur WindowBuilder.....	4
7	Avis sur Lauch4J et Inno Setup.....	4
8	Avis sur la création d'un paquet deb	7
9	Manuel d'utilisation.....	7
9.1	Page d'accueil.....	8
9.2	Jeu solo local	9
9.3	Jeu multi en ligne.....	9
9.4	Options	10
9.5	Jouer.....	11
10	Thèmes	11
10.1	Bannières	12
10.1.1	Default (thème par défaut).....	12
10.1.2	Zelda.....	12
10.1.3	Mario.....	12
10.1.4	Sonic.....	12
10.2	Tableaux de jeu et leurs boutons	13
10.2.1	Default (thème par défaut).....	13
10.2.2	Zelda.....	13
10.2.3	Mario.....	14
10.2.4	Sonic.....	14
11	Conclusion	15
11.1	Commentaires personnel	15
12	Déploiement	16

1 Introduction

Ce document décrit notre Projet IHM Puissance 4 (PIHMP4). Il contient les maquettes de l'application, un manuel d'utilisation ainsi que les points importants du développement.

2 But

Le but du projet est de concevoir une application selon les principes vus en cours. Nous avons choisi de réaliser un jeu. Grâce à ce dernier nous avons pu mettre en pratique l'utilisation de différents outils/composants comme par exemple le MigLayout et le SwingWorker.

3 Spécification

Notre application contiendra :

- L'internationalisation : Choix des langues en français, anglais ou allemand.

- Une interface utilisateur animée : Le tableau du jeu contient des mouvements animés (des pastilles qui « tombent »)

- L'utilisation de MigLayout pour toutes les fenêtres de l'interface.

- Quatre panels imbriqués dans la page de jeu pour que l'utilisateur puisse suivre le déroulement du jeu graphiquement et textuellement ainsi que les statistiques de chaque joueur.

- Deux patterns Observables pour mettre à jour la vue grâce aux notifications

- La possibilité de changer le thème de l'application.

- La possibilité de jouer de la musique

- L'utilisation du réseau pour jouer avec 2 ordinateurs différents

- L'utilisation du SwingWorker

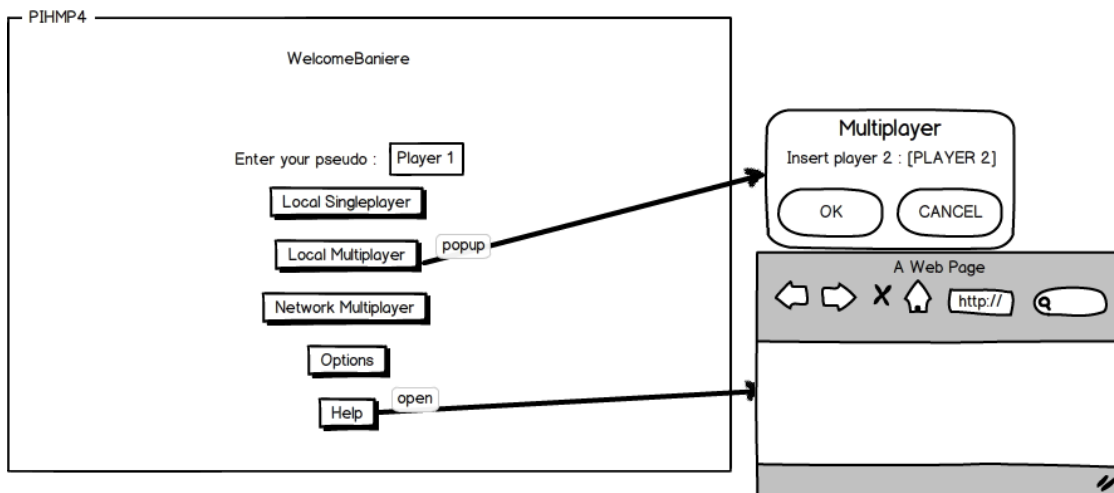
4 Règles du jeu

Le but du jeu est d'aligner 4 pastilles (horizontalement, verticalement et diagonalement) sur une grille de 6 lignes et 7 colonnes. Il se joue à 2 joueurs. À chacun son tour, un joueur place une pastille en cliquant sur une colonne de son choix, la pastille coulisse alors jusqu'à la position la plus basse possible et c'est ensuite à l'autre joueur de jouer. Le jeu se termine lorsque un joueur aligne 4 pastilles en ligne, en colonne ou en diagonale dans ce cas il gagne ou quand le plateau est plein et il y a ex aequo.

