Problem 1:

Interface 1: Xbox 360 gamepad

Context:

Video games usually don’t require mouse or keyboard like desktop computers. In order to have a video game console, gamepads need to be designed to fit many different game types as well as they should fit our hands well to be comfortable to grab. It has a traditional directional pad along with two analog sticks on each side for more accurate control as well as 8 main buttons which can be used for various things in the game.

Strengths

1. It is very well designed and comfortable to grab on hand.
2. It is in right size for many people.
3. Intuitive analog sticks make people learn video game without problem.
4. Guide access button in the middle helps people to get into system menu without jumping through hoops.

Weaknesses

1. It’s got too many buttons. Some people are reluctant to even grab it because, from the way it is designed, people are intimidated by overly complicated button configurations of the game. Although it can be improved by the game designers to use as few buttons as possible, it is easily a temptation to use all the buttons when designing the game, resulting in stiff learning curve. Reducing number of buttons would make too much sense because they are more complicated games which require many buttons. So it has to be improved software-wise, in a way that the game should have enough tutorial to get people started with the system.
2. It doesn’t come in variety of sizes. I found it personally fits my hands perfectly. But many kids with smaller hands found it too big in their hands. Gamepads are already designed in more rounded shape in order to fit different sizes. But it’s not perfect because there are absolute differences in size of the hands among people. Many third party developers came up with smaller gamepads that are more suitable for younger people.
3. Directional pads work awkwardly. After releasing of Xbox 360, many games have avoided using directional pads for their main character movements. However, there are some games which work well only using digital directional pad such as street fighters, pro-evolution soccer games. There have been many user-modeed controllers which address this issue. Eventually Microsoft had to release new controller which significantly improves the way directional pad moves.
Interface 2: credit card

Context:

Credit card has gotten rid of the issue of carrying too much money. It gave people to buy things without paying everything at once. Modern credit cards are successor of what was used in the past because the era of computer. Many concepts were tried using credit stamps, credit note etc.

Strengths

1. It got rid of risk of carry too much money
2. It gave people a chance to buy lots things that they can afford to pay at once.
3. It’s not very complicated to learn how to use it

Weaknesses

1. Although it should encourage people to spend easily, it’s only limited to people who are authorized to use it. Only people with good credit standing are issued with a credit card. It is not designed for everyone. There has to be more clear process of how people can get one.
2. There are very little means of checking the ownership of the card. The card only has numbers associated with the card in the magnetic strip. There is no practical way of detecting cloned cards unless person who’s the original owner of the card realizes that it is been cloned and request for suspending the account.
3. Although magnetic strips helped people to use it easily without typing number into a credit-card terminal, newer technology has to be implemented such as RFID, for easier use and security purposes

Interface 3: Sony noise canceling headphones MDR-NC11A

Context: It is supposed to get rid of low-frequency noise that can be heard. It is not easy to find an environment where we can listen to music without getting interfered by noises. It was introduced to address this problem

Strengths

1. It gets rid of low-frequency noise.
2. It comes in many different color for different needs of people
3. It has built in volume control in case the music device in the place where we can’t reach easily.

Weaknesses

1. Noise canceling device is too bulky and heavy. It is in inconvenient spot. Earbud often falls out of the ear because of the device that is attached in the middle.
2. Noise canceling device requires AAA battery. It adds more mass to already-heavy device.
3. Clip on the device is very weak that it doesn’t hold itself onto sleeves or other things we want it to hold on to. They should have used stronger clip.

Problem 2:

Person 1 – Nam

Nam is a senior in the university who has been using Windows Vista since the first day it was released. He had several issues with usage of Windows vista.

1. He often complains about how Windows Vista can slows down the productivity because of the new security mechanism that was designed to request people for administrators’ right. While it is designed to protect the system from malicious software, the way it work (how screen goes dark when it is prompted) can be quite annoying especially when computer is doing some heavy work. If it is necessary due to the security reason, it can still be a simple popup, not the screen which makes everything dark.
2. Taskbar on the bottom has been around since Windows 95, but Windows never natively supported rearrangement of the tasks. It can be achieved with small program by community developer. But it’s rather buggy. People are used to rearranging things in excel and tabbed web browsers. It is a simple implementation that can improve people’s productivity greatly.
3. Microsoft constantly changes the way updates are installed. First through applications individually, and through windowsupdate.com, and now through windows-update which is built in the Vista. Since system changes every time they release windows, he needs to figure out a new way to learn it. Microsoft needs to stick with well-designed update system in order to minimize the confusion.

