

République Algérienne Démocratique et Populaire  
Université Abou Bakr Belkaid– Tlemcen  
Faculté des Sciences  
Département d'Informatique

Mémoire de fin d'études

Pour l'obtention du diplôme de Licence en Informatique

*Thème*

**Conception et implémentation d'une application  
java cas de gestion du code de la route**

Réalisé par :

- KHEDIM ABDELKADIR
- YAHLA MOHAMMED REDA

*Présenté le 9 Juin 2014 devant la commission d'examination composée de MM.*

- Mr CHOUITI SIDI MOHAMMED (Encadreur)
- Mr SMAHI MOHAMMED ISMAIL (Examineur)
- Mm ILES N. (Examineur)

Année universitaire : 2013-2014

# REMERCIEMENTS

*Avant tout, je remercie mon Dieu qui m'a donné la force, le*

*Courage et la volonté pour achever ce modeste travail, et je dis*

*« Al hamdoulillah ».*

*Qu'il nous est permis de remercier tous ceux qui nous ont encouragés, soutenus et aidés à réaliser ce projet.*

*Nous remercions tout d'abord tous les professeurs de la faculté de sciences des ingénieurs pour toutes les informations qu'il nous a données durant notre période d'étude universitaire.*

*Nous adressons nos sincères mercis au chef de notre département, ainsi que notre encadreur Mr CHOUITI .S pour toutes les informations et tous les conseils qu'il nous a donné durant notre projet de fin d'étude.*

*Enfin, nous remercions tous les nombre du service d'enseignement pour l'aide précieuse et le service personnel et la secrétaire de département d'informatique*

*Je dédie ce modeste travail à :*

**A tous ceux qui, de près ou de loin, ont participé à mon  
éducation, m'ont aidé dans les moments difficiles à sur  
Monter mes problèmes.**

**A ceux qui ont partagé avec moi mes bonheurs et mes  
Soucis.**

**A mes très chères parents pour leurs sacrifices et leur  
Soutien durant toutes mes années d'étude.**

**A mon chère frère AMINE.**

**A mes très chères sœurs.**

**Et toute la famille « YAHLA ».**

**A tous mes amis de plus proche au plus loin,  
Abdelkrim, Abdelkader, Samad, Benaouda, Fathi,...et les  
autre chacun de**

**Son nom.**

**A ma promotion «LMD INFORMATIQUE 2013/2014 »**

**YAHLA MOHAMMED REDA**

*Je dédie ce modeste travail à :*

**A tous ceux qui, de près ou de loin, ont participé à mon éducation, m'ont aidé dans les moments difficiles à sur**

**Monter mes problèmes.**

**A ceux qui ont partagé avec moi mes bonheurs et mes**

**Soucis.**

**A mes très chères parents pour leurs sacrifices et leur**

**Soutien durant toutes mes années d'étude.**

**A mes chères frères Abdelkrim et Mohammed**

**A ma chère sœur.**

**Et toute la famille « khedim ».**

**A tous mes amis de plus proche au plus loin,**

**Reda, Abdelkrim, Ahmed, Ilyes , Fayçal ,Hafid ,samad ,mouad  
,snoussi**

**Et les autre chacun de**

**Son nom.**

**A ma promotion «LMD INFORMATIQUE 2013/2014 »**

**KHEDIM ABDELKADIR**

## *Table des matières*

---

Liste des figures.....	1
Introduction Générale.....	3
<b>Chapitre 1 : Code de la route et conception de l'application</b>	
1 Définition du code de la route.....	4
1.1 Que peut contenir un code de la route?.....	4
2 Les types des signaux définis a l'échèle mondiale .....	5
2.1 Signaux d'avertissement de danger : .....	5
2.2 Signaux de réglementation : .....	5
2.3 Signaux d'indication : .....	5
3 Les caractéristiques des signaux .....	6
4 Panneau de signalisation routière.....	7
4.1 Structure d'un panneau.....	8
4.2 Constituants du panneau .....	8
4.3 Dimensions .....	9
4.4 Position.....	9
1 Le langage UML .....	11
1.1 Définition d'UML .....	11
1.2 Les point forts d'UML .....	11
1.3 Les points faibles d'UML.....	11
1.4 UML est un standard .....	12
1.5 UML est un langage de modélisation et pas une méthode .....	12
1.6 Les digrammes.....	12
2 Conception .....	12
2.1 Le processus unifié (UP) .....	12
2.1.1 Élaborer les cas d'utilisation.....	13
2.1.2 Diagrammes de séquences.....	13
2.1.3 Diagramme de classe.....	14
2.2 Diagramme des paquetages.....	14

## ❖ **Chapitre 2 : Outils utilisés**

1 Présentation du langage JAVA.....	18
-------------------------------------	----

1.1	Introduction.....	18
2	Particularité du langage java .....	18
3	Fichier source JAVA .....	20
4	Objectif du langage java .....	21
5	Environnements de développement.....	21
6	Présentation de IDE éclipse.....	21
6.1	Définition de IDE éclipse.....	21
6.2	Workspace? .....	22
6.3	Architecture d'éclipse.....	22
6.4	Le noyau 'Eclipse Project' .....	23
6.5	Eclipse Platform.....	24
6.6	Concurrents .....	24
7	Les bibliothèques utilisées .....	24
7.1	Abstract window toolkit .....	24
7.2	Swing .....	25
	<b>❖ <u>Chapitre 3 : Application</u></b>	
1	Premier interface de l'application.....	26
1.1	Le logo d'entré à l'application .....	26
2	Menu pricipale .....	27
3	Menu de types des Panneaux .....	30
3.1	Avant animation .....	30
3.2	Avec animation.....	31
4	Menu des panneaux.....	32
5	Menu de définition des panneaux .....	34
6	Menu exercice .....	35
6.1	L'interface.....	35
6.2	Correction.....	37
6.3	Notation.....	38
	Conclusion Générale.....	40
	Références bibliographiques.....	41
	Résumé.....	42

### 1

#### ❖ Chapitre 1 : Code de la route et conception de l'application

Figure I.1 : différents Constituants du panneau .....	9
Figure I.2 : digramme de cas d'utilisation .....	13
Figure I.3 : digramme de paquetages.....	14
Figure I.4 : diagramme de composant du paquetage Exercice .....	14
Figure I.5 : diagramme de composant du paquetage Deuxemeessai .....	15
Figure I.6 : diagramme de composant du paquetage MenuDesPanneaux .....	16
Figure I.7 : diagramme de composant du paquetage MenuDefinition.....	17

#### ❖ Chapitre 2 : Outils utilisés

Figure II.1 cycle d'exécution d'un fichier source écrit en langages compilé traditionnels .....	19
Figure II.2 : cycle d'exécution d'un fichier source écrit en langage java .....	20

#### ❖ Chapitre 3 : Application

Figure III.1 : logo 1 .....	26
Figure III.2 : logo 2 .....	26
Figure III.3 : menu principale.....	27
Figure III.4 menu principale arabe.....	28
Figure III.5 message de coordonnées.....	29
Figure III.6 message de confirmation .....	29
Figure III.7 menu des panneaux .....	30
Figure III.8 menu des panneaux arabe .....	30
Figure III.9 menu de type des panneaux animé .....	31
Figure III.10 menu de type des panneaux arabe animé .....	31
Figure III.11 menu des panneaux (cas danger).....	32
Figure III.12 menu des panneaux arabe (cas danger) .....	32
Figure III.13 menu des panneaux (cas interdiction) .....	33
Figure III.14 menu des panneaux arabe (cas interdiction).....	33
Figure III.15 menu de définition des panneaux.....	34

Figure III.16 menu de définition des panneaux arabe.....	34
Figure III.17 menu d'exercice :interface.....	35
Figure III.18 menu d'exercice arabe :interface.....	36
Figure III.19 menu d'exercice : correction.....	37
Figure III.20 menu d'exercice arabe : correction.....	38
Figure III.21 menu d'exercice : notation.....	38
Figure III.22 menu d'exercice arabe : notation .....	39

# Introduction Générale

## Introduction Générale

Actuellement le monde connaît une avance technologique considérable dans tous les secteurs et cela grâce à l'informatique qui est une science qui étudie les techniques de traitement automatiques de l'information.

L'informatique est de plus en plus présente dans notre vie quotidienne car elle offre à toutes les autres disciplines un outil puissant pour la recherche d'informations, leur exploitation et leur présentation. Comme par exemple grâce à l'informatique un grand nombre de logiciels permettant l'aide à l'apprentissage dans des domaines divers existent actuellement ce qui facilite énormément plusieurs tâches.

L'objectif de notre projet présenté dans ce mémoire est la conception et implémentation d'une application Java qui permet l'apprentissage du code de la route aux candidats au examen de permis de conduire.

Cette application propose deux parties :

- Partie panneaux qui permet l'apprentissage de tous les types de panneaux
- Partie exercice qui permet d'évaluer notre connaissance sur les panneaux

Donc on peut dire qu'il s'agit d'un **didacticiel** (logiciel interactif destiné à l'apprentissage des savoirs sur un thème ou un domaine donné et incluant généralement un autocontrôle de connaissance) en plus d'un **exerciceur** (logiciel réalisé à base d'exercices d'entraînement)

Nous avons organisé ce mémoire de la façon suivante :

- Le premier chapitre est consacré à la présentation du code de la route et la partie de conception il s'achève par un diagramme de cas d'utilisation
- Le deuxième chapitre consiste à définir tous les outils utilisés lors du développement de l'application
- Le troisième chapitre consiste à détailler tous les interfaces de notre application pour comprendre en mieux leur fonctionnalités.

# Chapitre 1 –Code de la route et conception de l’application

---

Ce chapitre est scindée en deux partie la première consiste à introduire brièvement les notions de base du domaine du code de la route. Puisque le sujet de notre PFE est une application qui aide à l’apprentissage du code de la route, et la deuxième partie consiste à la conception de notre application.

## Partie I –Code de la route

### 1 Définition du code de la route

Le **code de la route** est l'ensemble des lois et règlements relatifs à l'utilisation des voies publiques (trottoirs, chaussées, autoroutes, etc.) par les usagers (piétons, cyclistes, deux-roues à moteur, automobilistes, etc.). Il s’agit dans certains pays francophones d'un code législatif, au même titre que le code de l'environnement ou le code pénal, mais il désigne aussi plus vaguement la combinaison de lois, réglementations et règles de civisme et de savoir-vivre.

Le non-respect des dispositions du code de la route expose le contrevenant à des sanctions de divers niveaux (de l'amende à la peine de prison) selon la gravité de l'infraction.

#### 1.1 Que peut contenir un code de la route?

Un code de la route peut contenir des règles relatives :

- aux qualités ou compétences qu'un usager doit posséder selon l'usage qu'il fait de la voie publique (permis de conduire, âge minimum, examen médical)
- aux caractéristiques qu'un véhicule doit posséder (homologation des véhicules, contrôle technique régulier, couverture par assurance de la responsabilité du conducteur, ...)
- à la façon dont un usager peut utiliser la voie publique (rouler à droite ou à gauche, limitations de vitesse, stationnement, présence de frein et de bandes réfléchissantes sur les pédales, ...)
- à la façon dont les usagers interagissent entre eux (règles de priorité par exemple)
- à la façon dont les autorités peuvent imposer des règles locales sur certaines portions de voie publique (signalisation, policier, ...)

Le code de la route n'est pas un instrument figé, puisqu'il variera selon la perception des autorités et de la population.