5 Mockups MyBalsamiq

Voici les maquettes de l'application faites avec l'application MyBalsamiq

5.1 MainView

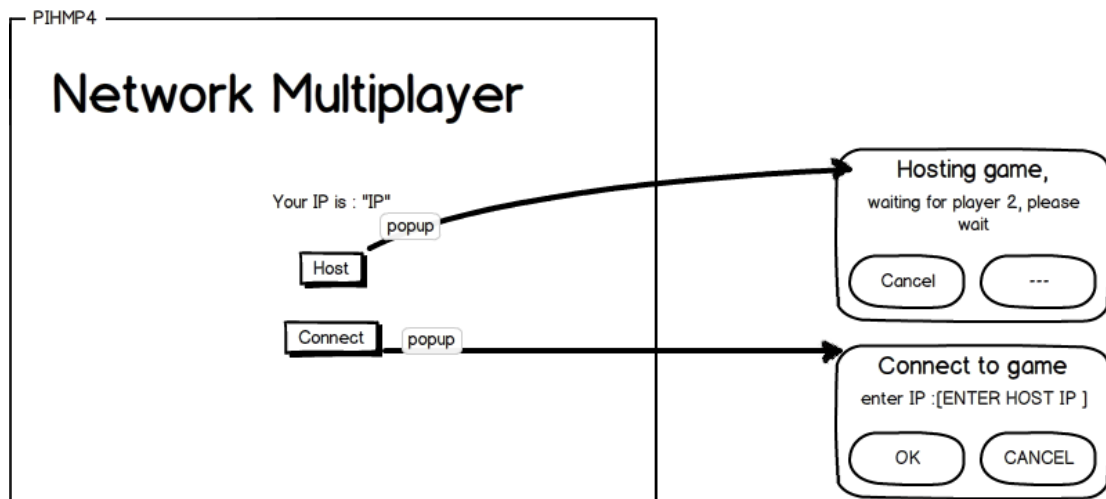


5.2 Options

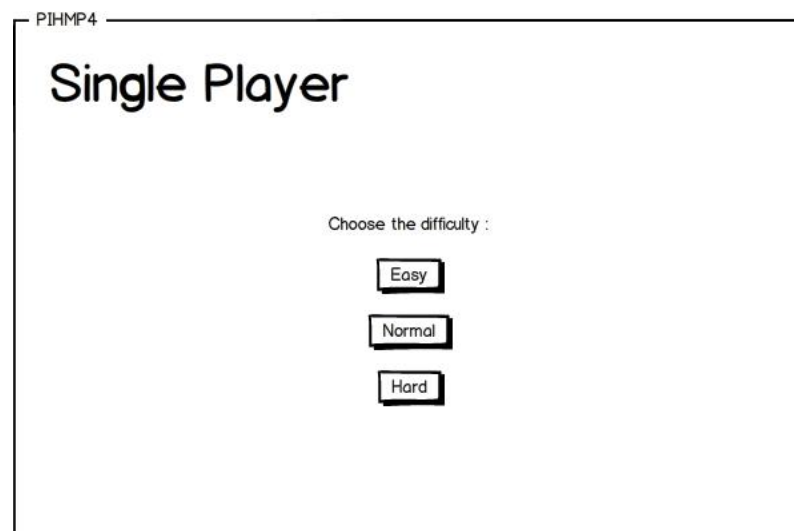
The Options interface (PIHMP4) displays the following settings:

- Options
- Language: ▼
- Theme: ▼
- Music: ☐
- Sounds: ☐

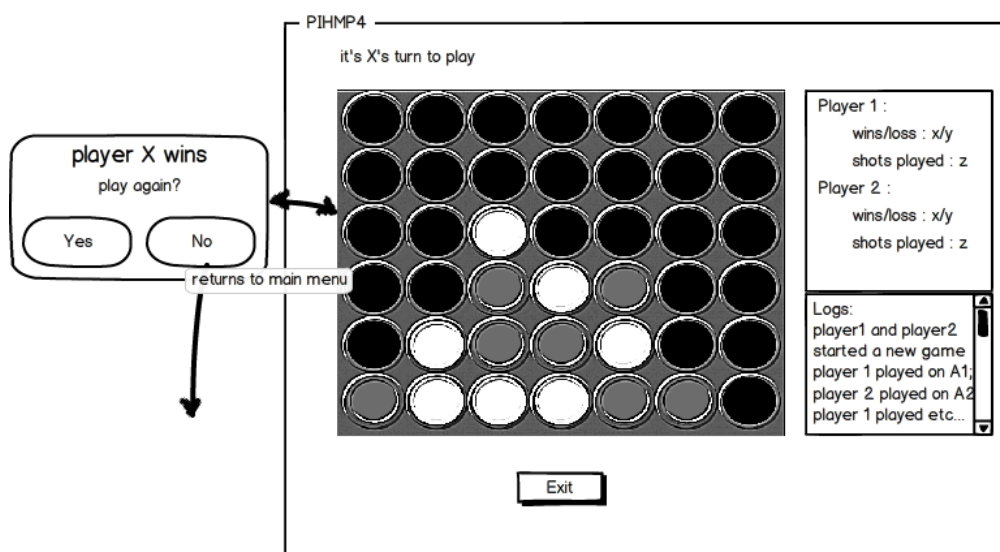
5.3 Network Multiplayer



5.4 SinglePlayer



5.5 GameView



5.6 Avis sur MyBalsamiq

Nous avons trouvé que les différents composants dans MyBalsamiq n'étaient pas très faciles à localiser. L'interface n'est pas intuitive à utiliser. MyBalsamiq, en tant que plugin est assez lent et lourd à utiliser. MyBalsamiq a quand même été à la hauteur de la création de maquette pour cette petite application mais nous pensons que pour des maquettes plus complexes il serait un peu trop limité.

6 Avis sur WindowBuilder

WindowBuilder est très pratique pour faire une première version de la fenêtre. Mais l'outil graphique est vite limité pour certaines options. Ce qui rend presque indispensable le fait d'aller modifier le code à la main (c'est ce que nous avons fait). Il permet un gain de temps significatif pour créer la première ébauche de la fenêtre. Le placement des différents composants est plus facile à réaliser ainsi qu'en codant. Comme nous avons décidé d'utiliser l'IDE Netbeans, il nous a été indispensable d'avoir Eclipse pour installer et utiliser WindowBuilder. Cette méthodologie nous a confirmé que le code est indépendant de la plateforme tant que les différentes bibliothèques sont présentes.

7 Avis sur Launch4J et Inno Setup

Ces 2 applications sont très très simples à utiliser et permettent de déployer une application java sur Windows.

Avec Launch4J il suffit de donner la version minimale requise de Java et le chemin du jar (éventuellement de l'icône) et le programme se charge de faire l'exé. C'est extrêmement pratique car il enregistre en même un xml de configuration qui permet d'automatiser la création des exé.

Le fichier xml que nous avons utilisé est celui-ci.

```
<launch4jConfig>
<dontWrapJar>>false</dontWrapJar>
<headerType>gui</headerType>
<jar>C:\Users\Tiag\Desktop\dist\PIHMP4.jar</jar>
<outfile>C:\Users\Tiag\Desktop\dist\pihmp4.exe</outfile>
<errTitle></errTitle>
<cmdLine></cmdLine>
<chdir></chdir>
<priority>normal</priority>
<downloadUrl>http://java.com/download</downloadUrl>
<supportUrl></supportUrl>
```

```
<customProcName>>false</customProcName>
<stayAlive>>false</stayAlive>
<manifest></manifest>
<icon>C:\Users\Tiag\Desktop\dist\pihmp4.ico</icon>
<jre>
<path></path>
<minVersion>1.7.0_09</minVersion>
<maxVersion></maxVersion>
<jdkPreference>preferJre</jdkPreference>
</jre>
<versionInfo>
<fileVersion>1.0.0.2</fileVersion>
<txtFileVersion>1.0</txtFileVersion>
<fileDescription>First Stable build of PIHMP4</fileDescription>
<copyright>EIA-FR & Croustillants</copyright>
<productVersion>1.0.0.2</productVersion>
<txtProductVersion>1.0</txtProductVersion>
<productName>PIHMP4</productName>
<companyName></companyName>
<internalName>pihmp4</internalName>
<originalFilename>pihmp.exe</originalFilename>
</versionInfo>
</launch4jConfig>
```

Pour inno setup il faut faire un script qui sera lu et créera le setup. Encore une fois cela permet d'automatiser la création des installations ce qui est très bien ! Le script en lui même est assez facile à comprendre et utilise des abréviations ce qui lui permet de rester près de l'anglais. De plus on peut définir des variables ou utiliser celles qui existent déjà.

```
; -- PIHMP4 Installation script --

#define MyAppName "PIHMP4"

#define MyVersion "1.0.0.2"

[Setup]

AppName={#MyAppName}
AppVersion={#MyVersion}
DefaultDirName={pf}\pihmp4
DefaultGroupName=pihmp4
UninstallDisplayIcon={app}\pihmp4.exe
Compression=lzma2
SolidCompression=yes
OutputDir=Output
OutputBaseFilename=pihmp4_setup_{#MyVersion}
VersionInfoVersion={#MyVersion}

[CustomMessages]

AppName=PIHMP4
LaunchProgram=Start PIHMP4 after finishing installation

[Files]

Source: "pihmp4.exe"; DestDir: "{app}"

Source: "README.TXT"; DestDir: "{app}"; Flags: isreadme

Source: "miglayout-4.0-swing.jar"; DestDir: "{app}\lib"

[Icons]

Name: "{group}\pihmp4"; Filename: "{app}\pihmp4.exe"
```



```
Name: "{commondesktop}\pihmp4"; Filename: "{app}\pihmp4.exe"
```

```
[Run]
```

```
Filename: {app}\pihmp4.exe; Description: {cm:LaunchProgram,{#MyAppName}}; Flags: no-  
wait postinstall skipifsilent
```