Person 2 – JungYoon

Nam is a senior in the university who uses MacOSX.

1. When selecting multiple files with keyboard, he has to select the first file and hold shift and press up or down to select more files. But when more files are selected accidently, (say pressing down), he wants to unselect the last file by pressing up. But this makes him choose one more file to the top of the initially selected files. This can be advantage or disadvantage depending on how familiar people are with the system.
2. Size of windows cannot be adjusted from all four sides. There are some applications such as adobe readers or word processor which need to be adjusted when working with multiple documents. But OSX requires people to use the bottom right corner to adjust the window.
3. Confirmation window for simple yes no needs auto focusing. When confirmation window popup to ask use a simple yes-or-no question (or ok-cancel), it is not auto focused, so that he cannot
use left or right arrow to choose it. He has to choose the window by himself using keyboard or mouse. It can be improved easily by implementing windows with 4-side frame.

Person 3 – Young

Young is my sister who uses tablet PC with windows XP. Tablet version of the windows doesn’t really have anything special for tablet, except some hand-writing recognition. They use exactly the same interface as regular windows.

1. Using tablet to select a menu can be quite tedious. First of all, the start button is still located in the corner of the screen which requires her to move the arm all the way, and select very specific menu she wants to go to in order not to lose focus of the menu. More simple start-menu access can be helpful (like big icons on the bottoms like Mac OSX.)

2. Double click is still required to launch application. While many people learn to use double clock these days, using tablet makes it very difficult because there is no guarantee that she can hit the same stop twice using tablet pen. There are some corrective measure built in, but it can be triggered when she didn’t mean to. They should have option to change things into sing-touch icon.

3. When typing password, programs usually hide what the user types. But tablet PC’s often show it on the screen for verification purposes. It makes her very hesitant to even get into e-mail while she’s in public. It will be helpful if some gesture based password system was implemented with tablet version of windows. Or at least the system that could translate the gesture to characters for the password.

Problem 3: Participant: JungYoon

<table>
<thead>
<tr>
<th>Trial</th>
<th>Correctness</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>correct</td>
<td>14.109</td>
</tr>
<tr>
<td>1</td>
<td>correct</td>
<td>6.088</td>
</tr>
<tr>
<td>2</td>
<td>correct</td>
<td>6.617</td>
</tr>
<tr>
<td>3</td>
<td>correct</td>
<td>8.537</td>
</tr>
<tr>
<td>4</td>
<td>correct</td>
<td>5.622</td>
</tr>
<tr>
<td>5</td>
<td>correct</td>
<td>5.962</td>
</tr>
<tr>
<td>6</td>
<td>correct</td>
<td>4.544</td>
</tr>
<tr>
<td>7</td>
<td>correct</td>
<td>3.330</td>
</tr>
<tr>
<td>8</td>
<td>correct</td>
<td>4.736</td>
</tr>
<tr>
<td>9</td>
<td>correct</td>
<td>3.922</td>
</tr>
<tr>
<td>10</td>
<td>correct</td>
<td>2.890</td>
</tr>
<tr>
<td>11</td>
<td>correct</td>
<td>3.963</td>
</tr>
<tr>
<td>12</td>
<td>correct</td>
<td>4.236</td>
</tr>
<tr>
<td>13</td>
<td>correct</td>
<td>4.003</td>
</tr>
</tbody>
</table>
Tn = T1 * n ^ (-a)

3.165 = 14.109 * 22 ^ (-a)

a = 0.48354555415794513254874279232986

The program was made on XNA framework on Visual Studio 2005
Source Code

```csharp
using System;
using System.Diagnostics;
using System.Collections.Generic;
using Microsoft.Xna.Framework;
using Microsoft.Xna.Framework.Audio;
using Microsoft.Xna.Framework.Content;
using Microsoft.Xna.Framework.GamerServices;
using Microsoft.Xna.Framework.Input;
using Microsoft.Xna.Framework.Storage;

namespace QWERTY
{
    static class Program
    {
        static void Main(string[] args)
        {
            using (Qwerty game = new Qwerty())
            {
                game.Run();
            }
        }
    }

    public class Qwerty : Microsoft.Xna.Framework.Game
    {
```
GraphicsDeviceManager graphics;
SpriteBatch spriteBatch;
SpriteFont spriteFont;

public Qwerty()
{
    graphics = new GraphicsDeviceManager(this);
    Content.RootDirectory = "Content";
    graphics.PreferredBackBufferWidth = 800;
    graphics.PreferredBackBufferHeight = 500;
    Window.AllowUserResizing = true;
}

