Mystery Doors-Team A6

Department of Information Science & Engineering,
BVBCET Hubli

May 22, 2012

ACKNOWLEDGMENT

We have been bestowed the privilege of expressing my gratitude to everyone who helped us in completing the mini project work.

Firstly, we would like to thank our guide Mrs.Ranjini S, Department of Information Science & Engineering, BVBCET Hubli, whose guidance was great use to us during the project work.

We would like to express our sincere thanks to Dr. Meena S.M Prof & Head of the Department, Information Science & Engineering, BVBCET Hubli, for providing us an opportunity to carry out this project work successfully.

We find our acknowledgement incomplete without thanking Dr. Ashok Shettar, Principal BVBCET, Hubli for the inspiration and co-operation.

We would also like to thank our project coordinator Ms. P.G. Sunitha Hiremath, Associate Prof and Mr. Amitkumar Gundad, Lecturer Department of Information Science & Engineering, BVBCET Hubli, for coordinating and planning the mini-project schedule and also the faculty members of ISE dept.

Finally we thank one and all who have directly and indirectly assisted us in the project work.

Mr.Harish Patil
Mr.Pavankumar B L
Mr.Pradeep Channalli
Ms.Priyanka Patil

ABSTRACT

This application is basically a puzzle-based game. The game is made more interesting by adding a catchy storyline and scenarios. In this game there will be four characters playing Alex, Maya, Dhaka and David. At the beginning there will be conversations between first three characters which introduce the theme of story. Later fourth character is introduced as players instructor. There will be 5 levels of game i.e. 5 doors to be opened in this context. In each level there will be a set of questions or puzzles which needs to be solved by player in order to open the door and move to next level. This level and type of puzzles will be increased level increments. There will be a surprise or bonus for a player after every door he opens. The game will be having very interesting climax to end..

Contents

List of Figures

List of Tables

Chapter 1

INTRODUCTION

The report mainly includes the drawbacks of the previously existing system, Motivation, Problem definition of the project and Scope of the project etc. The Software Requirements Specifications (SRS) document defines the requirements for the system and the methods to be used to ensure that each requirement is satisfied for the project. This document encapsulates all the characteristics and features expected of the system including functionality, user interfaces, performance and attributes. Any constraints to the implementation of the system are also discussed here.

1.1 Existing Sytem and Its Drawbacks

In the past video game industries were booming exponentially. But with the success rate of personal computers and mobile phones the online gaming websites and applications have become a hit. Earlier games were just for entertainment, to pass the time or to just relax the mind. But with the world being so competitive the clich of entertainment in games has turned to learning. People tend to learn through games. People tend to learn in every moment. We believe in learning throughout the life. There are so many gaming applications which provide entertainment. But there are very few applications which provide knowledge through entertainment. This application is one among them.

1.2 Motivation

The motivation for the Mystery Doors Windows project has two considerations. First one is the importance of logical reasoning and practical

analysis. Second one is to learn building the applications for Windows operating system which will be the future extensive use in many mobile phones. Today, games have become a vital part in life. There is a huge competition between many mobile phone Operating Systems. To overcome this competition efficient and useful applications are necessary. Keeping this in mind we have developed this application for our windows OS users.

1.3 Problem Statement

To develop an gaming application which will push the users intelligent level to its best. Different levels of puzzles are taken with interesting constraints imposed on the user. And will challenge the user to answer those questions to reach the final goal.

1.4 Scope of project

- This mobile application is categorized under Games, where the main objective is entertainment.
- Entertainment is the main criteria of this application.
- The main features of this application are- Entertainment, increasing the reasoning skills and it is suitable for all age groups.
- The puzzle game instances that various players solve must be matched with the player ability.
- The difficulty level must be maintained according to the players ability. We call this combined challenge as the puzzle difficulty balance.

Chapter 2

SYSTEM ANALYSIS

It deals with the analyzing of the entire system i.e. the different entities and the interaction between those entities and also the different functionalities of the system. It consists of performance analysis, It also consists of the specification of different types of requirements like software requirements, communication requirements, database requirements etc.

2.1 System Model

The software system has been designed into two levels. At the first level the focus is on deciding which modules are needed for the system, the specification of these modules and how the modules should be interconnected. This is called as system design or top level design. In the second level, the internal design of the modules or how the specifications of the module can be satisfied is decided. This design level is often called detailed design or logic design.

The fig.2.1 shows the system model of the application. When the user clicks on the application icon, this game is launched in the windows phone. When the game is launched the first page or the menu page of the game is displayed. The menu page consists of two buttons, PLAY and EXIT. At the click of the play button, the introduction of the game is started. Here, the characters of the game is introduced, and tells the user the storyline of the game. After the introduction of the game different levels of the game is started. When the user loses all the attempts or if the time-out occurs, the user is displayed with a game over. Where the game control is navigated to the Menu page. Throughout the game the images for the game are accessed

from the database, and the audio player agent is linked to the game to play the audio at the background.

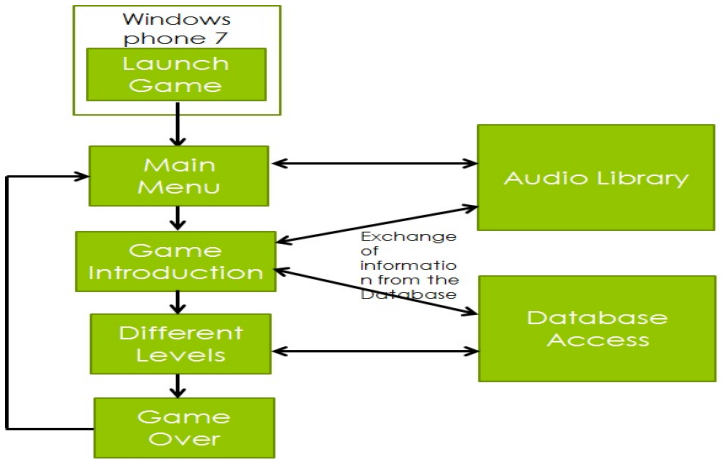

Figure 2.1: System Model

2.2 Functional Requirements

The functional requirements for our project are:

- At the start of the application, the main menu page shall be displayed.The main menu shall consist of 2 buttons, PLAY and EXIT.
- At the click of PLAY button, the user shall be given an option to enable or disable the sound of the application.
- If the user enables sound then background audio shall be played using background audio player agent.
- After enabling/disabling of the sound the application shall be navigated to the next page.
- The next page of the application shall introduce the story to the user.
- In this application .gif shall be used to enable the animation and the motion of the images.
- During gameplay different set and type of questions shall be displayed on screen and answers shall be validated before proceeding to next step.

