|  |
| --- |
|  |
| Assignment 1 |
| SWFR ENG 2AA4 |
|  |
|  |
| **2/28/2014** |

|  |
| --- |
|  |

**Jadeesuan Anton – 1213386**

**Connor Hallett – 1158083**

**Spencer Lee – 1224941**

**Nicolas Lelievre – 1203446**

**Zhiting Qian - 1217485**

**Classes/modules**

The coding language used in this project is Visual Basics.NET (VB.NET). This language is composed of a graphical outlay which can be designed through a visual interface or code. Upon each object, instances of actions can trigger events programed through subs. In additions to subs, functions can be made and called with references, both of which work similar to standard functions.

|  |  |  |
| --- | --- | --- |
| Function/Sub | Variables Involved | Description |
| **Form\_Load** |  | This function contains code that is executed as soon as the form, the interface which the user sees is loaded. In this specific instance, graphic instances created by coding and intuitive design are both used.  **Intuitively**: This part contains a checkers background image, labels, command buttons and a menu strip. Labels are used to show game titles and current mode and display relevant information to the user, the background image is set to the entire form and the initially invisible command buttons and selection are used for custom setup.  **Code**: Coding wise, the code uses determined points on the form to set plausible picture boxes in which red and black checker pieces are going to be placed, 32 total boxes are used, only covering black boxes. These 32 boxes forms a virtual control array and will be used to game play |
| **Get\_Click** |  | The Get\_Click sub is a protected sub operating under the virtual control and detects the clicks on each picture box. This sub also has a function call to determine which piece is actually placed |
| **Piece\_Position** |  | This is a created function used to decide which piece is going to be placed inside the playable picture boxes |
| **Initial\_Position** |  | Menu strip button that calls another function to get an array of the initial positions and sets them as such |
| **Custom\_Set** |  | Custom setup is used when the user wants to setup the orientation of pieces, this specific sub will change the Boolean Custom to True and also clear all of the existing pictures on the picture Boxes, In addition, it will enable access to otherwise invisible elements to decide which specific piece is going to be placed on the click area. |
| increasing |  |  |
|  |  |  |

**Form1**

The interface of Form1 consists of the game board. This includes a menu strip, 24 picture boxes (12 per side) as well as an autoscroll bar. Menu strip waits on the user to click on it. There are three options for the user to select: file, view and settings.

*Still needs behavior (semantics)*

*Description of uses relation*

*Relation with requirements*

*All dat private stuff*

***Internal review/eval of design*** *??*

*COMMENT THE CODE*

***Test report***

**Sub: Form\_load**

**Description:** The form\_load sub is done as soon as the program is loaded as a result, this sub becomes necessary for declaration of initial variables and drawing graphic pieces, as a result, the group will create the checker board array using this sub. In addition to drawing graphics, this sub also linked an array to the picture boxes in order to handle future action.

**Interface Specifications:**

Draw the Checkers board.

References: None

**Variables:**

|  |  |  |
| --- | --- | --- |
| **Variable Use** | **Variables** | **Variable Type** |
| Output | C\_track | Integer |
| Output | InitialX | Integer |
| Output | C\_trackarray | PictureBox Array(31) |
| Output | M\_custom | Boolean |
| Output | M\_standard | Boolean |
| Input | ErrorClick | PictureBox |

**Internal Implementation:**

For I in range 0 to 7

For j in range 0 to 3

C\_trackarray(c\_track) = new PictureBox

C\_trackarray(c\_track).position = (I\*50 + InitialX,j\*50)

C\_track +=1

ErrorClick.sendtoback

Next

Next

**Protected Sub: MakePiece**

**Description**: This sub will handle all drawing functionalities within the requirements. It has internal methods the can draw the normal checker board setup and will handle clicks during custom mode. This sub will later have methods to allow it to in cooperate save and load. This sub will not handle movements which will be implemented in a future sub as a part of assignment 2.

This sub was used to unify the custom and standard setups, both of which are triggered by clicks on valid checker board locations. The MakePiece sub is a leaf in the hierarchy and thus is a user interface. In keeping with modularity and logic, this sub was made instead of having each other module draw its own pieces.

**Interface** **Specifications**:

Draws checker pieces upon clicks from the user

Deletes Checker Pieces if custom mode is not selected or if piece limit is reached

**Variables:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Variable Use** | **Variable Name** | **Variable Type** | **Declaration Location** |
| Input | M\_custom | Boolean | Global |
| Input | M\_standard | Boolean | Global |
| Internal | ThisPB | PictureBox | Local |
| Internal | Gamesetup | Integer Array(31) | Local |
| Output | Notice | String | Local |
| Output | Future | String | Local |

**Internal Implementation:**

Standard Setup Logic:

For I =0 to 31

Trackarray(i).image = Gamesetup(i).image

Next

Output is a set of actions and messages, messages represent logic for states that appear in later assignments.

|  |  |  |
| --- | --- | --- |
| **Logic: Mode** | | **Output Action** |
| M\_custom | M\_standard | Message Notice |
| Not M\_custom | M\_standard | Setup Standard |
| M\_custom | Not M\_standard | Get Click |
| Not M\_custom | Not M\_standard | Message Future |

List of Modules

QuitGameToolStripMenuItem\_Click

Form\_FormClosing

Form1\_Load

StandardToolStripMenuItem

CustomToolStripMenuItem

ResetToolStripMenuItem

GameTimer\_Tick