protected override void Initialize()
{
    base.Initialize();
}

protected override void LoadContent()
{
    spriteBatch = new SpriteBatch(GraphicsDevice);
    ContentManager content = Content;
    spriteFont = content.Load<SpriteFont>("Gulim"); // add fonts to display texts in window
    answerKeys.Add(Keys.Z);
    answerKeys.Add(Keys.Y);
    answerKeys.Add(Keys.X);
    answerKeys.Add(Keys.W);
    answerKeys.Add(Keys.V);
    answerKeys.Add(Keys.U);
    answerKeys.Add(Keys.T);
    answerKeys.Add(Keys.S);
}

protected override void UnloadContent()
{
}

KeyboardState lastKeyboardState;
KeyboardState currentKeyboardState;
List<Keys> answerKeys = new List<Keys>();
List<Keys> keySequence = new List<Keys>();
String message = "Type last 8 letters of the alphabet backwards. \nIt starts when 'z' is pressed when ends when 's' is pressed";
String resetMessage = "Type last 8 letters of the alphabet backwards. \nIt starts when 'z' is pressed when ends when 's' is pressed";
String typing = "typing";
List<TimeSpan> trials = new List<TimeSpan>();
List<bool> correctness = new List<bool>();
Stopwatch sw = new Stopwatch();

protected override void Update(GameTime gameTime) // this is the function that gets called 60 times per second (60 frames per second)
{
    lastKeyboardState = currentKeyboardState;
    currentKeyboardState = Keyboard.GetState();

    if (lastKeyboardState.IsKeyUp(Keys.Escape) && currentKeyboardState.IsKeyDown(Keys.Escape))
        this.Exit();

    if (lastKeyboardState.IsKeyUp(Keys.Z) && currentKeyboardState.IsKeyDown(Keys.Z))
    {
        sw.Start();
        message = typing;
    }

    foreach (Keys key in currentKeyboardState.GetPressedKeys())
    {

```
if (lastKeyboardState.IsKeyUp(key) && currentKeyboardState.IsKeyDown(key) && sw.IsRunning)
    keySequence.Add(key);

if (lastKeyboardState.IsKeyUp(Keys.S) && currentKeyboardState.IsKeyDown(Keys.S))
{
    sw.Stop();
    trials.Add(sw.Elapsed);
    sw.Reset();

    bool same = true;
    if (keySequence.Count == answerKeys.Count)
        for (int i = 0; i < keySequence.Count; i++)
        {
            if (keySequence[i] != answerKeys[i])
                same = false;
            break;
        }
    else
        same = false;
    correctness.Add(same);
    keySequence.Clear();
    message = resetMessage;
}

base.Update(gameTime);

protected override void Draw(GameTime gameTime)
{
    graphics.GraphicsDevice.Clear(Color.White);
    spriteBatch.Begin();
    spriteBatch.DrawString(spriteFont, message, new Vector2(0, 0), Color.Black);

    string trialsPrint = "";
    for (int i = 0; i < trials.Count; i++)
    {
        trialsPrint += "Trial #" + i + " : ";
        if (correctness[i])
            trialsPrint += "O : ";
        else
            trialsPrint += "X : ";
        trialsPrint += trials[i].ToString();
        trialsPrint += "\n";
    }
    spriteBatch.DrawString(spriteFont, trialsPrint, new Vector2(0, 70), Color.Black);
    spriteBatch.End();
    base.Draw(gameTime);
}
Jungyoon didn’t have a particular trick, He just remembered the location on the keyboard and slightly improved the time overall.

Problem 4: Say $A = 548$, $B = 420$ (from lecture note)

(a) 

$$MT = A + B \times \log(n)$$

$$548 + 420 \times \log(12) = 2053.7\text{ms}$$

(b) 

50/50:

$$0.5 \times (548 + 420 \times \log(4)) + 0.5 \times (548 + 420 \times \log(8)) = 1598\text{ms}$$

75/25

$$0.75 \times (548 + 420 \times \log(4)) + 0.25 \times (548 + 420 \times \log(8)) = 1493\text{ms}$$

90/10

$$0.9 \times (548 + 420 \times \log(4)) + 0.1 \times (548 + 420 \times \log(8)) = 1430\text{ms}$$

Minimum choice time: the dynamic menu

$$548 + 420 \times \log(4) = 1388\text{ms}$$

Maximum choice time: the static menu

$$548 + 420 \times \log(8) = 1808\text{ms}$$

(C) Constant $A$ and $B$ can’t be calculated by experiment. The limit is that it cannot be applied accurately in the real world experience.