- Different set and type of questions and all the solutions shall be stored in an isolated file storage medium i.e. mobiles flash memory.
- At a different level of game suitable type of question must be fetched from file and displayed on screen.

2.3 External Interface Requirements

There are many additional hardware and software which are required for the development of any of the software application; they also constitute a major part in the application. This section provides the overview of the external interfaces used for the development of the Mystery Doors.

2.3.1 Hardware Interfaces

- Windows mobile phone
- 128Mb RAM
- Minimum storage space on hard disk is 15Mb.

2.3.2 Software Interfaces

- Visual Studio Express for Windows Phone.
- Windows phone emulator.

2.4 User Requirements

- **Home Screen**: The first screen to be displayed which allows user to enter the user to start the game.
- **GUI**: When the user chooses some other option, then the information pertaining to that choice will be displayed on to the screen.
- **Notifications**: When the user loses, then the necessary message like game over will be displayed.

2.5 Performance Requirements

Performance requirements coincide with the Windows platform limitations.

- **Memory**- Our app requires maximum of 15 Mb memory.
- **Response time**- This apps response time is 0.5s.
- **Speed**- Depends on the mobiles main memory.
- **Portability**- Meets out all the requirements of the customer.
- **Performance**- The performance is better if the user downloads the app properly.

Chapter 3

SYSTEM DESIGN

This document specifies the detailed design of the Mystery Doors project. It deals with system overview of the overall project, proposed architecture designs and alternate architecture design and justification for the proposed one. High level block diagrams which illustrates the database and application program interaction. High level use case report which includes all users and interaction between them. Low level design of each module consists individual use case reports, and related sub-module sequence diagrams, activity diagrams, pseudo code, Analysis and Asymptotic Notation for algorithms.

3.1 System Overview

GUI contains the welcome screen or main menu screen which must consist of two buttons. PLAY and EXIT. If user clicks on EXIT button the game must terminate. When user clicks on PLAY button, he/she must be given an option to enable or disable sound. If user enables sound then background audio need to be played using background audio player agent. When user clicks on PLAY button the game must navigate to next page showing the introduction to game using motion of characters. Characters are moved using motion of images(.gif to enable animation). During gameplay different set and type of questions need to displayed on screen and answers must be validated before proceeding to next step. Different set and type of questions and all the solutions must be stored in an isolated file storage medium i.e. mobiles flash memory. At a different levels of game suitable type of question must be fetched from file and displayed on screen.

3.2 Design Aspects

The goal of this design is to ensure that the architecture is preserved, and the relationship between the component and modules is clear. This design views the modules as classes. Object oriented approaches are believed to be more natural and provide richer structures for thinking and abstraction. As we are using C sharp programming language which is truly object oriented languages.

3.2.1 Object Oriented Design

During object-oriented design (OOD), a developer applies implementation constraints to the conceptual model produced in object-oriented analysis. Such constraints could include not only constraints imposed by the chosen architecture but also any non-functional technological or environmental constraints, such as transaction throughput, response time, run-time platform, development environment, or those inherent in the programming language. Concepts in the analysis model are mapped onto implementation classes and interfaces resulting in a model of the solution domain, i.e., a detailed description of how the system is to be built.

Activity Diagram

The fig.3.1 shows the activity diagram .It describes different activities performed by different components .

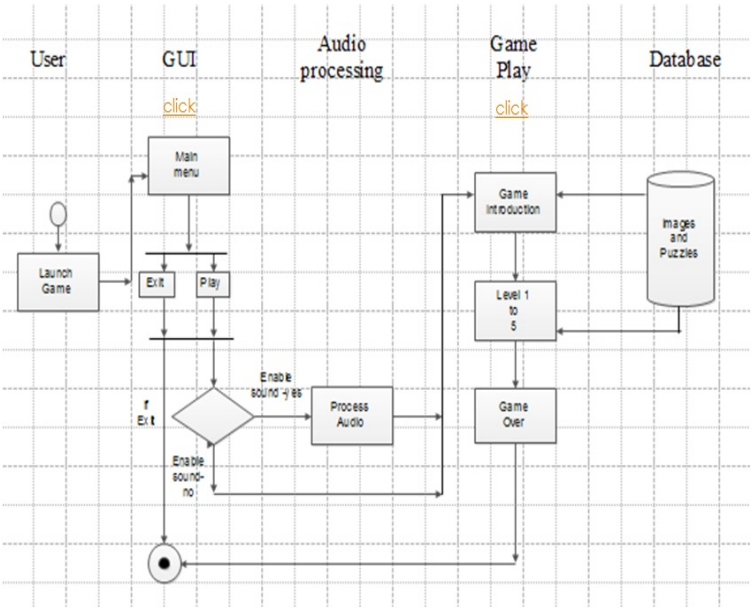

Figure 3.1: Activity-diagram

Class Diagram

In fig.3.2 shows the Class diagram. Class diagram is a type of static structure diagram that describes the structure of a system by showing the systems classes, their attributes and the relationship between the classes.

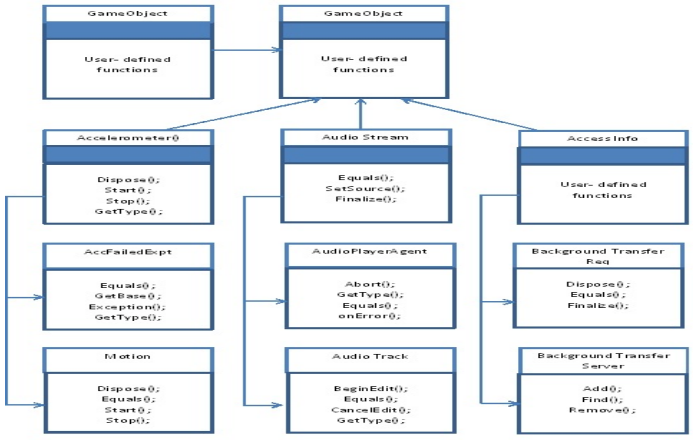

Figure 3.2: Class Diagram

Sequence Diagram

The fig.3.3 shows how processes operate with one another and in what order with the help of sequence diagram