## 2 Les types des signaux définis a l'échèle mondiale

Il existe trois types de signaux définit comme suit :

### 2.1 [Signaux d'avertissement de danger](#) :

Ces signaux ont pour objet d'avertir les usagers de la route de l'existence d'un danger sur la route et de leur en indiquer la nature.

### 2.2 [Signaux de réglementation](#) :

Ces signaux ont pour objet de notifier aux usagers de la route les obligations, limitations ou interdictions spéciales qu'ils doivent observer ; ils se subdivisent en :

- Signaux de priorité
- Signaux d'interdiction
- Signaux d'obligation
- Signaux de prescriptions particulières

### 2.3 [Signaux d'indication](#) :

Ces signaux ont pour objet de guider les usagers de la route au cours de leurs déplacements ou de leur fournir d'autres indications pouvant leur être utiles, ils se subdivisent en :

- Signaux d'information, d'installation ou de service
- Signaux de direction, de jalonnement ou d'indication
  - Signalisation avancée ou pré signalisation
  - Signaux de direction
  - Signaux d'identification des routes
  - Signaux de localisation
  - Signaux de confirmation
  - Signaux d'indication
- Panneaux additionnels

# Chapitre 1 – Code de la route et conception de l'application

## 3 Les caractéristiques des signaux

Le tableau I.1 donne les caractéristiques des principaux signaux

Type de panneau	Forme	Fond	Bord	Taille	Symbole
<b>Signaux d'avertissement de danger</b>					
Signal de danger	<a href="#">Triangle équilatéral</a>	<a href="#">Blanc</a> ou <a href="#">jaune</a>	<a href="#">Rouge</a>	0,9 m(normal) 0,6 m (petit)	Divers, <a href="#">noir</a> ou <a href="#">bleu</a> foncé
	<a href="#">Losange</a> (carré dont une diagonale est verticale)	Jaune	Noir	0,6 m (normal), 0,4 m (petit)	Divers, noir ou bleu foncé
<b>Signaux de priorité</b>					
<a href="#">Cédez-le-passage</a>	Triangle équilatéral inversé	Blanc ou jaune	Rouge	0,9 m (normal), 0,6 m (petit)	Aucun
<a href="#">Signal d'arrêt</a>	<a href="#">Octogone</a>	Rouge	Aucun	0,9 m(normal) 0,6 m(petit)	<i>Stop</i> , écrit en blanc
	<a href="#">Circulaire</a>	Blanc ou jaune	Jaune	0,9 m (normal), 0,6 m (petit)	<i>Stop</i> , écrit en bleu ou Noir
<a href="#">Route prioritaire</a>	Losange	Blanc	Noir	0,5 m (normal), 0,35 m (petit)	Carré jaune
Fin de priorité	Losange	Blanc	Noir	0,5 m (normal), 0,35 m (petit)	Carré jaune avec une série de traits noirs ou gris ou une ligne diagonale noire traversant le signal.
Priorité à la circulation venant en sens inverse	Circulaire	Blanc ou Jaune	Rouge	Non spécifié	la flèche indiquant le sens prioritaire est noire et celle qui indique l'autre sens est rouge.
Priorité par rapport à la circulation venant en sens inverse	<a href="#">Rectangle</a>	bleu	Aucun	Non spécifié	la flèche dirigée vers le haut est blanche, l'autre rouge.

### Signaux d'interdiction ou de restriction

Interdiction d'accès	Circulaire	Blanc ou Jaune	Rouge	0,6 m (normal), 0,4 m (petit)	Divers
Interdiction de stationnement	Circulaire	Bleu	Aucun	0,6 m (normal), 0,2 m (petit)	Divers
Fin d'interdiction	Circulaire	Blanc ou Jaune	Aucun	0,6 m (normal), 0,4 m (petit)	Noir ou ligne diagonale grise

### Signaux d'obligation

Direction obligatoire	Circulaire	bleu	Aucun	0,6 m (normal) 0,4 m (petit)	Divers, Blanc
	Circulaire	Blanc	Rouge	0,6 m (normal) 0,4 m (petit)	Divers, Noir

### Signaux de prescriptions particulières

Tous signaux	Rectangulaire	Bleu	Non spécifié	Non spécifié	Divers, Blanc
		Lumineux			Divers, noir

### Signaux d'information, d'installation ou de service

Signaux d'indication	Rectangulaire, quelquefois avec une flèche	Lumineux	Non spécifié	Non spécifié	Divers, sombre
		Sombre			Divers, lumineux
<a href="#">Autoroutes</a>	Rectangulaire	bleu ou vert			Divers, Blanc
Temporaire		Jaune ou range	Divers, Noir		

### Panneaux additionnels

Tous panneaux	Non spécifié	Blanc ou Jaune	Noir, bleu ou Rouge	Non spécifié	Divers, noir ou bleu foncé
		Noir ou bleu foncé	Blanc ou jaune		Divers, blanc ou Jaune

Tableau I.1 : les caractéristiques des signaux

## 4 Panneau de signalisation routière

Les **panneaux de signalisation routière** sont des éléments de la signalisation routière. Ils désignent à la fois le dispositif sur lequel est implanté un signal routier et le signal proprement dit.

# Chapitre 1 –Code de la route et conception de l'application

---

Placés sur le côté des routes, ils ont pour fonction

- de rendre plus sûre la circulation routière en informant les usagers d'éventuels dangers qu'ils peuvent rencontrer
- de faciliter cette circulation, en indiquant par exemple les directions à suivre
- d'indiquer ou de rappeler diverses prescriptions particulières de police
- de donner des informations relatives à l'usage de la route

## 4.1 Structure d'un panneau

Un panneau de signalisation, compris en tant que dispositif porteur d'un signal routier, est fixé sur un support qui est lui-même fiché dans le sol dans un massif de fondation ou ancrage.

## 4.2 Constituants du panneau

Le panneau proprement dit est constitué des éléments suivants :

- Le subjectile : matériau servant d'appui aux matériaux de façade des panneaux, qu'ils soient catadioptriques ou non.
- Le bord protecteur : Bord du panneau, pouvant être constitué d'un élément séparé du corps du panneau, conçu pour rigidifier le panneau et réduire la gravité des blessures en cas d'impact physique sur celui-ci
- La face de panneau : Il s'agit du ou des matériaux(x) appliqué(s) sur le subjectile pour produire le signal du panneau.
- Le décor est l'élément graphique, le pictogramme ou le texte qui visualise le message

La figure I.1 illustre les différents Constituants du panneau



Figure I.1 : différents Constituants du panneau

## 4.3 Dimensions

Les dimensions des **panneaux de signalisation** doivent être telles que le signal soit facilement visible de loin et facilement compréhensible quand on s'en approche. Elles tiennent compte de la vitesse usuelle des véhicules

Chaque pays définit sa propre réglementation concernant les dimensions de panneaux. En règle générale, il y a quatre catégories de dimensions pour chaque type de signal : petites, normales, grandes et très grandes dimensions

Les signaux de petites dimensions sont employés lorsque les conditions ne permettent pas l'emploi de signaux de dimensions normales ou lorsque la circulation ne peut se faire qu'à allure lente. Ils peuvent aussi être employés pour répéter un signal antérieur. Les signaux de grandes dimensions seront employés sur les routes de grande largeur à circulation rapide. Les signaux de très grandes dimensions seront employés sur les routes à circulation très rapide, notamment sur les autoroutes

## 4.4 Position

Les panneaux de signalisation sont placés de manière à pouvoir être reconnus aisément et à temps par les conducteurs auxquels ils s'adressent. Habituellement, ils sont placés du côté de

## **Chapitre 1 –Code de la route et conception de l’application**

---

la route correspondant au sens de la circulation. Toutefois, ils peuvent être placés ou être répétés au-dessus de la chaussée.

Tout signal placé du côté de la route correspondant au sens de la circulation devra être répété au-dessus ou de l’autre côté de la chaussée lorsque les conditions locales sont telles qu’il risquerait de ne pas être aperçu à temps par les conducteurs auxquels il s’adresse.

## Partie II –Conception de l’application

Avant de commencer la phase du codage l’ors du développent d’une application il est Très nécessaire de suivre une méthodologie de modélisation qui permettra de savoir D’une manière claire et précise la structure et le comportement de l’application

### 1 Le langage UML

Le langage UML est un standard est définit comme suit :

#### 1.1 Définition d’UML

C’est un langage de modélisation graphique à base de pictogrammes. Il est utilisé en développement logiciel, et en conception orientée objet. Logiciels c a dire il est utilisé pour spécifier, visualiser, modifier et construire les documents nécessaires au bon développement d'un logiciel orienté objet

#### 1.2 Les point forts d’UML

- UML est un langage formel et normalisé
  - gain de précision
  - gage de stabilité
  - encourage l'utilisation d'outils
- UML est un support de communication performant
  - Il cadre l'analyse
  - Il facilite la compréhension de représentations abstraites complexes.
  - Son caractère polyvalent et sa souplesse en font un langage universel.

#### 1.3 Les points faibles d'UML

La mise en pratique d'UML nécessite un apprentissage et passe par une Période d’adaptation

L’intégration d’UML dans un processus n’est pas triviale, et améliorer un processus est une tâche complexe et longue

# Chapitre 1 –Code de la route et conception de l'application

---

## 1.4 UML est un standard

Il est nécessaire qu'une méthode objet soit définie de manière rigoureuse et unique afin de lever les ambiguïtés. De nombreuses méthodes objet ont été définies, mais aucune n'a su s'imposer en raison du manque de standardisation. Ce qui a motivé les acteurs du monde de l'informatique a fondée en 1989 l'OMG (Object Management Group) , une organisation a but non lucratif, dont le but est de mettre au points des standards garantissant la compatibilité entre des applications programmées à l'aide de langage objet et fonctionnant sur des réseaux hétérogènes.

A partir de 1997, UML est devenue une norme de l'OMG, ce qui lui a permis de s'imposer en tant que méthode de développement objet et être reconnue et utilisée par des nombreuses entreprises

## 1.5 UML est un langage de modélisation et pas une méthode

UML permet d'exprimer d'élaborer et de modéliser au sens de la théorie de langage, de ce fait il contient les éléments de ce derniers : concepts, une syntaxe et une sémantique il met a notre disposition une multitude de concepts pour représenter des modèles et en plus suivant différentes vues. Il ne te fournit pas une démarche précise à **suivre** pour modéliser mais plutôt des outils pour bien accomplir tes démarches.

## 1.6 Les digrammes

Un digramme UML est une représentation graphique, qui s'intéresse à un aspect précis du Modèle

Chaque type de diagramme UML possède une structure et une sémantique précise

UML 2.3 propose 14 types de diagrammes (9 en UML 1.3)

## 2 Conception

Il existe plusieurs méthodes de développement logiciel construites sur UML comme la méthode : UP, RUP, TTUP, AUP, XUP, 2TUP ....