En conclusion nous trouvons ces 2 utilitaires très pratiques.

8 Avis sur la création d'un paquet deb

La création d'un paquet Debian est vraiment simple. Il suffit de créer, dans un répertoire « projet » arborescence système avec nos fichiers. On n'y met que les dossiers utiles. Une fois que tous les fichiers du programme y sont, nous y créons le dossier « DEBIAN » avec à l'intérieur plusieurs fichiers.

- Les scripts d'installation (Optionnel)
- Le fichier contrôle

Le fichier contrôle est très important. Il ressemble à ceci :

```
Package: pihmp4  
Version: 1.0-1  
Section: base  
Priority: optional  
Architecture: all  
Depends: bash  
Maintainer: maintenir <maintener @mail.com>  
Description: Puissance 4  
Homepage: http://outadoc.fr
```

Une fois ceci, il faut s'assurer que le dossier de projet ai un nom qui suit cette directive :

```
<foo>_<VersionNumber>-<DebianRevisionNumber>_<DebianArchitecture>.deb
```

Puis on lance le programme dpkg pour créer le paquet :

```
dpkg-deb --build mydeb
```

Et voila, un paquet Deb installable sous les Debian et dérivés (Ubuntu, etc).

9 Manuel d'utilisation

Voici la partie qui explique les différentes fenêtres et leurs utilités.

9.1 Page d'accueil



Ceci est la page principale du jeu. C'est ici que nous pouvons choisir le mode de jeu que nous voulons, modifier les options ou consulter l'aide.

Jeu solo local : jouer seul contre un adversaire virtuel (AI)

Jeu multi local : jouer contre un adversaire humain sur le même ordinateur

Jeu multi en ligne : jouer contre un adversaire humain sur 2 machines différentes en passant par le réseau

Options : Modifier les options de l'application (thèmes, langue, activer/désactiver la musique).

Aide : L'aide est sensible à l'internationalisation. Une aide sous forme de page « .html » dans une fenêtre dédiée.

9.2 Jeu solo local



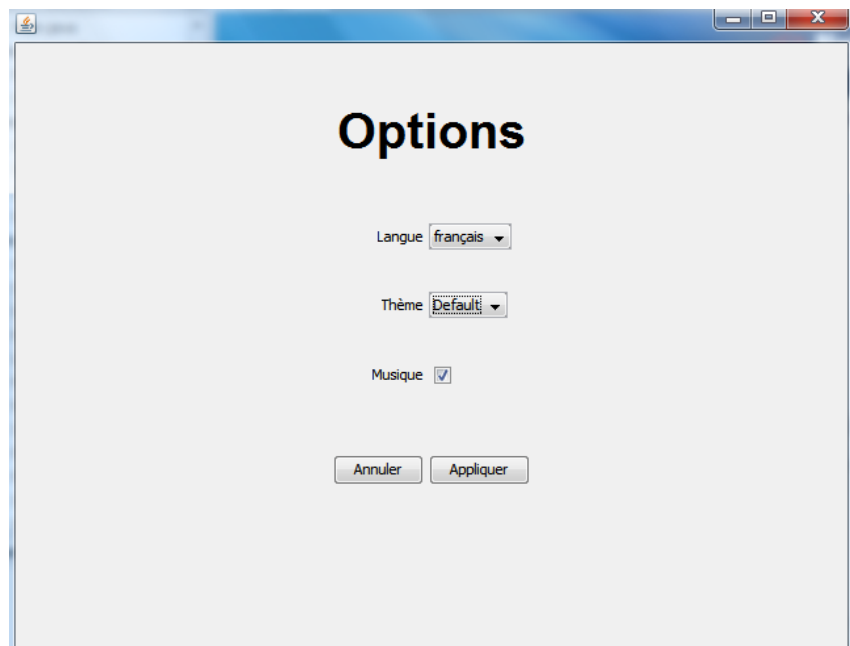
Ici, nous devons simplement choisir le niveau difficulté de l'IA lorsque nous jouons avons choisi Jeu solo local.