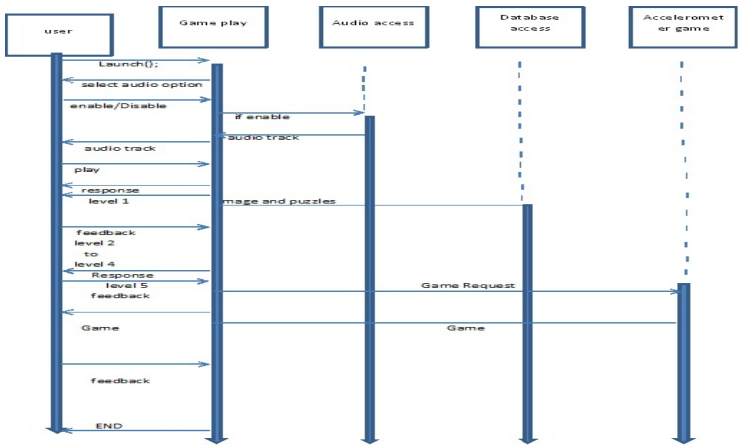

Figure 3.3: Sequence Diagram

3.3 Prototype Level Design

Since we have followed evolutionary prototyping method in developing our app, the system development happens in increments.Each prototype level design describes the incremental process followed in that phase of development.

3.3.1 Prototype : 1

When the player (Alex) gets to the first door, he encounters Dhaka. This level is used to make the player get acquainted to the game. He should answer 3 different types of questions to open the 1st Mystery Door. This section contains the 1st phase of the app development. Features implemented during this phase were:

- Gameplay till level 2.
- Questions were based on General Knowledge, Visual question and tricky riddle
- Editing of images for the characters in the story line

Algorithm for Gameplay

This is the algorithm for Gameplay
ALGORITHM:GAME PLAY
//INPUT the buttons clicked (PLAY, EXIT)
//OUTPUT proper navigation of xaml pages corresponding to those buttons
Begin
If(button is assigned to the right xaml page)
Introductory page is displayed;
else
error;
End

Algorithm for introductory page

This is the algorithm for Introductory page
ALGORITHM: intro page()
//INPUT the START button
//OUTPUT the motion of the images begin to give an introduction to the

game
Begin
If(button-clicked)
Start the motion of the image and start the introduction;
else
no motion of the images;
end

Algorithm for game levels

This is the algorithm for Game levels ALGORITHM: level one two()
//INPUT the correct answers and the CONTINUE button
//OUTPUT go to the next level
Begin
If(CONTINUE is clicked)
Go to the first level of the game;
Display the number of attempts;
Display the questions;
If(answers are right)
Notify the user;
Go to the next question or level;
Else
Notify the user about the wrong answer;
Decrement an attempt;
Else
Restart the game from the first level;
end

3.3.2 Prototype : 2

When the player reaches the 3rd door, the level of difficulty increases. The player is provided a time constraint and the use of password. The player is provided with a different set of questions, through which he can unveil the password, the password revealed will be used to unlock the 3rd door. This section contains the 2nd phase of the app development. Features implemented during this phase were:
Creation of level 3, 4 and 5

- Introducing the time-constraint in 3rd, 4th and 5th level.
- Introducing the concept of password.

Algorithms for creation of level 3, 4 and 5

Algorithm for Game Levels
ALGORITHM: rest levels ()
//INPUT the correct answers and the CONTINUE button
//OUTPUT continued to the next level and at the end a message to display
Begin
If(CONTINUE is clicked)
Go to the third level of the game;
Display the number of attempts;
Display the questions;
If(answers are right)
Notify the user;
Go to the next question or level;
Else
Notify the user about the wrong answer;
Decrement an attempt;
Else
Restart the game from the 3rd level;
end

Algorithm for time-constraint

ALGORITHM: time ()
//INPUT state where the user has completed 2 levels
//OUTPUT timer starts

Begin
if (level==3 or 4 or 5)
start timer;
if(timer==120 or 100 or 75 seconds)
terminate level and goto level 3 ;
End

Algorithm for password generation

This algorithm is for password generation
ALGORITHM: password ()
//INPUT three correct answers
//OUTPUT 2 letters of the password are revealed after answering one question

Begin
For (ques=3 ques¡4 ques++)
Reveal 2 letters;
end

3.3.3 Prototype : 3

This section contains the 3rd phase of the app development. Features implemented during this phase were:

- Introducing the concept of life in 3rd, 4th and 5th level.
- Concept of game over.

Algorithm for bonus life option

This algorithm is for bonus life option
ALGORITHM: life ()
//INPUT when the user completes a level
//OUTPUT Increase in life i.e. displays a heart symbol at the top of the screen.
Begin
If (correct answer and 3 or 4 level is crossed)
Increment In life;
Else If (user loses 3 attempts)
Decrement a life;
Else
end

Algorithm for game over

This algorithm is for game over
ALGORITHM: game over ()
//INPUT incorrect answers
//OUTPUT game over page
Begin
If (time exceeds the constraint or the lives are lost)
Display game over;
Display the button to RESTART and EXIT ;
Else

Continue the game;
Navigate the control to the last page;
end

Chapter 4

PROTOTYPE TESTING

Testing is the activity conducted to detect the errors or exceptions in the system. It is a quality management process carried on so that the product can be checked with the expectations of the system. The system is tested in two phases with unit and system testing unit testing is carried on and when the each module is coded. After the successful completion of each modules the system is integrated and validated through a complete testing to creep the errors. The software is tested with the necessary test cases to verify the accuracy of the product developed.

Test cases are the checking conditions for knowing whether the program or the part of the program is working properly. The test cases or defined before we start implementing the code. The test cases are at module level and system level as well. The test cases for our program are as follows:

4.1 Flash Screen(Valid)

Table 4.1 gives the description of valid test case for Flash screen. when the application is loaded ,it opens with the welcome screen displaying welcome to Mystery doors and it displays it.

Component	Description
Test Case ID	TC6_1
Unit to test	Flash screen.
Test Data	Welcome screen.
Steps to be executed	1. Display the Welcome screen Welcome to Mystery doors for .
Expected result	Displays the flash screen
Actual result	Displays the flash screen
Pass/Fail	Pass
Comment	———

Table 4.1: Valid test case for Flash Screen

4.2 Flash Screen(InValid)

Table 4.2 gives the description of invalid test case for Flash Screen.The GUI fails due to abnormal jumping of xaml pages or it may due to the thread created for 2 sec may not invoked so flash screen is not displayed

Component	Description
Test Case ID	TC6_1
Unit to test	Flash Screen.
Test Data	Welcome screen.
Steps to be executed	1. Display the Welcome screen Welcome to mystry doors for 2 sec.
Expected result	Displays the flash screen
Actual result	Force close
Pass/Fail	Fail
Comment	1.Error in application due to abnormal jumping of xaml pages. 2.The thread created for 2 sec is not invoked so flash screen is not displayed.

