

**UNIVERSITE “POLITEHNICA” DE BUCAREST  
FACULTE D’INGINERIE EN LANGUES ETRANGERES  
ORDINATEURS ET TECHNOLOGIE DE L’INFORMATION**

**PROJET  
DEVELOPPEMENT DES APPLICATIONS POUR LES  
PLATEFORMES MOBILES**

**Solve IT!**

**ETUDIANTS**

**Bărbătei Ioana-Alexandra**

**Dumitrescu Vlad**

**Grădinescu Cristiana**

**Petrescu Andra**

**Tătaru Ioana-Miruna**

**Groupe 1240F**

**BUCAREST  
2015**

# **SOMMAIRE**

## **1. INTRODUCTION**

## **2. IMPLEMENTATION**

### **2.1. SPECIFICATIONS**

### **2.2 DESIGN**

### **2.3 ARCHITECTURE**

### **2.4 FONCTIONALITES**

### **3.5 STRUCTURE DU CODE**

### **2.6 BIBLIOTHEQUES**

## **3. CONCLUSIONS ET PERSPECTIVES**

## **4. REFERENCES**

# 1 INTRODUCTION

L'objectif principal de l'application Android Solve IT! est l'évaluation de la mémoire de courte durée.

Sur le marché actuel, il y existe beaucoup de jeux de puzzle, mais cette application a un chose de plus: le niveau de difficulté est choisi en fonction de la capacité de la mémoire.

La mémoire à court terme désigne en psychologie le type de mémoire qui permet de retenir et de réutiliser une quantité limitée d'informations pendant un temps relativement court, environ une demi-minute.

Postman & Phillips (1975) et Glanzer & Cunitz (1966) ont présenté des listes de mots de différentes longueur puis ont demandé à leurs sujets de rappeler les mots dont ils se souvenaient dans l'ordre qu'ils souhaitaient (rappel libre).

Quand le rappel a lieu immédiatement les premiers et les derniers éléments de la liste ont le plus de chance d'être rappelés tandis que peu de sujets se souviennent du milieu de la liste. Quand le rappel a lieu après 15 ou 30 secondes, seuls les éléments du début de la liste ont une forte probabilité de rappel.

Les auteurs interprètent ces expériences comme une preuve de l'existence d'une mémoire à court terme (short-term store) responsable de l'effet de récence. Alors que le début de la liste serait rappelé de la mémoire à long terme (effet de primauté) et donc encore disponible après 30 secondes, la fin de la liste serait stockée en mémoire à court terme et disparaîtrait rapidement après la présentation.

On a intégré cette recherche dans le développement de l'application.

Pour résoudre le problème énoncé auparavant on a proposé trois niveaux de difficulté: easy (15 sec), medium (10 sec), hard (5 sec).

Les technologies et les outils utilisées pour le développement de l'application sont: Eclipse avec SDK Tools et le plugin Android Development Tools (ADT) et pour tester on a utilisé un smartphone HTC One Mini 2 qui a la version Android 4.4 (KitKat).

## **2 IMPLEMENTATION**

### **2.1 SPECIFICATIONS**

L'application Solve IT! représente un jeu, un puzzle qui doit être résolu par l'utilisateur. Une fois l'application démarrée, l'utilisateur va voir pour 3 secondes l'intro du jeu, avec le nom et le logo, puis le menu principal va apparaître, une fenêtre avec trois boutons : un bouton qui lance un nouveau jeu (New Game), un bouton d'aide (How to play) et un bouton pour voir le About. Si l'utilisateur va choisir d'appuyer sur le bouton New Game, une nouvelle fenêtre va apparaître, où l'utilisateur va pouvoir introduire son nom, choisir la difficulté, puis appuyer sur Start ou Exit

( pour sortir de l'application ).

Si l'utilisateur va choisir Start, une nouvelle fenêtre va apparaître, avec une image qui va rester sur l'écran pour quelque secondes, en fonction de la difficulté choisie. Après ça, une nouvelle fenêtre sera ouverte, avec l'image originale divisée en plusieurs parties dispersées sur la surface de l'écran. L'utilisateur devra reconstruire l'image initial et s'il va réussir de le faire, à la fin du jeu, un message de félicitations sera affiché sur l'écran.

### **2.2 DESIGN**

Pour utiliser l'application il y a plusieurs gestes à faire.

La première interaction est le touch qui lance l'application, quand l'utilisateur appuie sur le logo. Ensuite, il y a des interactions comme: touch et swipe/drag. Presque tous les gestes que l'utilisateur doit faire sont de gestes de touch, l'exception est le geste fait pour recomposer l'image qui est un geste de type swipe/drag.