### 2.1 Le processus unifié (UP)

Le processus unifié est un processus de développement logiciel itératif, centré sur l'architecture, piloté par des cas d'utilisation et orienté vers la diminution des risques

# Chapitre 1 –Code de la route et conception de l'application

C'est un patron de processus pouvant être adaptée a une large classe de systèmes logiciels ,a différents domaines d'application, a différents types d'entreprises, a différents niveaux de compétences et a différents tailles de l'entreprise

Une démarche de la méthode générique UP consiste à :

- Elaborer les cas d'utilisation
- Diagrammes de séquences
- Diagrammes de classes

## 2.1.1 Élaborer les cas d'utilisation

La figure I.2 illustre le digramme de cas d'utilisation

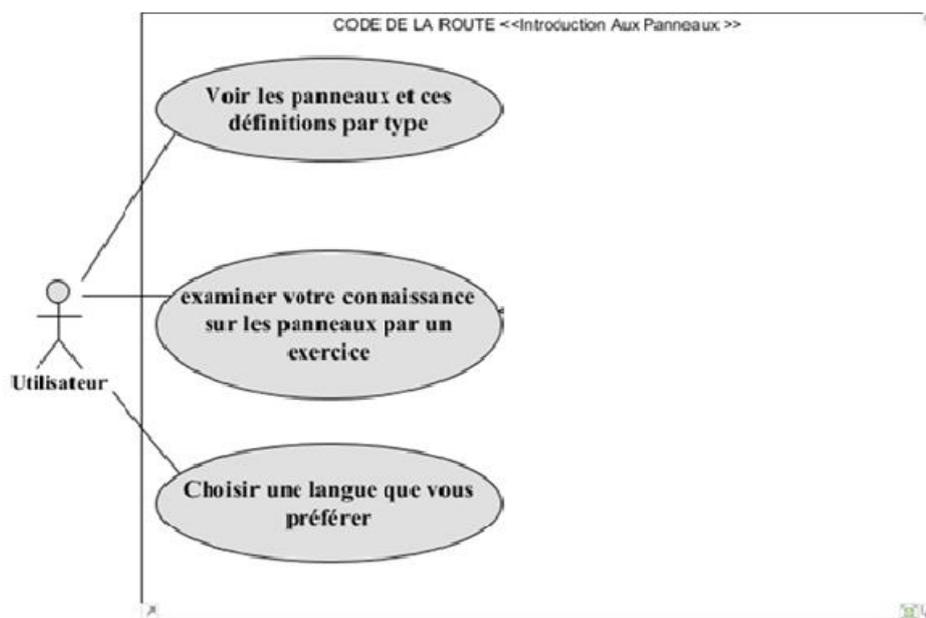


Figure I.2 : digramme de cas d'utilisation

## 2.1.2 Diagrammes de séquences

- ✓ Identification des acteurs

Un acteur représente un ensemble cohérent de rôles joués par les utilisateurs en générale un acteur représente un rôle qu'un être humain, une machine ou même un autre système joue avec le système et dans notre application l'acteur est un simple utilisateur qui cherche a l'apprentissage du code de la route

- ✓ Identification des activités

Elles représentent tous simplement tous ce que fait le système du point de vue de l'utilisateur est dans notre application on peut distinguer deux activités :

# Chapitre 1 –Code de la route et conception de l’application

- Voir les panneaux et ses définitions par types
- Examiner votre connaissance sure les panneaux par des exercices sous formes de QSM

## 2.1.3 Diagramme de classe

On peut à partir de diagrammes de séquence et de la liste d’objets tirés de dresser le diagramme de classe qui expose les relations entre ces objets.

## 2.2 Diagramme des paquetages

Les **diagrammes de paquetages** sont la représentation graphique des relations existant entre les paquetages et la figure illustre le diagramme de paquetages de notre application :