Table 4.2: Invalid test case for Flash Screen

4.3 Ok Button(Valid)

Table 4.3 gives the description of valid test case for ok button.Initially ok button is with page is opened with text areas for entering answer and to confirm. Enter the correct answer and then click on to ok button. If there is a correct match a toast message with right answer is displayed.

Component	Description
Test Case ID	TC6_1
Unit to test	ok button.
Test Data	Text area (enter correct answer.) Button(ok)
Steps to be executed	1. Enter the correct answer. 2.Click on to ok button.
Expected result	right answer
Actual result	right answer.
Pass/Fail	Pass
Comment	.....

Table 4.3: Valid test case for ok button

4.4 Ok Button(InValid)

Table 4.4 gives the description of valid test case for ok button.Initially ok button is opened with text areas for entering answer and to confirm. Enter the wrong answer for more than 3 times and then click on to ok button. If there is a incorrect match a toast message with wrong answer is displayed.

Component	Description
Test Case ID	TC6_1
Unit to test	ok button.
Test Data	Text area (enter wrong answer more than 3 times.), Button(ok)
Steps to be executed	1. Enter the wrong answer more than 3 times . 2.Click on to ok button.
Expected result	correct answer.
Actual result	wrong answer.
Pass/Fail	Fail
Comment	1.If there is no match found between entered answer and correct answer User has failed to give correct answer and cannot go to the next level.

Table 4.4: Invalid test case for ok button

4.5 Continue Button(Valid)

Table 4.5 gives the description of valid test case for continue button.If the answer is correct then he is allowed move on to the next level.

Component	Description
Test Case ID	TC6_1
Unit to test	after level page.
Test Data	correct answer
Steps to be executed	1. Enter the correct answer .
Expected result	Opening of next xaml page.
Actual result	Opening of next xaml page.
Pass/Fail	Pass
Comment	.....

Table 4.5: Valid test case for continue button

4.6 Continue Button(InValid)

Table 4.6 gives the description of invalid test case for continue button.If the

entered answer is correct, the user is not allowed to move to next level.

Component	Description
Test Case ID	TC6_1
Unit to test	.after level page
Test Data	correct answer
Steps to be executed	1. Enter the correct answer .
Expected result	start of next level.
Actual result	jumps to next level after a delay.
Pass/Fail	Fail

Table 4.6: Invalid test case for continue button

Chapter 5

IMPLEMENTATION

This chapter gives a brief description about implementation details of the system by describing each module with its code skeleton. The language used for implementation is C# and Platform used is Windows phone 7.

5.1 Prototype : 1

In prototype 1 implementation of introduction, level 1 and level 2 was been carried.

5.1.1 Introduction

Code skeleton:

```
   namespace MD
{

   public partial class Page2 :  PhoneApplicationPage

public Page2()

InitializeComponent()
;
   void leve1()
```

```
   public void doStuff()


   System.Threading.Thread startupThread =
new System.Threading.Thread(new System.Threading.ThreadStart(pDelay2));
startupThread.Start();

public void doStuff2()


   System.Threading.Thread startupThread =
new System.Threading.Thread(new System.Threading.ThreadStart(pDelay3));
startupThread.Start();

public void doStuff3()

button2.Visibility = Visibility;

BitmapImage im2 = new BitmapImage(new Uri("/MD;component/Images/sds.png",
UriKind.Relative));
BitmapImage im1 = new BitmapImage(new Uri("/MD;component/Images/z.png",
UriKind.Relative));
BitmapImage im3 = new BitmapImage(new Uri("/MD;component/q.png",
UriKind.Relative));
BitmapImage im4 = new BitmapImage(new Uri("/MD;component/w.png",
UriKind.Relative));
BitmapImage im5 = new BitmapImage(new Uri("/MD;component/Xtvv.jpg",
UriKind.Relative));
BitmapImage im6 = new BitmapImage(new Uri("/MD;component/dhaka2.png",
UriKind.Relative));
BitmapImage im7 = new BitmapImage(new Uri("/MD;component/ww.png",
UriKind.Relative));
BitmapImage im8 = new BitmapImage(new Uri("/MD;component/v1.png",
UriKind.Relative));
BitmapImage im9 = new BitmapImage(new Uri("/MD;component/v2.png",
UriKind.Relative));

   private void button1_Click_1(object sender, RoutedEventArgs e)

   System.Threading.Thread startupThread =
```

```
new System.Threading.Thread(new System.Threading.ThreadStart(pDelay));
startupThread.Start();
// DoStuff();

private void button1_Click(objectsender, RoutedEventArgse)

   void pDelay3()

int k = 20, j = 1, m = 0;
Boolean flag2 = true;
for (int i = 0; i < 50; i++)
{
System.Threading.Thread.Sleep(150);

   this.Dispatcher.BeginInvoke(() =>
{
image1.Source = im5;

   {
image3.Visibility = Visibility;
if (flag2)
{
if (image3.Height != 0 || image3.Width != 0)
{
image3.Height = image3.Height - 10;
image3.Width = image3.Width - 10;
}
else
{
button2.Visibility = Visibility;
}

   }
   } }
);
}
// doStuff3();
}
void pDelay2()
{
```

```
int k=20,j=1,m=0;
Boolean flag2 = true;
for (int i = 0; i < 50; i++)
{
System.Threading.Thread.Sleep(150);

   this.Dispatcher.BeginInvoke(() =>
{
image1.Source = im5;
image2.Source = im3;

   {
image3.Visibility = Visibility;
if (flag2)
{
if (image3.Height < 110 || image3.Width < 130)
{
image3.Height = image3.Height + 10;
image3.Width = image3.Width + 10;
}
else
{
//NavigationService.Navigate(new Uri("/Page2a.xaml", UriKind.Relative));
if (j == 0)
{
image3.Source = im8;
j=1;
}
else
{
image3.Source = im9;
j = 0;
}
} }
   }
}
);
}
doStuff2();
}
```

```
   void pDelay()
{
int j = 0, k = 6,m=550,n=550;
Boolean flag = true;

   for (int i = 0; i < 50; i++)
{

   System.Threading.Thread.Sleep(150);

   this.Dispatcher.BeginInvoke(() =>
{
if(flag)
{

   if (j == 0)
{
image2.Source = im1;
image2.Margin = new Thickness(k, 26, 0, 0); k = k + 10;
j = 1; m=m-10;
if (m == 0)
{
flag = false;
}

   }
else
{
image2.Source = im2;
image2.Margin = new Thickness(k, 26, 0, 0); k = k + 10;
j = 0; m=m-10;
if (m == 0)
{
flag = false;
}
} }
   else
{
if (j == 0)
```

```
{
image2.Source = im3;
image2.Margin = new Thickness(k, 26, 0, 0); k = k - 10;
j = 1;

   m = m + 10;
if (m == 550)
{
flag = true;
}
}
else
{
image2.Source = im4;
image2.Margin = new Thickness(k, 26, 0, 0); k = k - 10;
j = 0; m = m + 10;
if (m == 550)
{
flag = true;
}
} }
   }
);
}
doStuff();
}

   void azzDelay1()
{

   for (int i = 0; i < 400; i++)
{
Thread.Sleep(250);

   this.Dispatcher.BeginInvoke(() =>
{

   }
);
}
```

```
    }

    private void image2_ImageFailed(object sender, ExceptionRoutedEventArgs e)
{

    }

    private void button2_Click(object sender, RoutedEventArgs e)
{
NavigationService.Navigate(new Uri("/Page3.xaml", UriKind.Relative));
}

    } }
```