9.3 Jeu multi en ligne



Pour pouvoir jouer à deux joueurs en ligne, il faut que les deux joueurs aient installé l'application. Un des joueurs commence par cliquer sur le bouton « Héberger ». Le bouton « Retour » devient « Annuler » et les deux autres boutons seront grisés. L'autre joueur devra aussi aller en mode « jeu multi en ligne » mais cliquer sur « Connecter » une fois que l'autre ai cliqué sur « Héberger ». Une fenêtre de dialogue lui demander d'entrer l'adresse IP du serveur qui est indique en haut de la fenêtre du joueur qui héberge. Une fois connecté le jeu démarre.

9.4 Options



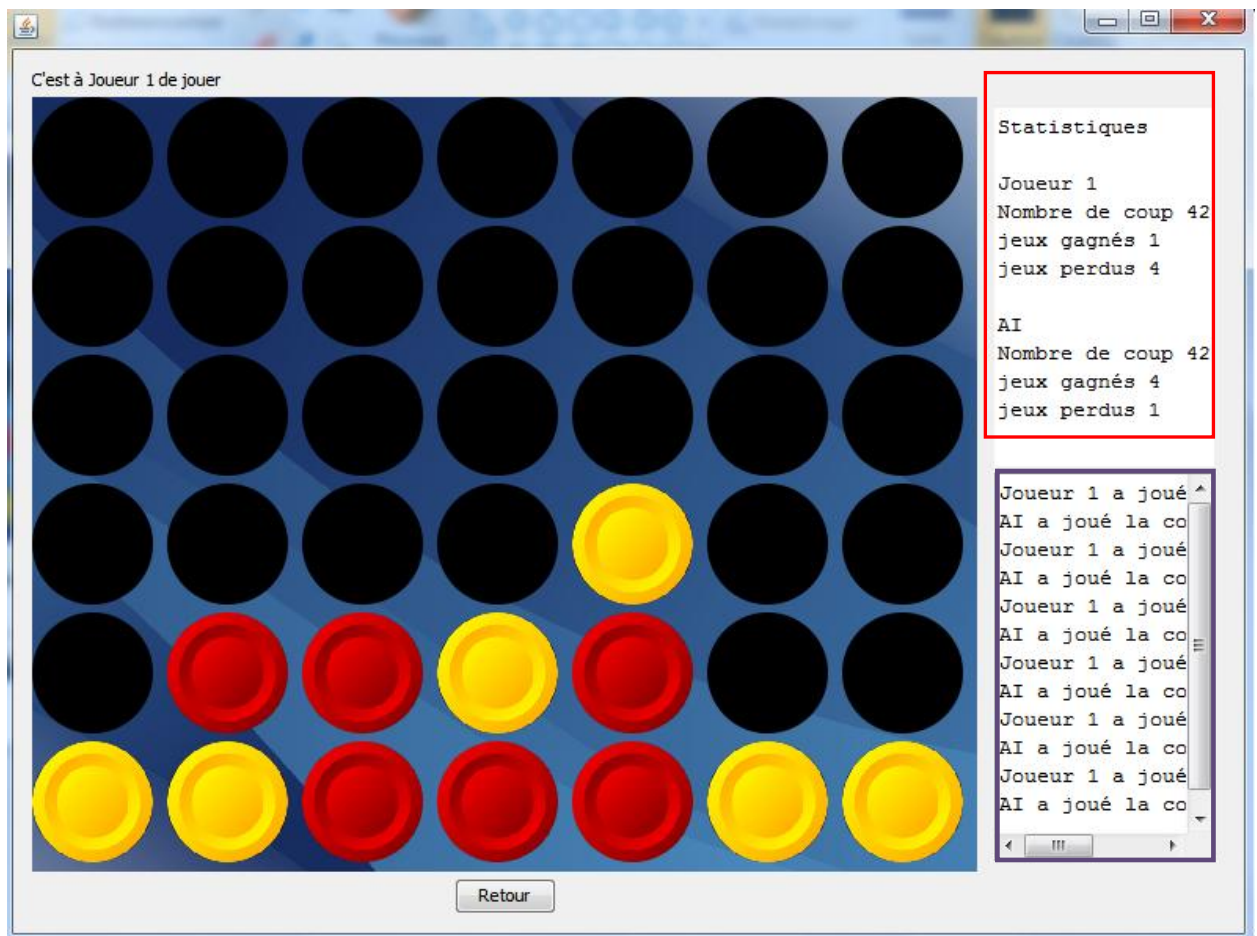
Cette fenêtre nous permet de modifier différentes options.