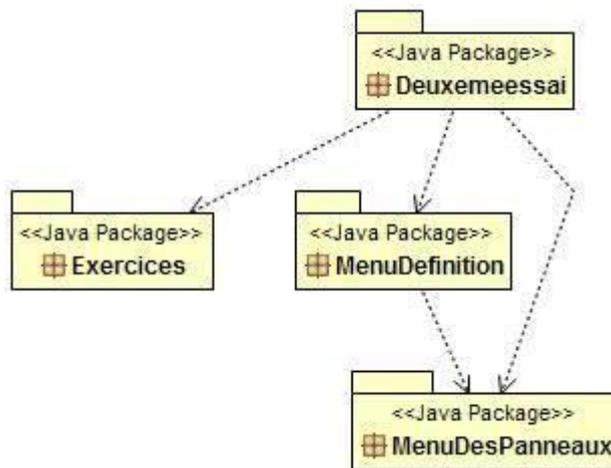


Figure I.3 : digramme de paquetages

On peut aussi définir les diagrammes de composants de chaque paquetage

La figure I.4 illustre le diagramme de composant du paquetage Exercice

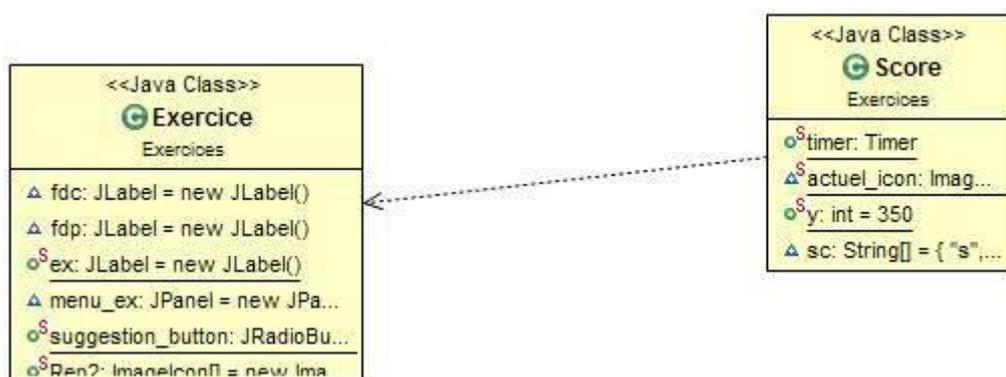


Figure I.4 : diagramme de composant du paquetage Exercice

# Chapitre 1 –Code de la route et conception de l’application

La figure I.5 illustre le diagramme de composant du paquetage Deuxemeessai

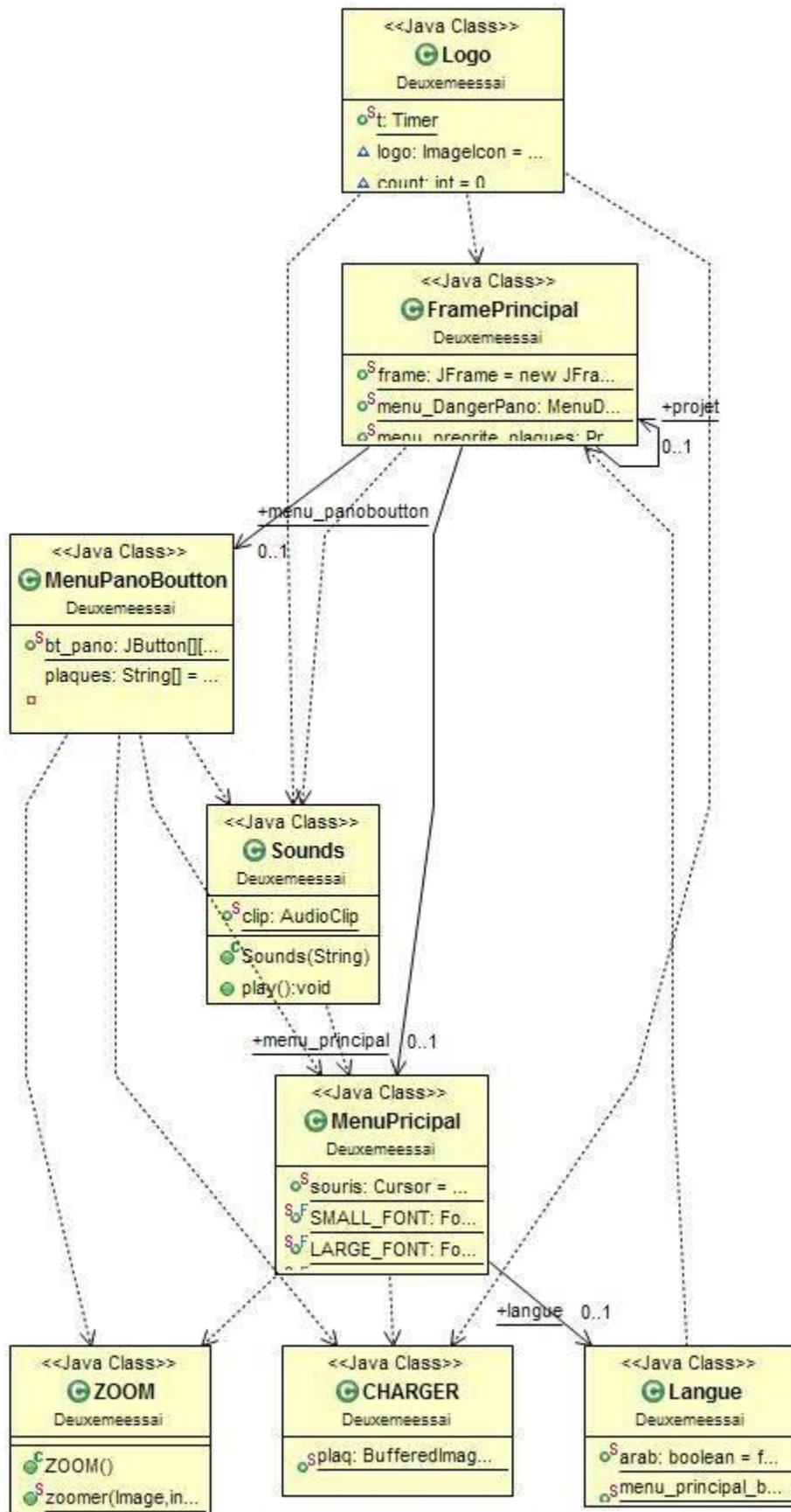


Figure I.5 : diagramme de composant du paquetage Deuxemeessai

# Chapitre 1 –Code de la route et conception de l’application

La figure I.6 illustre le diagramme de composant du paquetage MenuDesPanneaux

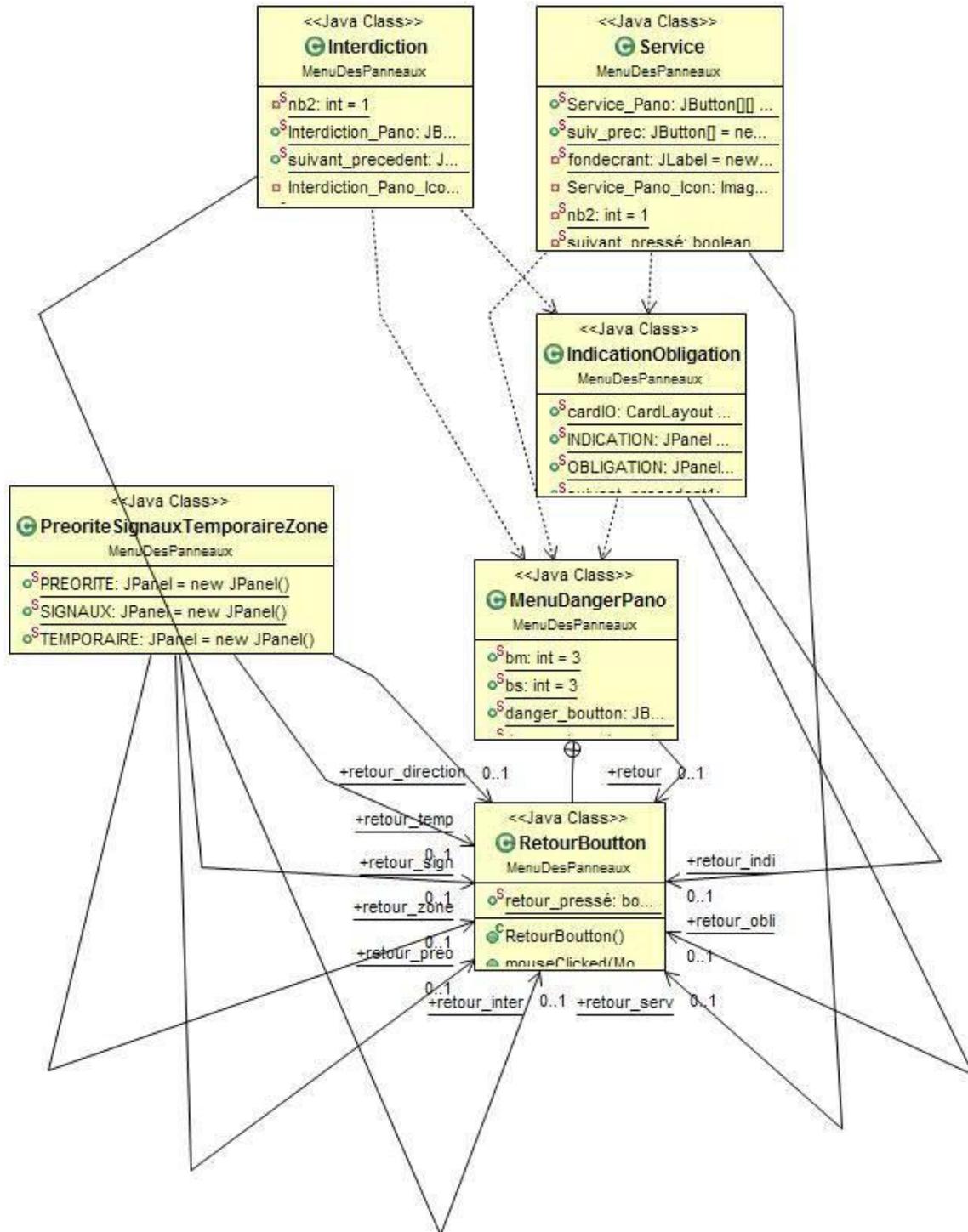


Figure I.6 : diagramme de composant du paquetage MenuDesPanneaux

La figure I.7 illustre le diagramme de composant du paquetage MenuDefinition

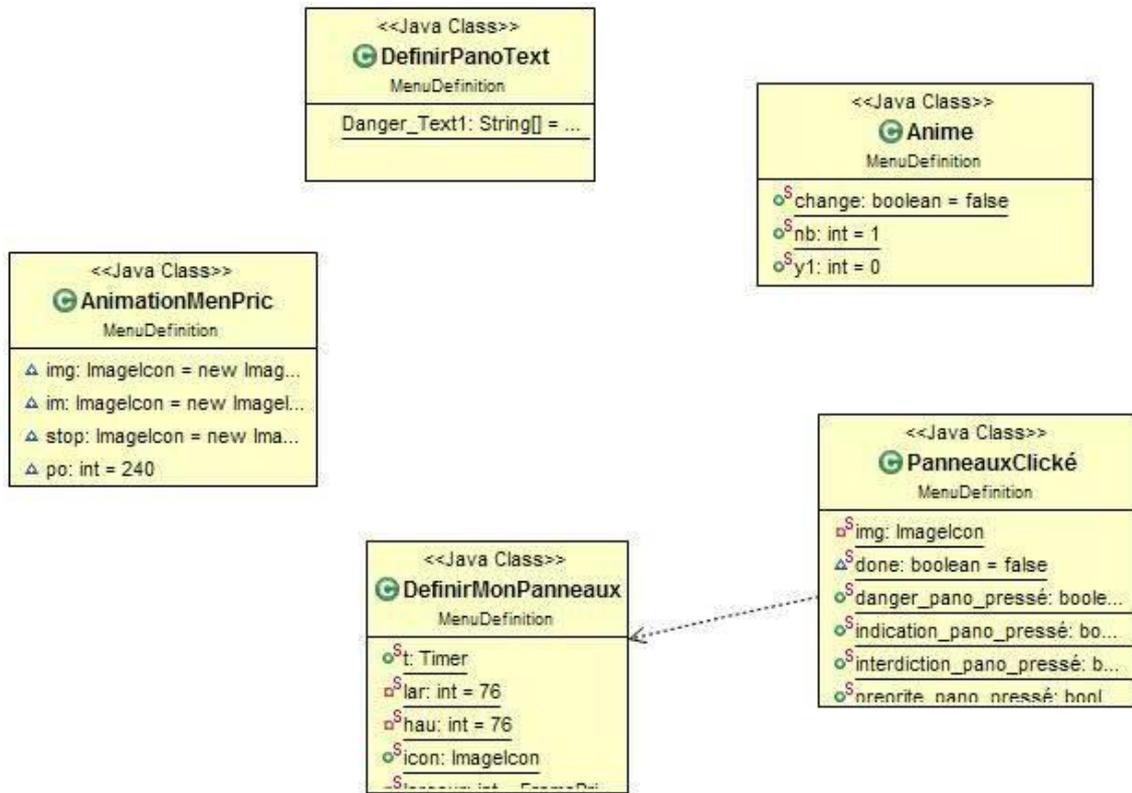


Figure I.7 : diagramme de composant du paquetage MenuDefinition

Dans ce chapitre nous allons introduire tous les outils que nous avons utilisés lors du développement de l'application

### 1 Présentation du langage JAVA

#### 1.2 Introduction

Le langage Java est un langage de programmation informatique orienté objet créé par James Gosling et Patrick Naughton, employés de Sun Microsystems, avec le soutien de Bill Joy (cofondateur de Sun Microsystems en 1982), présenté officiellement le 23 mai 1995 au *SunWorld*.

La société Sun a été ensuite rachetée en 2009 par la société Oracle qui détient et maintient désormais Java.

Le langage Java reprend en grande partie la syntaxe du langage C++, très utilisée par les informaticiens. Néanmoins, Java a été épuré des concepts les plus compliqués du C++ et à la fois les plus déroutants, tels que les pointeurs et références, ou l'héritage multiple contourné par l'implémentation des interfaces. Les concepteurs ont privilégié l'approche orientée objet de sorte qu'en Java, tout est objet à l'exception des types primitifs (nombres entiers, nombres à virgule flottante, etc.).

### 2 Particularité du langage java

La particularité et l'objectif central de Java est que les logiciels écrits dans ce langage doivent être très facilement portables sur plusieurs systèmes d'exploitation tels que UNIX, Windows, Mac OS ou GNU/Linux, avec peu ou pas de modifications. Pour cela, divers plateformes et Framework associés visent à guider, sinon garantir, cette portabilité des applications développées en Java

Cette particularité est obtenu par :

- des bibliothèques standard fournies pour pouvoir accéder à certains éléments de la machine hôte (le graphisme, le multithreading, la programmation réseau...) exactement de la même manière sur toutes les architectures.
- des compilateurs Java qui compilent le code source « à moitié » afin d'obtenir un **bytecode** (plus précisément le **bytecode Java**, un langage de type assembleur, proche de la machine virtuelle et spécifique à la plate-forme Java), faisant apparaître autant de

## Chapitre 2 – Outils utilisés

fichiers avec l'extension `.class` que de classes développées dans le code source. Toutefois, contrairement aux langages compilés traditionnels, pour lesquels le compilateur crée un fichier binaire directement exécutable par un processeur donné (c'est-à-dire un fichier binaire contenant des instructions spécifiques à un processeur).

La figure II.1 illustre le cycle d'exécution d'un fichier source écrit en langage compilé traditionnel

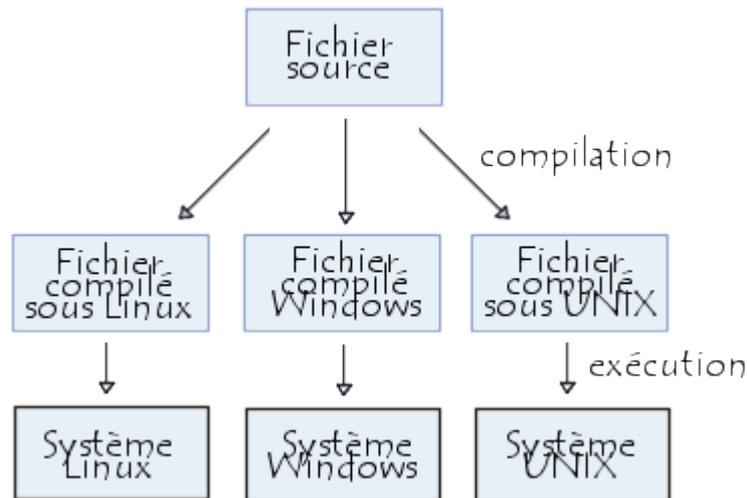


Figure II.1 cycle d'exécution d'un fichier source écrit en langages compilé traditionnels

Cette caractéristique est majeure, car c'est elle qui fait qu'un programme écrit en Java est portable, c'est-à-dire qu'il ne dépend pas d'une plate-forme donnée. En réalité le code intermédiaire n'est exécutable sur aucune plate-forme sans la présence d'une machine virtuelle, un interpréteur la machine virtuelle est d'ailleurs parfois appelée *interpréteur Java* (JVM en anglais), un programme écrit spécifiquement pour la machine cible qui interprète le bytecode Java et fait exécuter par la machine les instructions traduites en code natif.

Noter que même s'il y a explicitement une première phase de compilation, le bytecode Java est soit interprété, soit converti à la volée en code natif par un compilateur à la volée (*just in time*, JIT).

La figure II.2 illustre le cycle d'exécution d'un fichier source écrit en langage compilé traditionnel

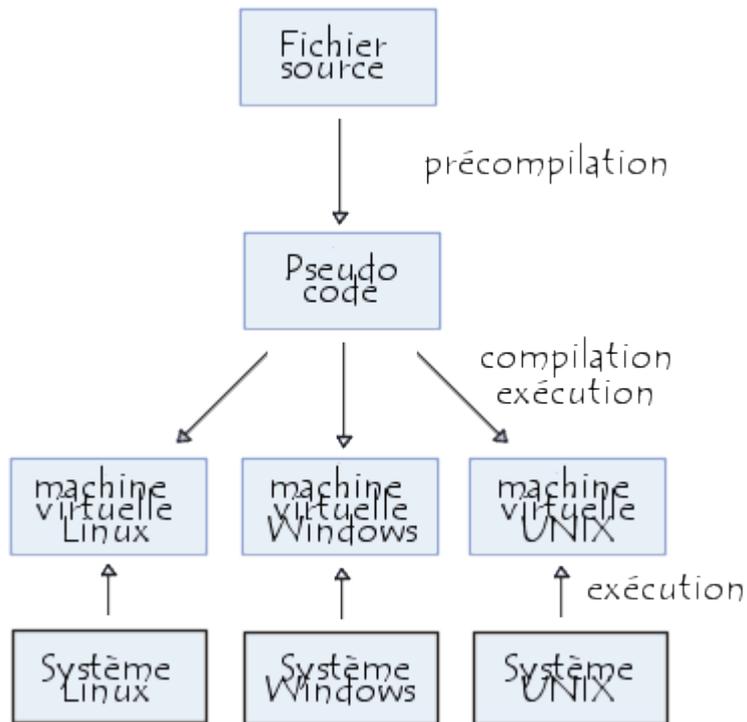


Figure II.2 : cycle d'exécution d'un fichier source écrit en langage java

Ainsi, pour peu qu'une plate-forme (windows 95, Unix, Linux, Amiga, ...) possède une machine virtuelle fonctionnant sous son système, celle-ci est capable d'exécuter n'importe quelle application Java !