5.1.2 Level1

```
    namespace MD
{
public partial class Page4 :  PhoneApplicationPage
{
public Page4()
{
InitializeComponent();
}
private void image1_ImageFailed(object sender, ExceptionRoutedEventArgs e)
{

    }
int i = 3, m = 2;
private void button3_Click(object sender, RoutedEventArgs e)
{
strings1;
s1 = textBox1.Text;
if(s1 == "crossbreed")
{
textBlock2.Visibility = Visibility.Collapsed;
textBlock3.Visibility = Visibility.Collapsed;
```

$button1.Visibility = Visibility;$
$textBlock5.Text = "YougotitRight.PressContinue";$

```
   }
else
{
textBlock5.Text = "Sorry You are wrong !!!";
textBox1.Text = "";
if (m != 0)
{

   if (m == 2)
{

   textBlock5.Text = "Sorry You are wrong !!!";
textBlock3.Text = "2"; m--;
}
else if (m == 1)
{
textBlock5.Text = "Sorry You are wrong again !!!";
textBlock3.Text = "1"; m--;
}
} else {
NavigationService.Navigate(new Uri("/Page35.xaml", UriKind.Relative));
}
}
}
```

$\texttt{private void button1}_{C}click(objectsender, RoutedEventArgse)$

```
{
```

$NavigationService.Navigate(newUri("/Page5.xaml", UriKind.Relative));$

```
}
}}
```

5.1.3 Level2

Code skeleton:

```
namespace MD
{
public partial class Page10 : PhoneApplicationPage
{
```

```
public Page10()
{
InitializeComponent();
}
int i = 3, m = 2;
private void image1_ImageFailed(object sender, ExceptionRoutedEventArgs e)
{

    }
private void button1_Click(object sender, RoutedEventArgs e)
{
NavigationService.Navigate(new Uri("/Page11.xaml", UriKind.Relative));
}

    private void button3_Click(object sender, RoutedEventArgs e)
{
string s1;
s1 = textBox1.Text;
if(s1 == "topsecret")
{
button1.Visibility = Visibility;
textBlock5.Text = "YougotitRight.PressContinue";
}
else
{
textBlock5.Text = "SorryYouarewrong!!!";
textBox1.Text = "";
if(m! = 0)
{

    if (m == 2)
{
textBlock5.Text = "Sorry You are wrong !!!";
textBlock3.Text = "2"; m–;
}
else if (m == 1)
{
textBlock5.Text = "Sorry You are wrong again !!!";
textBlock3.Text = "1"; m–;
}
```

} else
{
NavigationService.Navigate(new Uri("/Page35.xaml", UriKind.Relative));
} } } } }

5.2 Prototype : 2

In prototype implementation of level3, level 4 and level 5 was been carried.

5.2.1 Level3

Describe the component and write the actual code for the specific component. Code skeleton:

```
namespace MD
{ public partial class Page16 :  PhoneApplicationPage
{ public Page16()
{ InitializeComponent();
System.Threading.Thread startupThread =
new System.Threading.Thread(new System.Threading.ThreadStart(timeDec));
startupThread.Start();
} Boolean flag = true;
Boolean flag3 = true;
void timeDec()
{ int k = 20, j = 1, m = 0;

   for (int i = 0; i < 121; i++)
{ System.Threading.Thread.Sleep(1000);

   this.Dispatcher.BeginInvoke(() =>
{ if (flag)
{ if (flag3)
{ string s1;
s1 = textBlock7.Text;
int num = int.Parse(s1);
num--;
s1 = num.ToString();
```

```
textBlock7.Text = s1;

   if (i == 120)
{
flag = false;
}
} else
{
   } } else { // NavigationService.Navigate(new Uri("/Page36.xaml",
UriKind.Relative));
} }); } // doStuff3();
} private void image1_ImageFailed(object sender, ExceptionRoutedEventArgs e)
{
   } private void button1_Click(object sender, RoutedEventArgs e)
{NavigationService.Navigate(new Uri("/Page17.xaml", UriKind.Relative));
}int i = 3, m = 2;
Boolean flag1 = true;
Boolean flag2 = true;
private void button3_Click(object sender, RoutedEventArgs e)
{string s1;
s1 = textBox1.Text;
if (s1 == "moral support")
{flag3 = false;
button1.Visibility = Visibility;
textBlock10.Text = "E";
textBlock11.Text = "C";
textBlock2.Visibility = Visibility.Collapsed;
textBlock3.Visibility = Visibility.Collapsed;
textBlock5.Text = "You got it Right. 2 of 6 letters are unlocked... Press Continue";
}else{textBlock5.Text = "Sorry You are wrong!!!";
if (flag1 == false)
{
   } textBox1.Text = "";
if (m != 0)
{
   if (m == 2)
{ textBlock5.Text = "Sorry You are wrong !!!";
textBlock3.Text = "2"; m--;
} else if (m == 1)
{ textBlock5.Text = "Sorry You are wrong again !!!";
```

```
textBlock3.Text = "1"; m--;
} } else { if (flag2)
{ image3.Visibility = Visibility.Collapsed;
textBlock5.Text = "You lost a life !!!  ";
textBlock3.Text = "3";
m = 2;
flag2 = false;
} else { NavigationService.Navigate(new Uri("/Page35.xaml", UriKind.Relative));
} } } } } }
```

5.3 Prototype : 3

In prototype 3 implementation of timing constraints, bonus life option and climax was been carried.

