# Design Overview for <<Pikachu game >>

Name: Nguyen Dinh Dung

Student ID: SWH01661- 104772138

# Summary of Program

Describe what you want the program to do… one or two paragraphs.

* The program will demonstrate a game call pikachu. You can play by choose 2 pictures of a same pokemon and then they disappeared. When you make all the pictures disappeared in a period of time you will win. However to choose 2 picture of a pokemon, between them must not have any other pokemon, and the way the 2 pictures connect must less then three zigzags lines.
* I also upgrade it from D level with the matrix contain 12x21 pokemon icons, the function which cause the icons to disappear now become very clear because I name those icon as number and when we click on to identical icon, it will compare the icon number instead of the icon it self. This also helps me to create a matrix and loops to spawn random icon easier because now I only have to loop the icon number instead of the icon picture. I wrap around each icon 0s to create their borders and so I could see my click action more obviously. When click on 2 unidentical icon, the first wrong 1 will disappear but immediately be replaced by it self but back to normal status, no border.

Include a sketch of sample output to illustrate your idea.



# Required Roles

Describe each of the classes, interfaces, and any enumerations you will create. Use a different table to describe each role you will have, using the following table templates.

Table 1: <<NewSelectForm>> details – duplicate

|  |  |  |
| --- | --- | --- |
| Responsibility | Type Details | Notes |
| Show a menu include of three options: start game, instructions, and exit. Include sound | * Window forms * Events: buttonStart\_Click, buttonGuilde\_Click, buttonExitClick * System.Windows.Forms.PictureBox(): custom picturebox( size, margin, padding, Image,….) * Sounds sound2 = new Sounds(AppDomain.CurrentDomain.BaseDirectory +….); |  |

Table 2: <<PlayForm>> details – duplicate

|  |  |  |
| --- | --- | --- |
| Responsibility | Type Details | Notes |
| Main form show the game, time, exit button or new game button | * Window forms * Events: buttonStart\_Click, buttonExitClick * Sounds sound2 = new Sounds(AppDomain.CurrentDomain.BaseDirectory +….); * ProgressBar(toolbox): timer.Start, timer\_Tick, step 1, maximum: 3000, if progressBar = maximum => end game * System.Windows.Forms.PictureBox(): custom picturebox( size, margin, padding, Image,….) | * 3000s * Increment by 1s |

Table 2: <<PanelContainIcon>> details

|  |  |
| --- | --- |
| Value | Notes |
| * arrayIconTg[0].BackgroundImage = global::Pikachu.Properties.Resources * ProcessRandomIcon(Panel panel):   numberMatrixIcon,p1, p2   * processPlay.FindRow: 2 picture the same in a row=> panel.Controls.Remove * processPlay.FindCol: 2 picture the same in a column=> panel.Controls.Remove * processPlay.ZigZugLeftToRight: 2 picture the same in a 3 lines zigzug left to right=> panel.Controls.Remove * processPlay.ZigZugUptoDown: 2 picture the same in a 3 lines zigzug up to down=> panel.Controls.Remove * else: pictureBox.Size = new Size();   pictureBox.BorderStyle = BorderStyle.None; | * Add pokemon icons to panel * Load random icons until all picturebox are filled * Save Icon position to matrix * Possibilities from player choices to delete two same pictures * Else: put 2 picture back to normal, delete border, reduce size. |

# Class Diagram

Provide an initial design for your program in the form of a class diagram.

NewSelectForm

buttonExit

buttonGuide

buttonStart

sounds

PlayForm

InstructionForm

Picture demonstration

Picturebox with text demonstration

buttonBack

buttonContinue

ProgressBar(timer)

Sounds

PlayPanel:

Random pokemon icons, saving to matrix,

processPlay.FindCol

processPlay.ZigZugLeftToRight

processPlay.ZigZugUptoDown

processPlay.FindRow

A screenshot of a computer program

Description automatically generated

# Sequence Diagram

Provide a sequence diagram showing how your proposed classes will interact to achieve a specific piece of functionality in your program.

NewSelectForm

buttonName\_Click(object sender, EventArgs e)

Sounds

* Sounds sound2 = new Sounds(AppDomain.CurrentDomain.BaseDirectory +….);

PlayForm

InstructionForm:

Instruction inst = new Instruction();

inst.Show();

Picture demonstration:

System.Windows.Forms.PictureBox(): custom picturebox( size, margin, padding, Image,….)

Picturebox with text demonstration:

this.textBox1.Text = "”

buttonBack:

NewSelectForm inst = new NewSelectForm();

inst.Show();

buttonContinue:

PlayForm inst = new PlayForm();

inst.Show();

ProgressBar(toolbox): timer.Start, timer\_Tick, step 1, maximum: 3000, if progressBar = maximum => end game

PlayPanel:

Random pokemon icons, saving to matrix,

processPlay.FindCol

processPlay.ZigZugLeftToRight

processPlay.ZigZugUptoDown

processPlay.FindRow

A screenshot of a computer

Description automatically generated