Langue : Choisir la langue de l'application. Nous pouvons choisir, français , anglais ou allemand.

Thème : Nous pouvons choisir entre plusieurs thèmes pour l'application. Cela va changer la bannière de la page d'accueil, le thème de la bannière et jouer une musique pendant le jeu qui correspond au thème choisis.

Musique : Activer ou désactiver la musique.

9.5 Jouer



Cette fenêtre est le jeu puissance 4. Nous voyons au centre en bleu le tableau du jeu. Pour jouer, l'utilisateur doit cliquer sur la colonne où c'est qu'il veut déposer sa pastille.

En rouge, Statistiques des nombres de coups jouer pour chaque joueur et les jeux gagnés et perdu.

En violet, des logs pour savoir qui est-ce qui a joué à quelle colonne. Les défaites et les victoires sont aussi loggées.

10 Thèmes

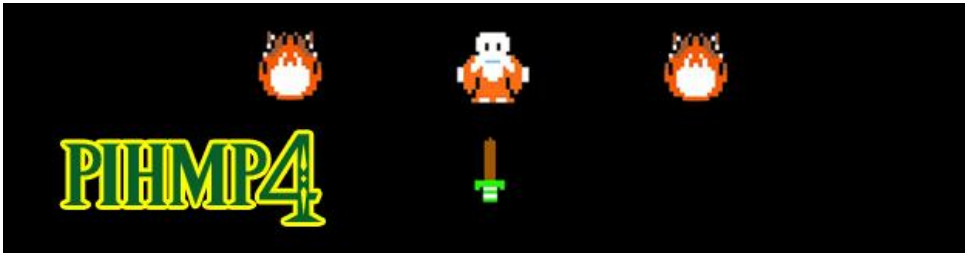
Nous avons de la musique différente, des bannières différentes et des tableaux différents pour chaque thème. Voici les différentes bannières et tableaux selon le thème choisi.

10.1 Bannières

10.1.1 Default (thème par défaut)



10.1.2 Zelda



10.1.3 Mario

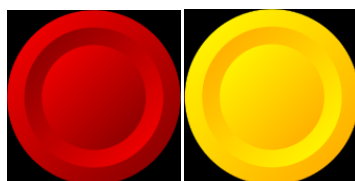
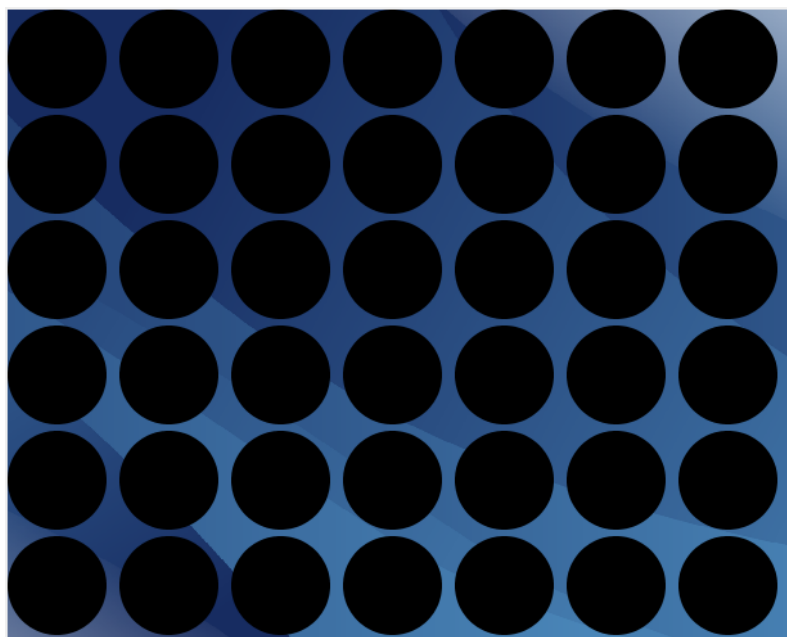