5.3.1 Climax

Code skeleton:

```
   namespace MD { public partial class Page33 :  PhoneApplicationPage
{ public Page33() { InitializeComponent();
System.Threading.Thread startupThread =
new System.Threading.Thread(new System.Threading.ThreadStart(pDelay));
startupThread.Start(); }
BitmapImage im1 = new BitmapImage(new Uri("/MD;component/Images/d.png",
UriKind.Relative));
BitmapImage im2 = new BitmapImage(new Uri("/MD;component/Images/dhj.png",
UriKind.Relative));
BitmapImage im3 = new BitmapImage(new Uri("/MD;component/Images/asa.png",
UriKind.Relative));
BitmapImage im4 = new BitmapImage(new Uri("/MD;component/Images/asb.png",
UriKind.Relative));
void pDelay() { int k = 0, j = 1, m = 0,k1=460,k2=300,k3=400;
Boolean flag = true;
Boolean flag1 = true;
for (int i = 0; i < 500; i++) { System.Threading.Thread.Sleep(150);
   this.Dispatcher.BeginInvoke(() => { if (flag) { if (flag1) {
if (j == 0) { image3.Source = im3;
image2.Source = im1;
image2.Margin = new Thickness(k, 222, 0, 0); k = k + 5;
```

```
image3.Margin = new Thickness(k1, 170, 0, 0); k1 = k1 - 5;
j = 1; if (k > 240  k1 < 255) flag1 = false;
   } else { image3.Source = im4;
image2.Source = im2;
image3.Margin = new Thickness(k1, 170, 0, 0); k1 = k1 - 10;
image2.Margin = new Thickness(k, 210, 0, 0); k = k + 10;
j = 0; if (k > 240  k1 < 260) flag1 = false; } } else { image4.Margin
= new Thickness(111,k2, 0, 0); k2 = k2 - 5;
image5.Margin = new Thickness(27, k2, 0, 0); k2 = k2 - 5;
if (k3 > 270) { image6.Margin = new Thickness(277, k3, 0, 0); k3
= k3 - 5; } else flag = false;
   } } else { NavigationService.Navigate(new Uri("/Page34.xaml",
UriKind.Relative)); } }); } // doStuff3(); } } }
```

Chapter 6

RESULTS & DISCUSSIONS

This chapter includes the results and discussions of all the prototypes.

6.1 Prototype : 1

This includes result and screenshots of introduction and first level.

6.1.1 Introduction

When the user clicks on the application icon, the Menu Page is displayed. The Menu Page Consists of the Main Title, and two buttons, PLAY and EXIT which is shown in the fig.6.1. When the user clicks on the PLAY button, this introductory Page is opened where the introduction of the game takes place which is shown in the fig.6.2 and fig.6.3.

Figure 6.1: Introduction

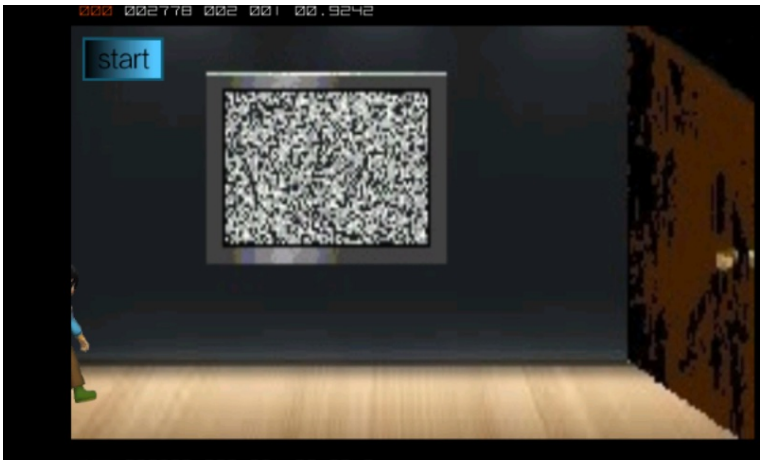

Figure 6.2: Introduction

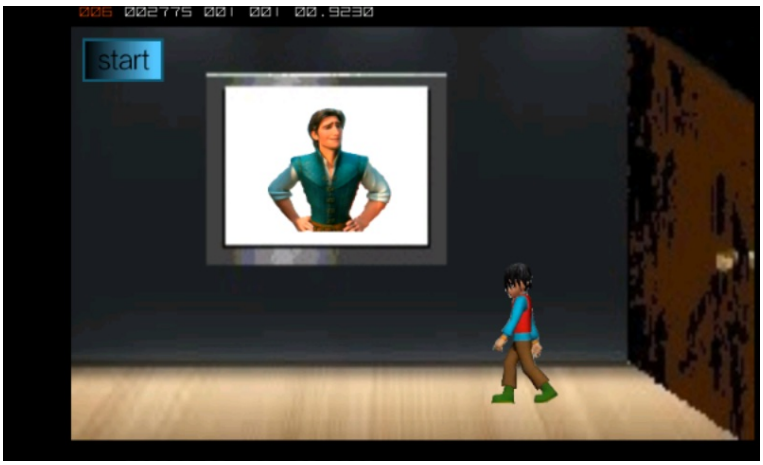

Figure 6.3: Introduction

6.1.2 Level1

After the completion of the introduction the CONTINUE button is enabled. When the user clicks on the CONTINUE button the first question of the first level is displayed which is shown in the fig.6.4. When the user types the right answer, the next question is unlocked.After all the 3 questions of level 1 are answered, the page is displayed that the user has completed the first level which is shown in the fig.6.5.