De cette façon, l'ensemble des navigateurs permettant d'utiliser des applets possèdent une machine virtuelle...

### 3 Fichier source JAVA

Le fichier source d'un programme écrit en Java est un simple fichier texte dont l'extension est par convention *.java*. Ce fichier source doit être non formaté, c'est-à-dire un fichier texte dans sa plus simple expression, sans mise en forme particulière (pas de texte souligné, en italique, ou autres). En revanche le langage Java supporte les caractères Unicode, on peut donc y introduire des caractères spéciaux, en particulier des lettres accentuées.

### 4 Objectif du langage java

Lors de la création du langage Java, il avait été décidé que ce langage devait répondre à cinq objectifs:

1. simple, orienté objet et familier
2. robuste et sûr
3. indépendant de la machine employée pour l'exécution
4. très performant
5. interprété, multitâches et dynamique

### 5 Environnements de développement

La programmation peut se faire depuis une invite de commande en lançant un compilateur Java (souvent nommé javac), mais pour avoir plus de confort, il est préférable d'utiliser un environnement de développement intégré ou IDE, certains sont gratuits. On peut citer :

- BlueJ
- CodeWarrior
- Eclipse
- IntelliJ IDEA
- JBuilder
- JCreator
- jDeveloper
- NetBeans
- Xcode

### 6 Présentation de IDE eclipse

#### 6.1 Définition de IDE eclipse

Eclipse est un IDE, *Integrated Development Environment* (EDI environnement de développement intégré en français), c'est-à-dire un logiciel qui simplifie la programmation en proposant un certain nombre de raccourcis et d'aide à la programmation. Il est développé par IBM, est gratuit et disponible pour la plupart des systèmes d'exploitation.

Son objectif est de produire et fournir des outils pour la réalisation de logiciels, englobant les activités de programmation (notamment environnement de développement intégré et

frameworks) mais aussi d'AGL recouvrant modélisation, conception, testing, gestion de configuration, reporting... Son EDI, partie intégrante du projet, vise notamment à supporter tout langage de programmation à l'instar de Microsoft Visual Studio.

Bien qu'Eclipse ait d'abord été conçu uniquement pour produire des environnements de développement, les utilisateurs et contributeurs se sont rapidement mis à réutiliser ses briques logicielles pour des applications clientes classiques. Cela a conduit à une extension du périmètre initial d'Eclipse à toute production de logiciel : c'est l'apparition du framework Eclipse RCP en 2004.

Figurant parmi les grandes réussites de l'Open source, Eclipse est devenu un standard du marché des logiciels de développement, intégré par de grands éditeurs logiciels et sociétés de services. Les logiciels commerciaux *Lotus Notes 8*, *IBM Lotus Symphony* ou *WebSphereStudio Application Developer* sont notamment basés sur Eclipse.

Au fur et à mesure que vous programmez, eclipse compile automatiquement le code que vous écrivez, en soulignant en rouge ou jaune les problèmes qu'il détecte. Il souligne en rouge les parties du programme qui ne compilent pas, et en jaune les parties qui compilent mais peuvent éventuellement poser problème (on dit qu'eclipse lève un avertissement, ou *warning* en anglais). Pendant l'écriture du code, cela peut sembler un peu déroutant au début, puisque tant que la ligne de code n'est pas terminée (en gros jusqu'au point-virgule), eclipse indique une erreur dans le code.

Il est déconseillé de continuer d'écrire le programme quand il contient des erreurs, car eclipse est dans ce cas moins performant pour vous aider à écrire le programme.

### 6.2 Workspace?

Le workspace est le répertoire dans lequel sont stockés :

- les projets, c'est-à-dire les différents programmes que vous allez réaliser.
- la configuration d'éclipse.

### 6.3 Architecture d'éclipse

Le métaprojet Eclipse est constitué et organisé en une galaxie de projets logiciels. Sa spécificité tient à son architecture totalement développée autour de la notion de plugin en

conformité avec la norme OSGi : toutes les fonctionnalités de l'atelier logiciel doivent être développées en tant que *plug-in* bâti autour de l'**Eclipse Platform**.

Eclipse propose un framework de développement logiciel fournissant des briques logicielles pour développer ces outils. En fait Eclipse est à la fois considéré comme un EDI, un framework ou une plateforme, selon que l'on considère le projet, ses composants, les EDI résultant de leur assemblage:

En effet le projet Eclipse propose également des 'packages' en téléchargement<sup>1</sup>: il peut s'agir :

- d'applications 'prêtes à l'emploi' facilitant la diffusion d'Eclipse, en intégrant chacune un ensemble cohérent de plugins autour de l'Eclipse Platform pour répondre à différents besoins spécifiques.
  - Il s'agit essentiellement d'IDE spécialisés, tels que *Eclipse Classic*, *Eclipse IDE for Java EE Developers*, *Eclipse IDE for C/C++ Developers*, *Eclipse for Mobile Developers*,
  - mais également d'AGL comme *Eclipse Modeling Tools*.

### 6.4 Le noyau 'Eclipse Project'

**Eclipse Project** constitue le projet fondateur autour duquel s'agrègent les autres composants Eclipse; il comprend en 2012 les sous-projets:

- *Platform*, qui définit les composants communs 'de base' de l'ensemble du modèle de développement Eclipse;
- Le *Plug-in development environment*, ou 'PDE', ensemble de plugins Eclipse permettant de développer et tester d'autres plugins Eclipse, en conformité avec OSGi sur lequel repose la philosophie Eclipse. PDE permet ainsi de créer les plugins Eclipse comme des Bundle OSGi puis, par assemblage, des applications RCP complètes;
- Les *Java Development Tools*, ou 'JDT' : cet ensemble regroupe les plugins couvrant les fonctionnalités usuelles et spécifiques aux environnements de développement tel que l'exécution de code avec débogage.

### 6.5 Eclipse Platform

L'IDE Eclipse Platform est principalement écrit en Java (à l'aide de la bibliothèque graphique SWT, d'IBM); ce langage, grâce à des bibliothèques spécifiques, est également utilisé pour écrire les extensions.

La base de l'environnement de développement intégré que constitue l'EDI *Eclipse Platform* est composée de :

*Platform Runtime* démarrant la plateforme et gérant les plug-ins

SWT, la bibliothèque graphique de base de l'EDI

JFace, une bibliothèque graphique de plus haut niveau basée sur SWT

*Eclipse Workbench*, la dernière couche graphique permettant d'organiser et manipuler des composants, tels que des vues, des éditeurs et des perspectives.

Tous ces composants de base peuvent être réutilisés pour développer des clients lourds indépendants d'Eclipse grâce au 'package' Eclipse RCP (*Rich Client Platform*).

### 6.6 Concurrents

Parmi ses concurrents, on trouve :

- NetBeans gratuit et développé par Sun.
- IDEA de JetBrains qui est payant.

## 7 Les bibliothèques utilisées

### 7.1 Abstract window toolkit

**Abstract Window Toolkit** (AWT) est une bibliothèque graphique pour Java, faisant partie de JFC (Java Foundation Classes). Cette bibliothèque a été introduite dès les premières versions de Java ; depuis Java 2, la bibliothèque de gestion de fenêtre officielle est Swing. Toutefois, AWT sert encore de fondement à Swing, dans la mesure où de nombreuses classes Swing héritent de classes AWT.

AWT emploie les composants natifs de la plate-forme, alors que Swing utilise des composants en pur Java.

### 7.2 Swing

**Swing** est une bibliothèque graphique pour le langage de programmation Java, faisant partie du package Java Foundation Classes (JFC), inclus dans J2SE. Swing constitue l'une des principales évolutions apportées par Java 2 par rapport aux versions antérieures.

Swing offre la possibilité de créer des interfaces graphiques identiques quel que soit le système d'exploitation sous-jacent, au prix de performances moindres qu'en utilisant Abstract Window Toolkit (AWT). Il utilise le principe Modèle-Vue-Contrôleur (MVC, les composants Swing jouent en fait le rôle du contrôleur au sens du MVC) et dispose de plusieurs choix d'apparence (de vue) pour chacun des composants standards.

Ce chapitre consiste à détailler tous les interfaces de notre application.

### 1 Premier interface de l'application

#### 1.1 Le logo d'entrée à l'application

Les deux figure III.1 et III.2 sont utilisées comme étant une entrée à l'application , la figure III.1 vient au début puis la figure III.2



Figure III.1 : logo 1

et tous ça se fait souvent un temp bien précis et une simple animation d'éclairage(effet de transparence du couleur noir).