10.1.4 Sonic

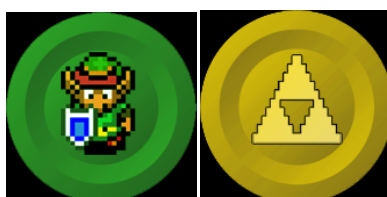
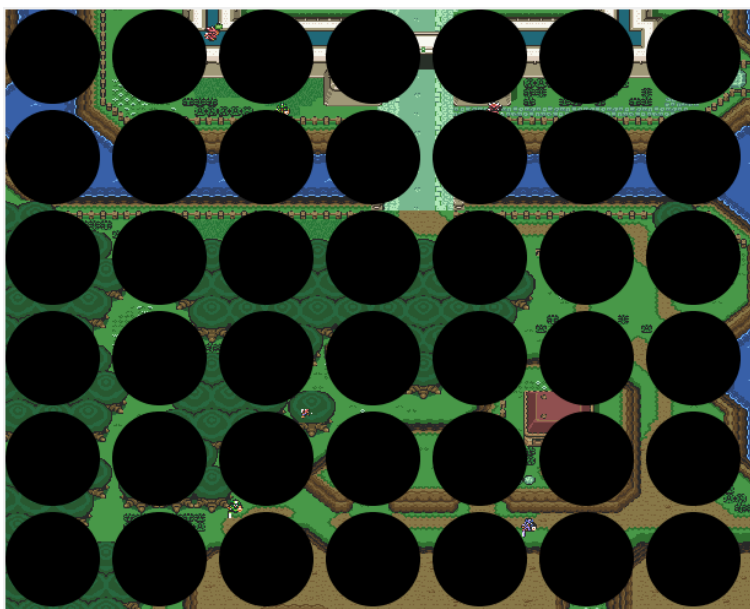


10.2 Tableaux de jeu et leurs boutons

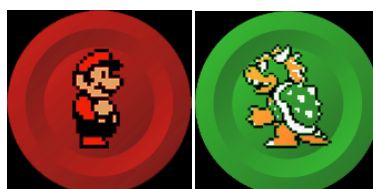
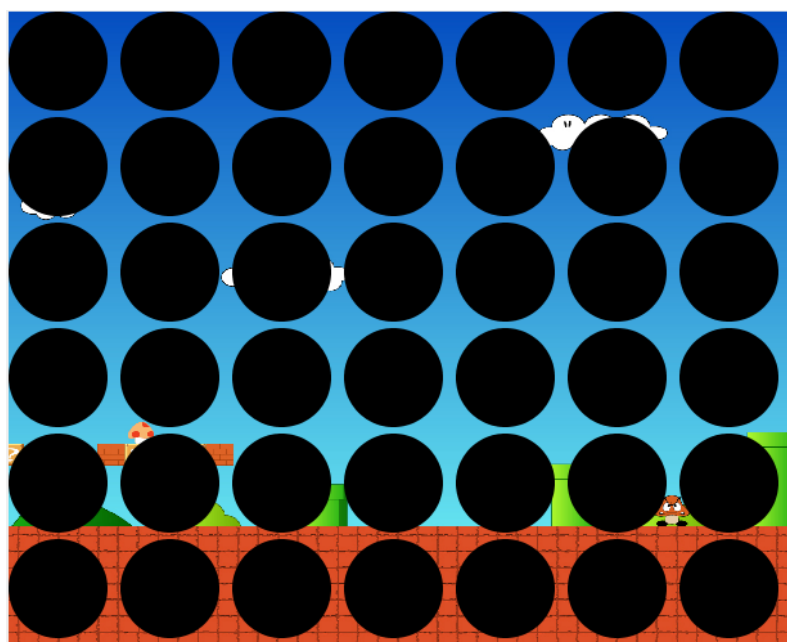
10.2.1 Default (thème par défaut)



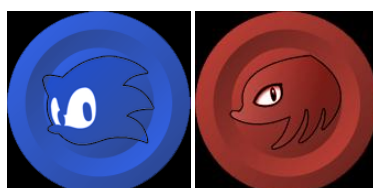
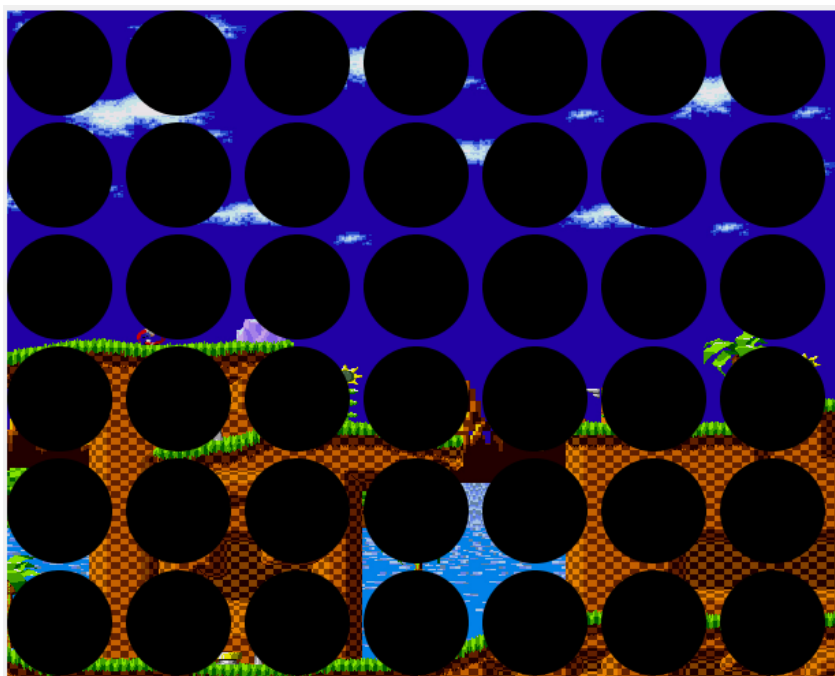
10.2.2 Zelda