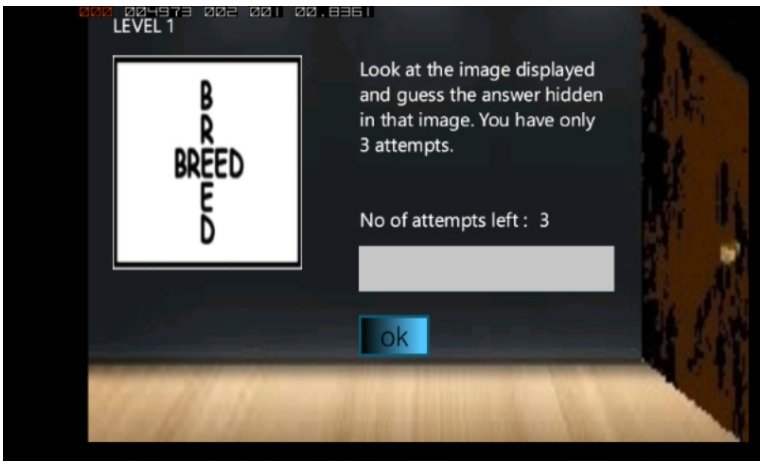

Figure 6.4: Level1

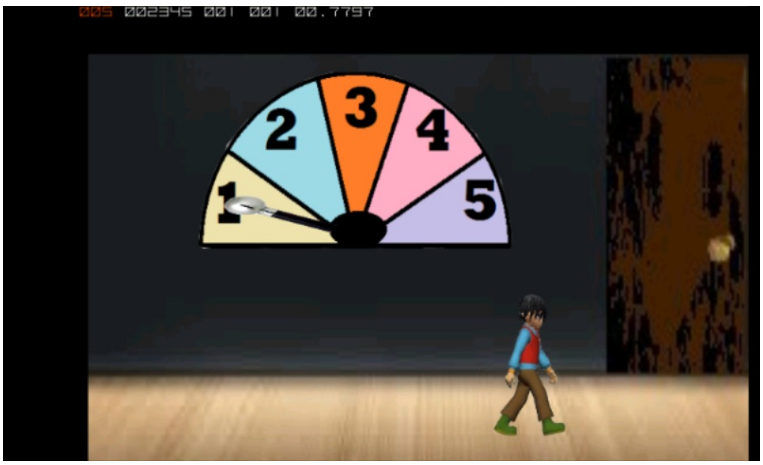

Figure 6.5: Completion of level1

6.1.3 Level2

The fig.6.6 shows the sample of the question displayed to the user, the text field is provided to write the correct answer. If the answer is wrong, the user is notified. After all the 3 questions of level 2 are answered, the page is displayed that the user has completed the second level which is shown in the fig.6.7.

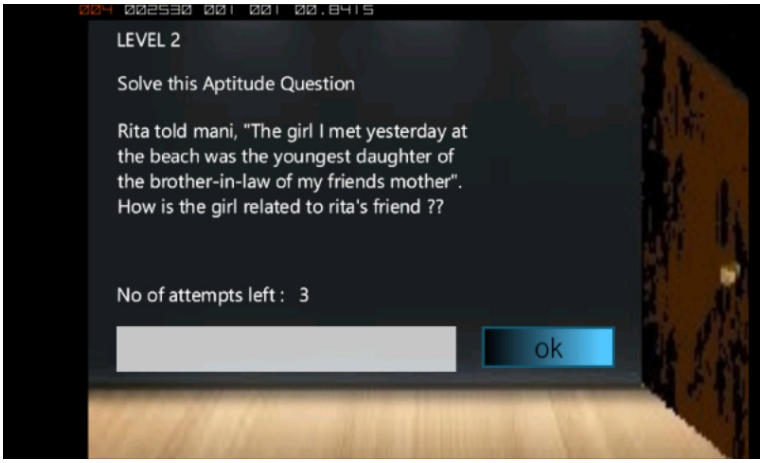

Figure 6.6: Level2

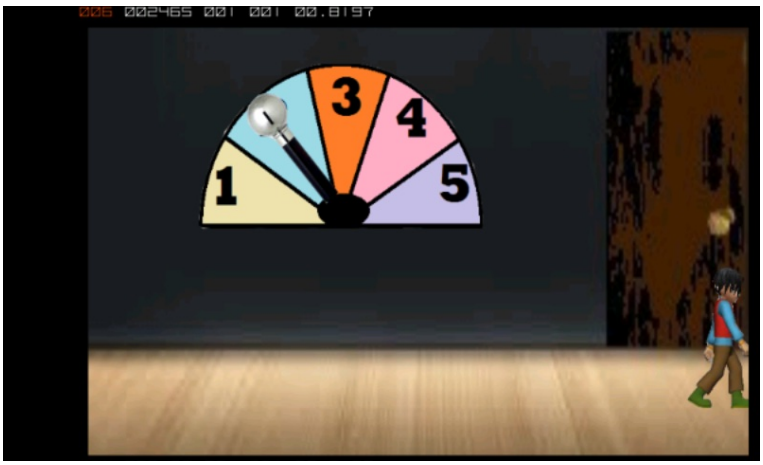

Figure 6.7: Completion of level 2

6.2 Prototype : 2

This includes result and screenshots of third level and fourth level.

6.2.1 Level3

The fig.6.8 shows the screenshot of the 3rd level. Here we have introduced a time-constraint and a bonus life for the user. In this level we have introduced the concept of the password. When the user answers 3 questions correctly, the jumbled password is unveiled, the user should then form this into the meaningful word which is shown in fig.6.9. After all the 3 questions of level 3 are answered, the page is displayed that the user has completed the third level which is shown in the fig.6.10.