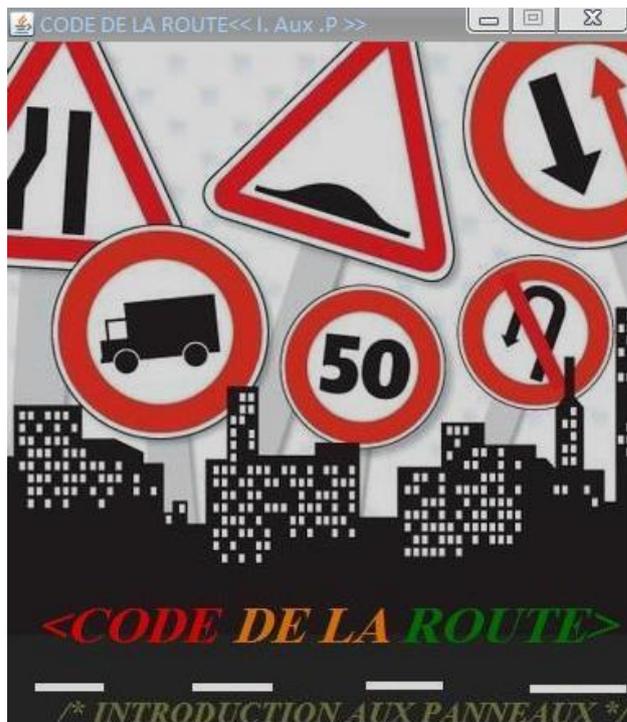


Figure III.2 : logo 2

## 2 Menu principale

La figure suivante représente le menu principal de l'application

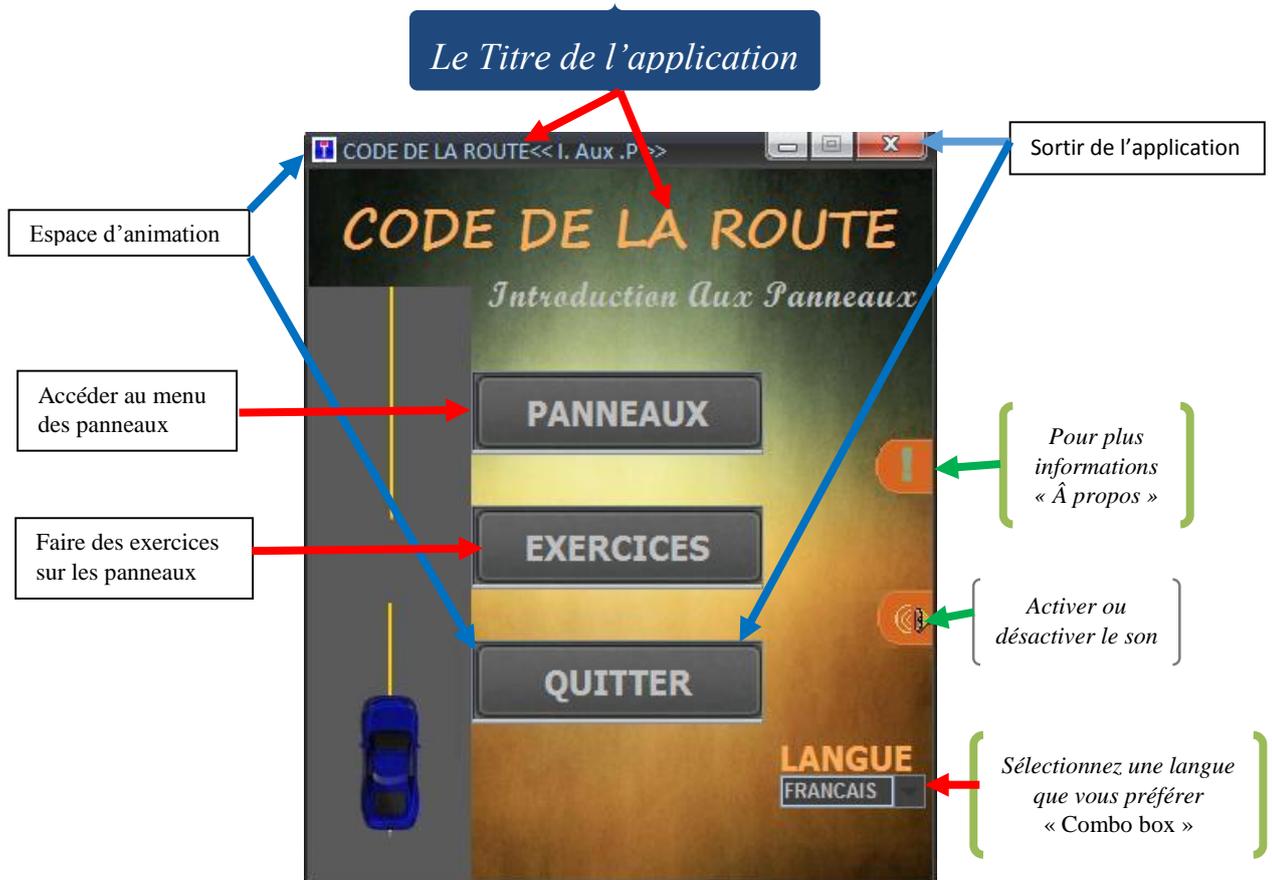


Figure III.3 : menu principale

C'est le premier menu qui apparaisse et qui permet à l'utilisateur de découvrir les différentes fonctionnalités de l'application. Comme vous voyez l'interface est un simple panel qui contient des simples composants (boutons, combo box ..).

Le but d'ajouter les espaces d'animations dans ce menu, est d'enrichir la décoration pour faire attirer l'attention de l'utilisateur.

Tous les éléments qui contiennent un texte (les boutons, combo box, le titre) sont reliés simultanément avec le composant langue (combo box), qui contient deux termes français et arabe. Le combo box est initialisé par le terme français, c'est-à-dire la langue est par défaut le français, l'utilisateur peut changer facilement la langue.

La figure suivante indique l'état du menu principale , après la sélection du langue arabe :



Figure III.4 menu principale arabe

Ces composants du menu principal sont des boutons optionelles tellque :



Pour plus informations.



Le son est desactivé.



Le son est activé.

Si vous appuyer sur



une boîte de message s'affiche, elle contient tous les coordonnées nécessaires pour le contact.

La figure suivante qui montre l'évènement du boutons a propos :

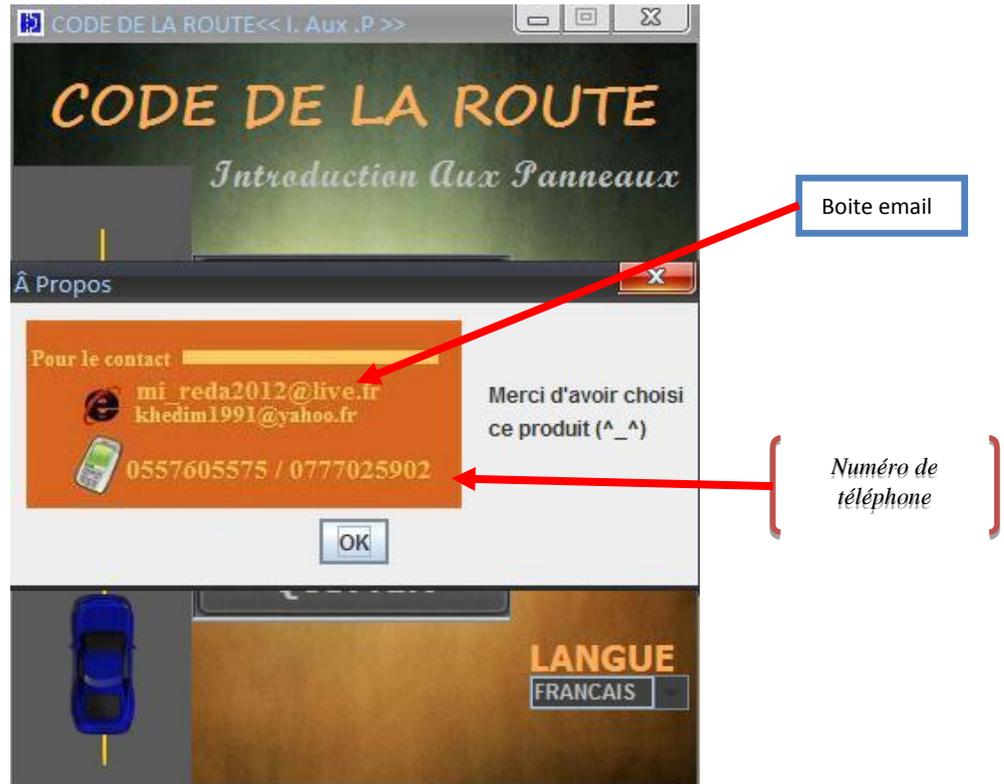


Figure III.5 message de coordonnées

Une boîte de confirmation dialogue apparaisse dès que vous appuyez sur le bouton « QUITTER » la figure suivante montre bien l'évènement :

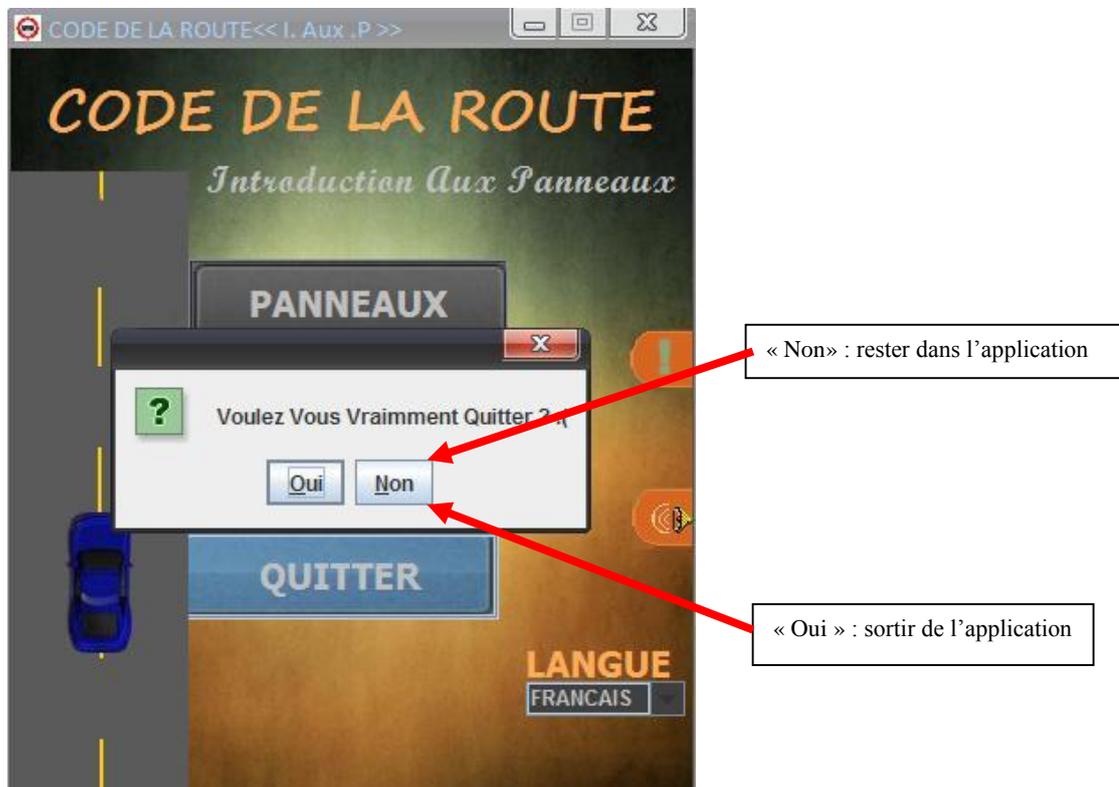


Figure III.6 message de confirmation

### 3 Menu de types des Panneaux

Ce menu est composé de 10 boutons.

#### 3.1 Avant animation

On obtient ce menu dès qu'on clique sur le bouton «PANNEAUX»

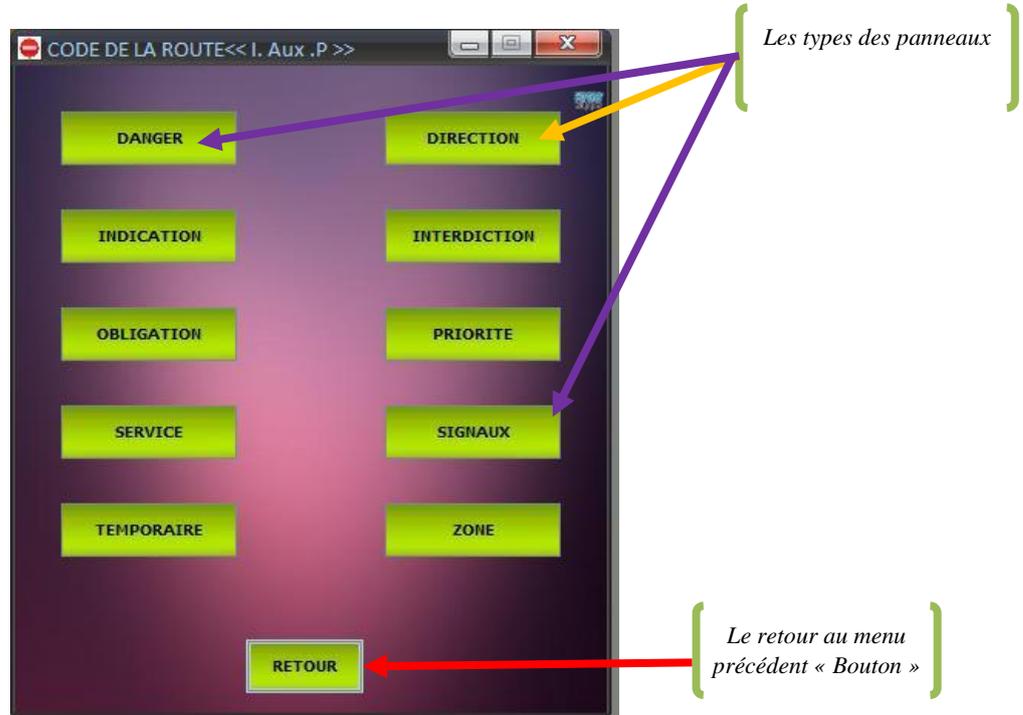


Figure III.7 menu des panneaux

Si la langue sélectionnée est arabe alors on aura la figure suivante :



Figure III.8 menu des panneaux arabe

3.2 Avec animation

La figure III.9 est une amélioration de la figure 8 ,on parle d'une sous frame et une animation qui apparaissent dès que le curseur se plante sur l'un des 10 boutons (les 10 types des panneaux) :



Figure III.9 menu de type des panneaux animé

La sous frame contient la définition de chaque type des panneaux choisi par l'utilisateur.