10.2.3 Mario



10.2.4 Sonic



11 Conclusion

Durant ce TP nous avons vu comment faire utiliser Java 2D pour créer des vues interactives. Nous avons également pu constater qu'il n'est pas simple de faire des animations et d'utiliser un MigLayout. Cependant, MigLayout est, une fois compris et pris en main, puissant et modulable.

Durant ce projet nous avons également utilisé des outils tels que ceux vu en cours de génie logiciel afin de se mettre d'accord au départ avant l'implémentation. Nous l'avons utilisé pour structurer et affecter les tâches que nous avons à faire. Ce qui nous a permis un gain de temps assez significatif.

11.1 Commentaires personnel

Jonathan :

Je trouve que ce TP s'est très bien déroulé et que le résultat final est remarquable. On voit qu'on devient de plus en plus à l'aise à coordonner le travail de chacun et assigner des tâches différentes à tout le monde. Je trouve que la liberté donnée quand au choix de l'application à réaliser une bonne source de motivation. C'est la première fois que je réalise une application qui contient une interface graphique dynamique et utile en Java.

Luis :

Ce projet a été, de mon point de vue, très intéressant et instructif. Le fait d'avoir quelque chose où l'on peut faire ce que l'on veut apporte un peu d'oxygène au milieu de tout les TP et mini-projet des autres cours. Le travail de groupe est très agréable. Et comme le projet est vide au départ, il est très intéressant d'avoir différentes idées de conceptions aux débuts. C'est également aux seins de ce projet que l'on peut constater quelques lacunes dans notre savoir. Bien que nous sachions programmer, nous avons à peine appris à sortir une application de la plate-forme de développement et encore moins à créer un installateur. C'est une chose importante que nous avons put apercevoir dans le cadre de ce projet.

Tiago :

J'ai beaucoup apprécié ce projet. Avec tous les travaux que nous avons eu à faire ce semestre, c'est vraiment rafraîchissant pour une fois d'avoir un projet « libre » où on peut laisser parler nos envies. Les restrictions imposées s'intégrant naturellement au développement d'une application ça n'as pas gênée notre motivation. C'était au contraire extrêmement intéressant de voir comment on code au final toutes ces choses qui sont normalement nécessaire à une bonne application mais qu'on ne fait jamais à l'école. C'est d'ailleurs pour ça que je me suis intéressé de plus près au déploiement de l'application bien que c'était facultatif.

Au final nous avons une application qui marche (avec quelques bug) et avons réussi à intégrer toutes les fonctions que nous voulions. Je suis donc très content de notre groupe et de notre synergie.

Fabien :

Ce projet ma beaucoup plus. Bien que, contrairement à mes collègues, je n'ai pas une multitude de projet ce semestre car étant en PFP, j'ai tout de même apprécié le fait d'être « libre ». Il est vrai qu'il y avait des contraintes mais celle-ci font que nos connais-

sances et notre façon de travaillé évolue et nous apportent plus de connaissance. Personnellement j'ai trouvé la cohésion du groupe bonne.

12 Déploiement

Pour notre application, nous avons prévu plusieurs moyens de déploiement. Il faut avoir installé au préalable java 1.7 sur sa machine.

Version Zip (pour n'importe quel système) contenant :

- Executable java (.jar)
- Dossier lib avec la librairie MigLayout
- Un fichier readme
- le rapport (ce document)

Version Windows : https://www.dropbox.com/s/ppcqaw4hp511o8/pihmp4_setup_1.0.0.2.exe

Fichier .exe de setup qui installe l'application

Version Debian et dérivés : https://www.dropbox.com/s/64m88rzo83voln5/pihmp4_1.0.0.2-1_amd64.deb

Fichier .deb de setup qui installe l'application