Dans la figure 2.2.1 on retrouve les wireframes et les interactions qui compose le design de l'application, principalement le flow:

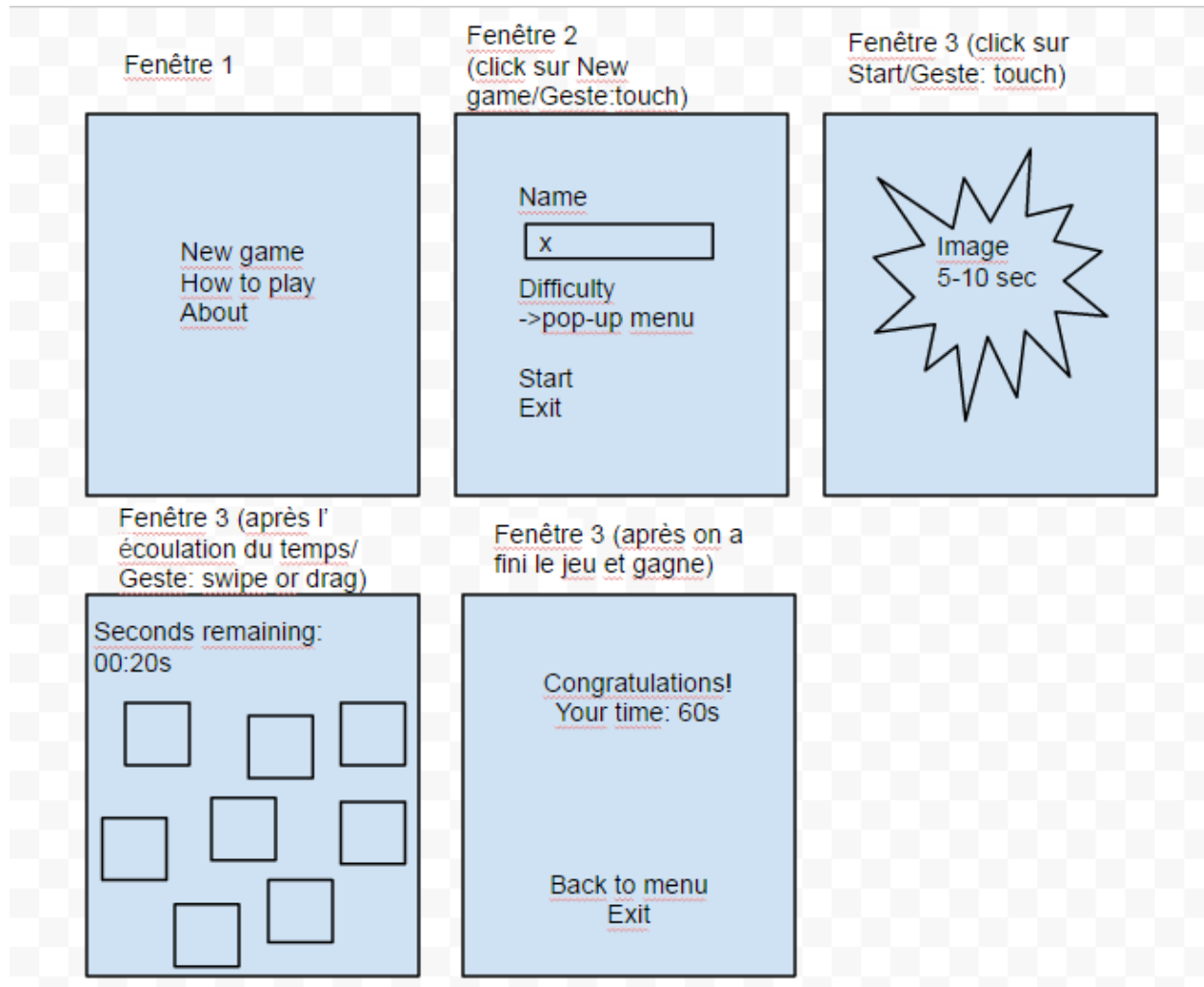


Figure 2.2.1

## 2.3 ARCHITECTURE

Les données de l'utilisateur et les records (les résultats) sont stockées dans un array. Les images utilisées se retrouvent dans un dossier. L'image est sélectionnée dans un mode aléatoire. Le flow de l'application est le suivant: l'utilisateur lance le jeu en appuyant sur le logo. Ensuite il va apparaître le menu principal avec les options: New Game, How to Play, About. Si l'utilisateur est nouveau, il peut voir comment il faut jouer le jeu (How to Play). Ici il y a l'aide, une explication du jeu, comment on le joue. Si l'utilisateur veut voir des détails sur les développeurs et la version de l'application il va appuyer sur About. Pour se jouer l'utilisateur appuie sur New Game. Il va écrire son nom et choisir la difficulté. Ensuite il va voir pour quelques secondes en fonction de la difficulté choisie une image qu'il doit retenir. Il doit recomposer l'image. L'utilisateur peut toujours sortir de l'application. S'il a réussi le jeu il va voir un message de félicitation (Congratulations!).

## 2.4 FONCTIONALITES

L'activité principale contient un menu (Create, Update, View) avec: New game, How to play, Show records, About (objet de type ListView avec ces items). Si l'utilisateur clique un de ces boutons, une nouvelle fenêtre va apparaître à l'écran. On va définir le layout pour chaque interface graphique dans un fichier .xml. Pour les boutons du menu on va avoir les méthodes onClick et setOnClickListener pour naviguer entre les fenêtres. Le bouton New game déclenche l'apparition d'une nouvelle fenêtre avec la suivante liste (ListView): Name et input (Text field), Difficulty et un menu de type pop-up (ListView) pour choisir la difficulté du jeu et boutons d'opération: Start, Exit. Ces boutons, aux méthodes onClick et setOnClickListener font les suivants: lance le jeu, retourne au menu principal et sort de l'application. Pour le bouton Start, il apparait un nouvelle interface graphique qui contient un photo (on a besoin de stocker des photos, la photo est choisit aléatoirement) qui restera sur l'écran pendant 5-10 secondes. Ensuite, elle sera divisé et la nouvelle interface contiendra: time, buton de exit et les parties de la photo. En utilisant swipe et drag, l'utilisateur reconstituera l'image initiale et il apparaîtra sur l'écran: Congratulations! avec le score et le temps.

## 2.5 STRUCTURE DU CODE

Il existe un projet applé DAPM dans l'intérieur duquel on trouve les dossiers: src où se trouvent les activités, res où on a les sous-dossiers drawable dans lesquels il y a les images (les ressources UI) qu'on les utilise, layout dans lequel il y a les fichiers .xml qui définissent l'interface avec l'utilisateur.



Figure 2.5.1 Fenêtre 1

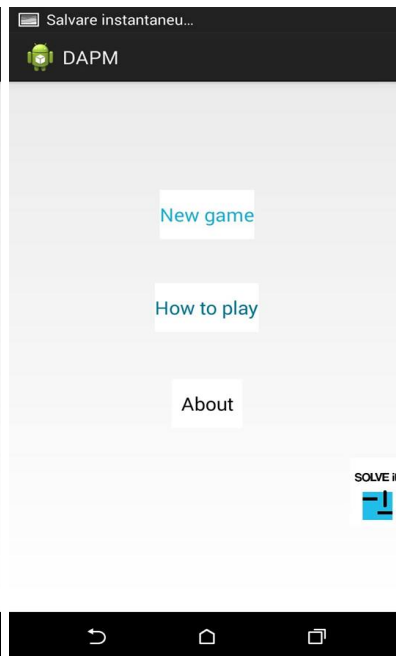


Figure 2.5.2 Menu

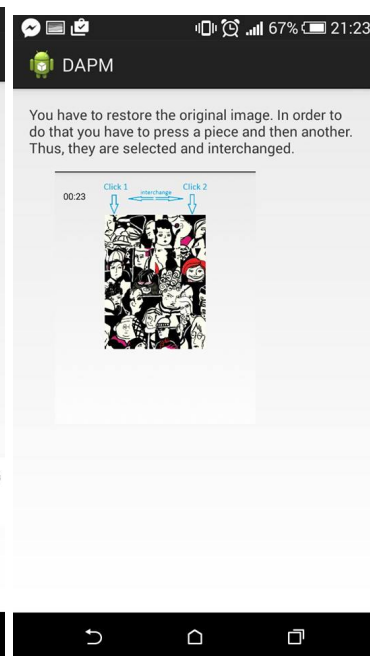


Figure 2.5.3 How to play

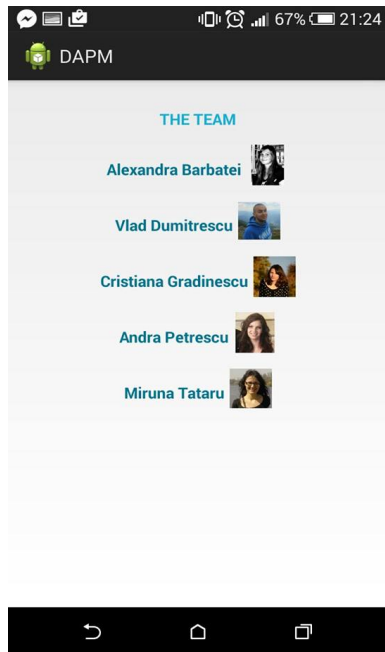


Figure 2.5.4 About

Le jeu:

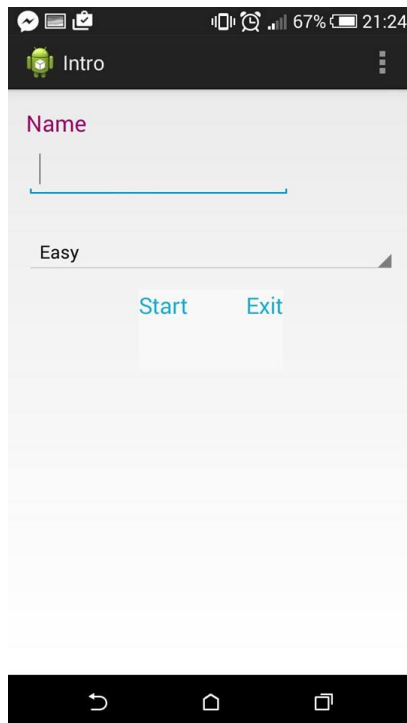


Figure 2.5.5 New game

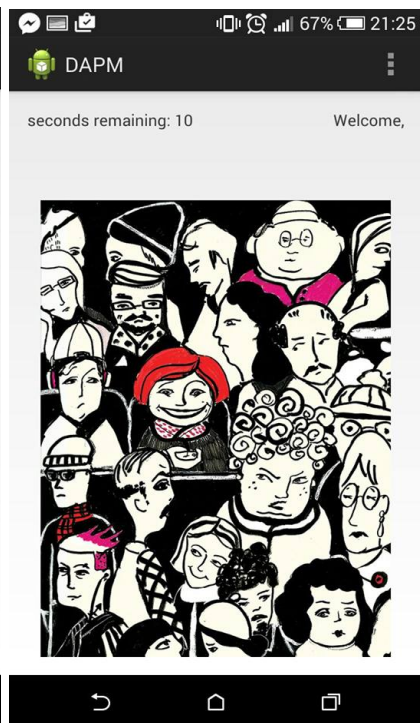


Figure 2.5.6 Image



Figure 2.5.7 Reconstitution

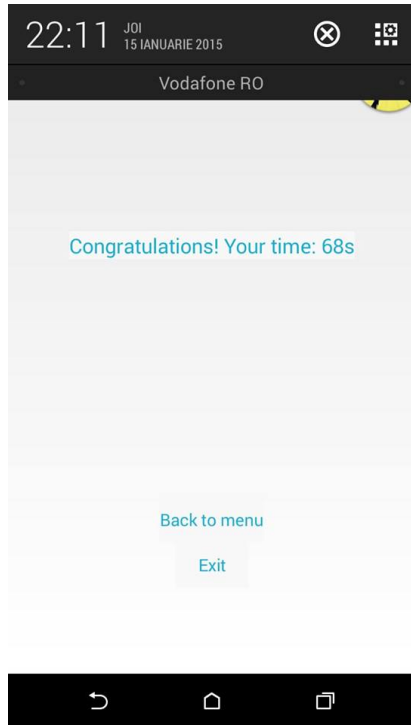


Figure 2.5.8 Fin

### 3 CONCLUSIONS ET PERSPECTIVES

En conclusion, le résultat du développement de l'application Android Solve IT! est une version préliminaire. Le projet est ouvert au futurs développements. Par exemple il peut inclure la connexion avec le compte de Facebook, Twitter ou Google+ pour partager le résultat avec les amis. Ensuite, l'application peut mémorer les records, les points obtenus par l'utilisateur. On peut aussi introduire la possibilité de jouer en groups de 2, 3 ou plusieurs.

### 4 REFERENCES

[http://en.wikipedia.org/wiki/Short-term\\_memory](http://en.wikipedia.org/wiki/Short-term_memory)  
<http://developer.android.com>  
<http://www.photoshop.com/products/photoshop>