L'animation montre tous les panneaux d'un type choisi par l'utilisateur. Si la langue sélectionnée est arabe alors on aura la figure suivante :



Figure III.10 menu de type des panneaux arabe animé

#### 4 Menu des panneaux

La Figure III.11 montre les panneaux d'un type de panneaux choisi(cliqué) dans le menu de type des panneaux.

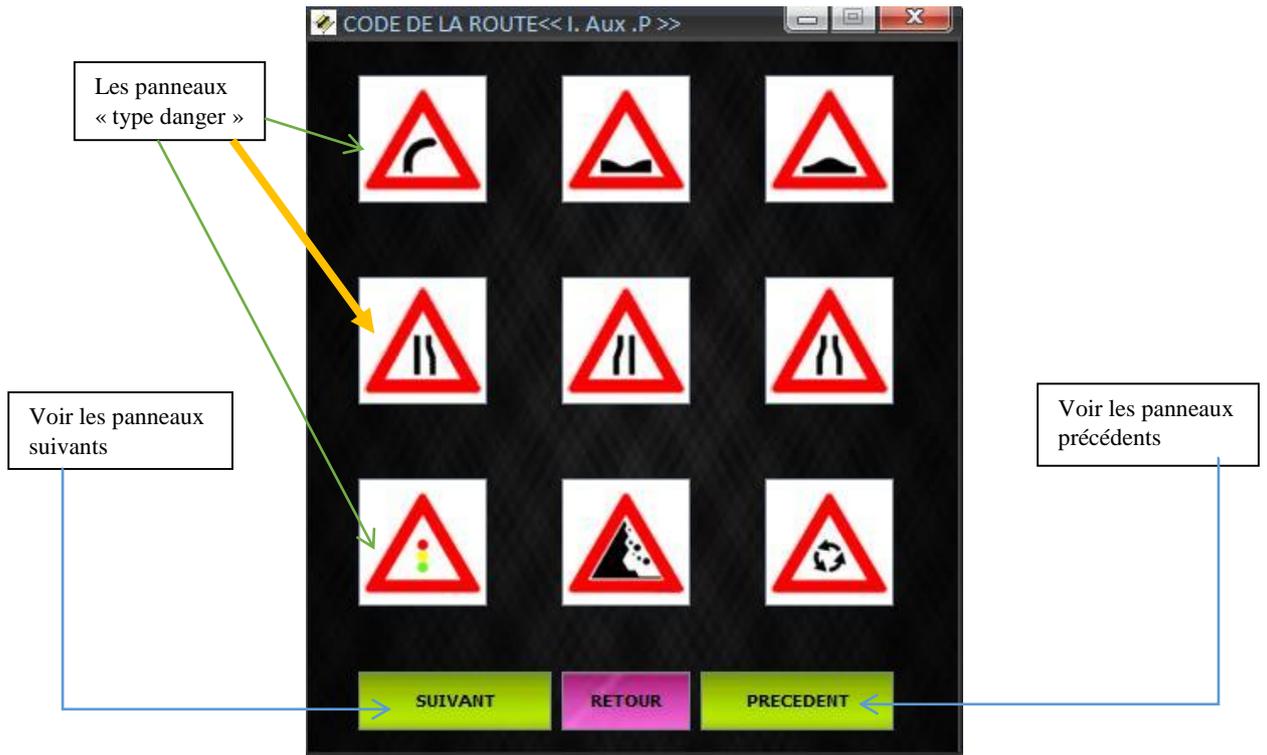


Figure III.11 menu des panneaux (cas danger)

Les panneaux dans cette figure sont des boutons. Les deux boutons(suivant et précédent) sert à voir tous les panneaux, si la langue sélectionnée est arabe alors on aura la figure suivante :



Figure III.12 menu des panneaux arabe (cas danger)

Voici un autre exemple d'un menu des panneaux , cas d'interdiction :



Figure III.13 menu des panneaux (cas interdiction)

La Figure III.14 montre la version arabe de la figure précédente, si la langue sélectionnée est arabe :



Figure III.14 menu des panneaux arabe (cas interdiction)

## 5 Menu de définition des panneaux

Ce menu sert à définir tous les panneaux un par un, c'est à l'utilisateur de choisir le panneau et le type, la figure suivante montre le fonctionnement de ce menu :

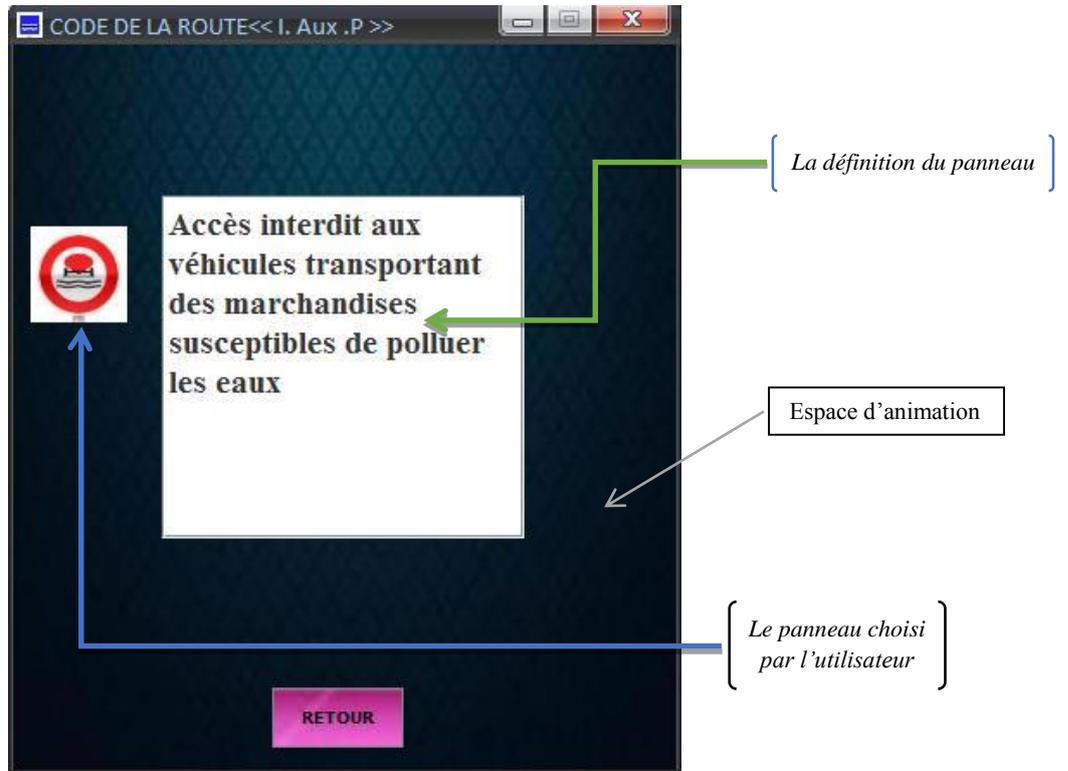


Figure III.15 menu de définition des panneaux

Une animation de panneau sélectionné, dans le but d'attirer l'attention de l'utilisateur.



Figure III.16 menu de définition des panneaux arabe

## 6 Menu exercice

On peut distinguer trois états dans ce menu, on parle de l'interface (partie question + partie score..), correction (la correction des réponses), une notation (l'affichage du score) :

### 6.1 L'interface

L'utilisateur obtient ce menu dès qu'il clique sur le bouton «EXERCICES» du menu principal, il est composé de trois composants (fond écran, question et le score).

La figure ci-dessous montre bien les composants de cette interface :

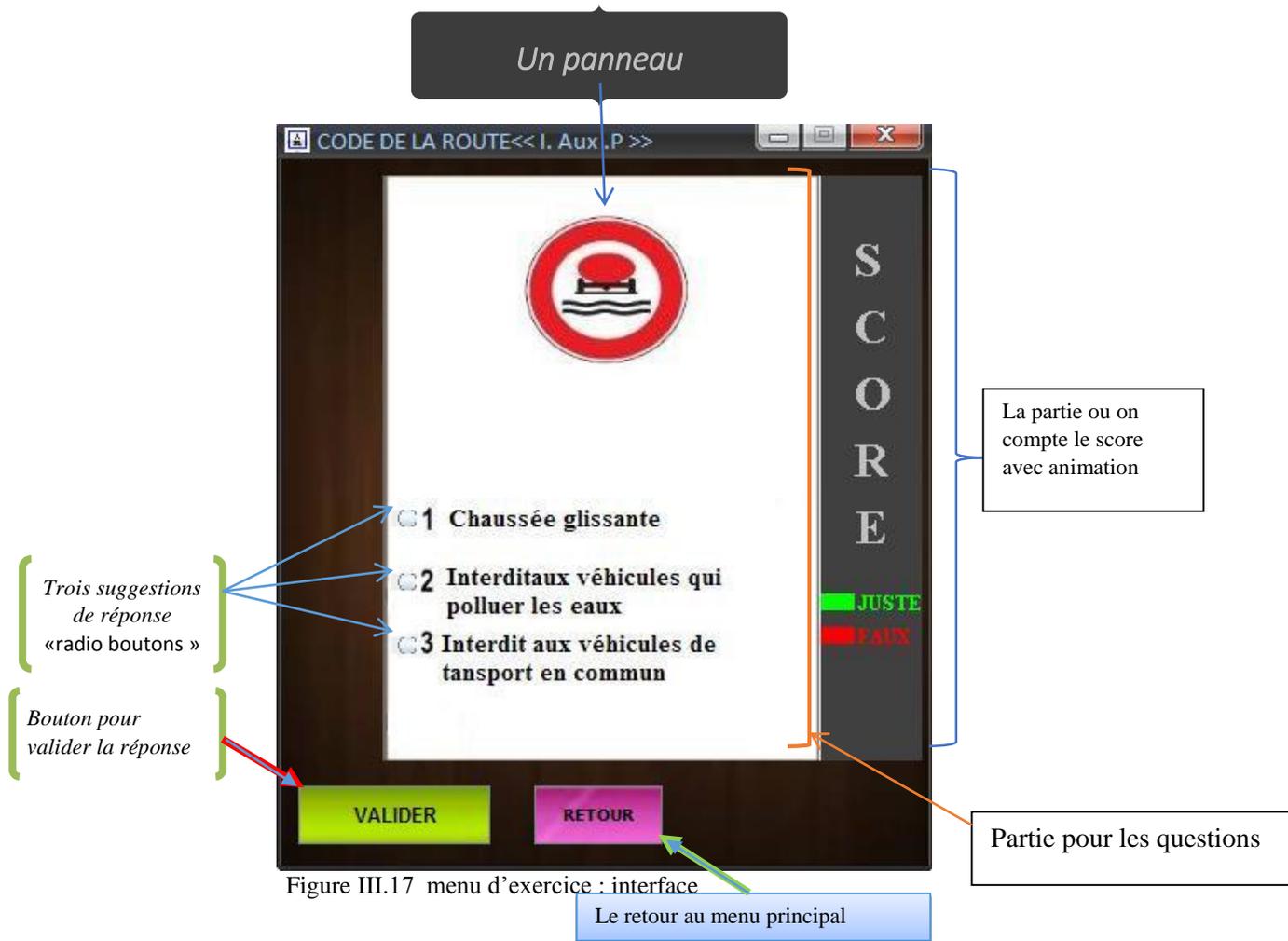


Figure III.17 menu d'exercice : interface

La partie fond écran contient deux boutons (VALIDER et RETOUR), le bouton valider sert à valider la réponse après la sélectionnée, et le bouton retour pour revenir au menu précédent.