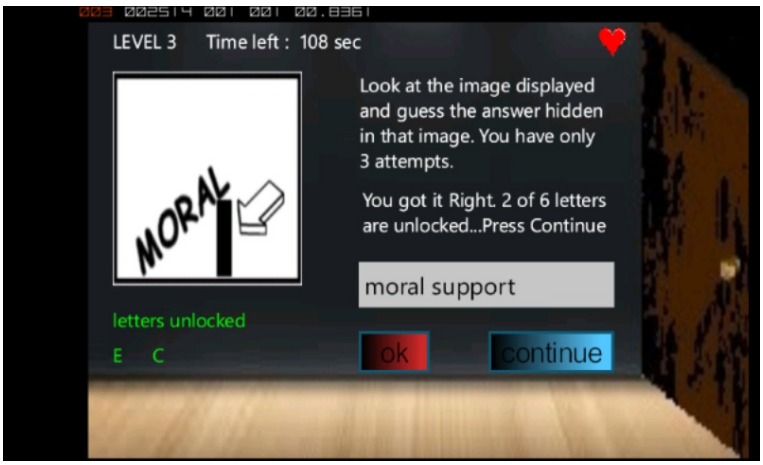

Figure 6.8: Level3

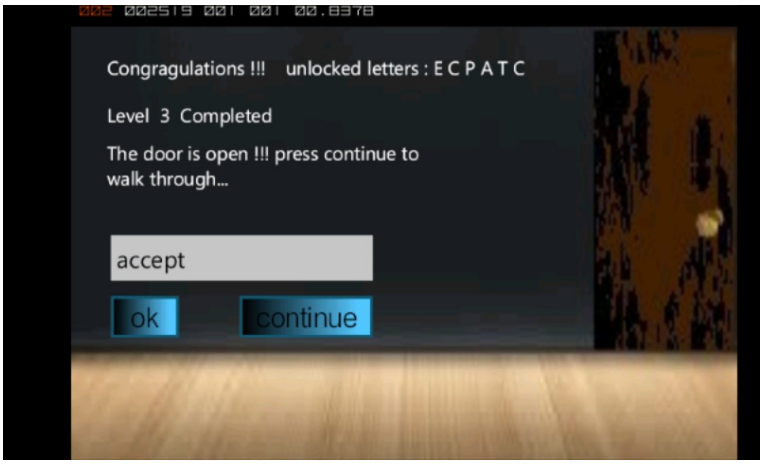

Figure 6.9: Level3

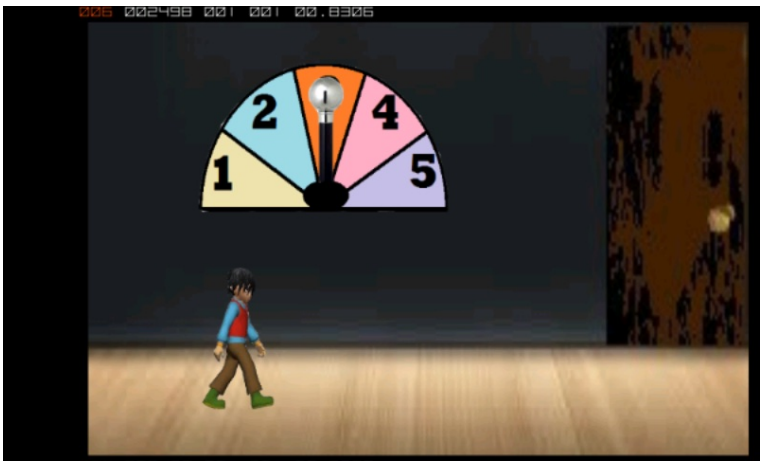

Figure 6.10: Completion of level 3

6.2.2 Level4

The fig.6.11 shows the 4th level of the game. The user is given a set of images and a related image. The user is supposed to give the right answer. After all the 3 questions of level 4 are answered, the page is displayed that the user has completed the fourth level which is shown in the fig.6.12.

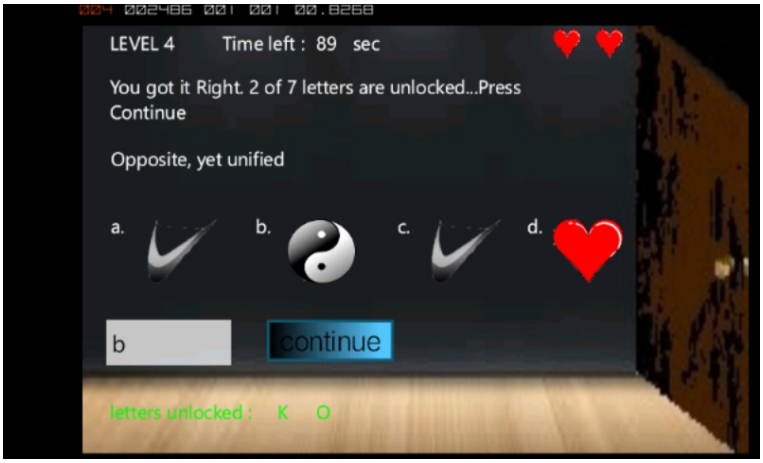

Figure 6.11: Level4

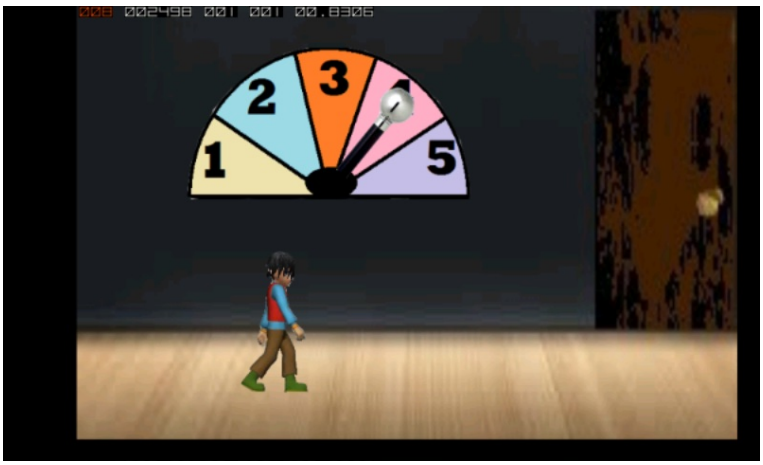

Figure 6.12: Completion of level 4

6.3 Prototype : 3

This includes result and screenshots of fifth level and climax.

6.3.1 Level5

The fig.6.13 shows completion of all the levels.

Figure 6.13: Completion of level 5

6.3.2 Climax

The fig.6.14 shows the last scene of the game, when the user unlocks all the doors he rescues the girl. At the end of the game, the credits are displayed which is shown in fig.6.16.

Figure 6.14: Climax

Figure 6.15: Climax

Figure 6.16: Climax

Chapter 7

Conclusion and Future Scope

7.1 Conclusion

With a large and growing user base that generates large revenues but also raises numerous technological applications, Mobile Games have recently started to attract the interest of the research community. In this work we have tried to identify the availability of the puzzle-based games in Windows phone Operating System. To address this, we have designed and implemented Mystery Doors, a puzzle-based gaming application for Windows phone. Our architecture focuses on puzzle game, which is one of the most important components of the generic game.

7.2 Future Work

This puzzle based game application can be enhanced and can be made more interesting by adding further levels with more features. We can also include functionality of random generation of puzzles. We can also make use of database system and isolated file storage system for storing puzzles and images.