La partie question contient trois radios boutons avec suggestions.

Le but de cet exercice est de faire détecter la bonne définition du panneau au-dessus, et sélectionner leur radio bouton qui convient, puis valider la réponse on appuyant sur le bouton « VALIDER ».

Les questions sont montrées de façon aléatoire et au hasard dans chaque nouveaux exercice ou après chaque correction ou notation.

Voilà la version arabe de ce menu, bien sûr après la sélection de la langue arabe dans le menu principal :

Dans la version arabe de ce menu la partie score ne changent pas de langue.



Figure III.18 menu d'exercice arabe : interface

Les suggestion et les deux boutons « VALIDER » et « RETOUR » changent de langue.

Les radios boutons changent leur position du gauche a droite lorsque la langue arabe est sélectionner.

6.2 Correction

Cette étape montre si la réponse est juste ou fausse, la figure suivante montre ça :

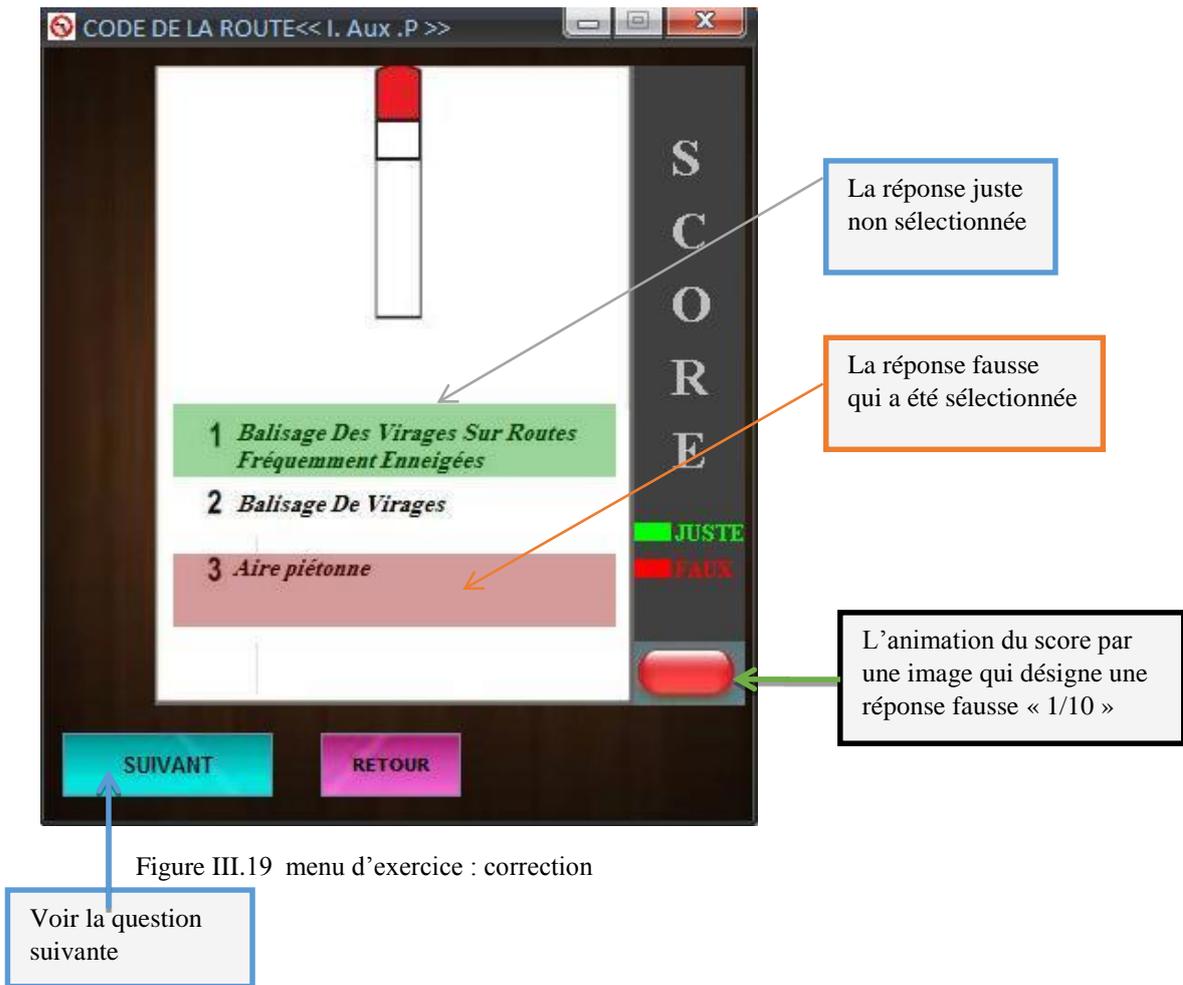


Figure III.19 menu d'exercice : correction

Après avoir sélectionnez une réponse dans les radios boutons puis cliquer sur le bouton valider, la correction est affiché automatiquement, cette figure représente le cas si la réponse sélectionnée est fausse.

La réponse juste est représentée par un rectangle vert.

La réponse fausse est représentée par un rectangle rouge.

Dans la partie score, une image est dessinée indique la nature de la réponse, si elle est rouge la réponse sélectionnée est fausse, si elle est verte la réponse est donc juste.

La partie score peut contenir dix images, c'est-à-dire on à dix question au maximum dans chaque tour.

Pour passer à la question suivante il faut cliquer sur le bouton suivant.

Voici la correction dans le cas où la langue arabe est sélectionnée :

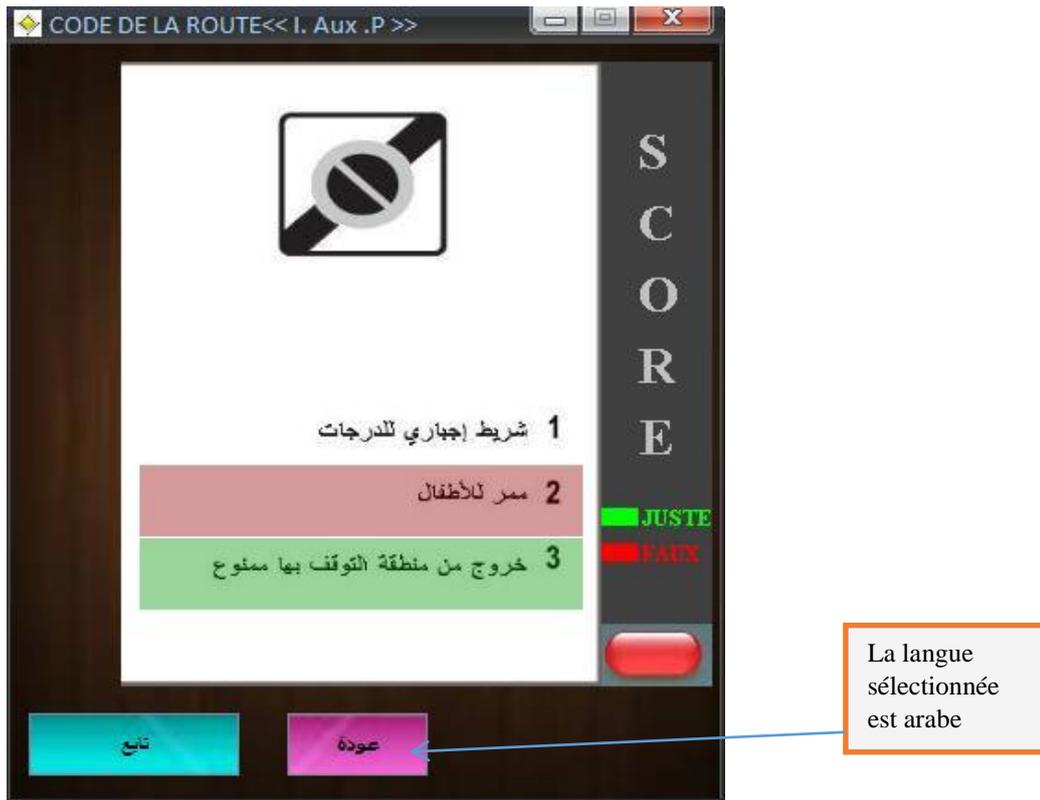


Figure III.20 menu d'exercice arabe : correction

### 6.3 Notation

La notation est reliée directement avec la partie score, voir la figure ci-dessous :



Figure III.21 menu d'exercice : notation

le nombre de rectangles verts représente le nombre de réponses juste, dans le cas du Figure III.21, le nombre de rectangles verts est 10 donc il y a 10 bonnes réponses.

la version arabe est représentée dans la Figure III.22 :



Figure III.22 menu d'exercice arabe : notation

## Conclusion générale

---

Ce projet a fait l'objet d'une expérience intéressante qui nous a permis d'améliorer nos connaissances

Et nos compétences dans le domaine de la conception et la programmation orientée objet

Nous nous sommes familiarisés avec les concepts objets.

Nous avons réussi à programmer en java en cernant plusieurs difficultés

Nous avons réussi à installer, configurer, ajouter des plug-ins et manipuler adéquatement l'EDI éclipse

Pour les perceptives on peut ajouter des images de intersections et de gestes de l'agent de circulation

Et leur signification

On peut aussi ajouter des questions et leur réponse par exemple quels sont les cas où la priorité est à droite,

L'ajout se fait dans les deux partie la partie cours et la partie exercices.

# Références bibliographiques

---

- [1]: <http://www.vulgarisation-informatique.com/interpretation-java.php>
- [2]: <http://www.commentcamarche.net/contents/550-java-caracteristiques-du-langage>
- [3]: [http://fr.wikipedia.org/wiki/Convention\\_de\\_Vienne\\_sur\\_la\\_signalisation\\_routiere](http://fr.wikipedia.org/wiki/Convention_de_Vienne_sur_la_signalisation_routiere)
- [4]: [http://fr.wikipedia.org/wiki/Panneau\\_de\\_signalisation\\_routiere](http://fr.wikipedia.org/wiki/Panneau_de_signalisation_routiere)
- [5]: <http://fr.scribd.com/doc/6770144/Methodologie-UML-Cours-Du-Cycle-B-Du-Cnam>
- [6]: <http://jwuydin.fr/toutsuruml/questceUml.php>
- [7]: <http://www.developpez.net/forums/d792863/general-developpement/alm/modelisation/uml/langage-uml-non-methode-conception-uml/>
- [8]: <http://liris.cnrs.fr/csolnon/publications/usdp.pdf>
- [9]: [http://fr.wikipedia.org/wiki/Unified\\_Process](http://fr.wikipedia.org/wiki/Unified_Process)
- [10]: [www.passetoncode.fr/panneaux-de-signalisation](http://www.passetoncode.fr/panneaux-de-signalisation)

## *Résumé*

---

Notre projet consiste à développer une application java (gestion de code de la route), dans le but d'aider à apprendre le code de la route.

L'implémentation est faite à l'aide du langage java sous eclipse IDE (version indigo),

Ce projet a fait l'objet d'une expérience intéressante qui nous a permis d'améliorer nos connaissances et nos compétences dans le domaine de la conception et la programmation orientée objet.