**Final Project – Programming 10** Gurtaj Khabra

*~~xxxx~~ = removed / xxxx = added*

*Reaction Mechanics* (name probably changes later) Specification:

In this application, players must react quickly to the targets which appear in picture boxes on screen and use multiple methods to destroy them (pressing a key or clicking a target). As the game goes on, more targets will appear and they will move faster (by decreasing timer interval and increasing speed). The game will continue until the player fails to destroy 5 targets or hits the incorrect key 5 times. As you destroy the targets, your score which is displayed in a label will consistently increase. The application will include multiple forms and timers. *\*\*\*The methods include pressing a key (q, w, e, r, d, f) or clicking (left or right).*

*Objects:*

* Labels
  + In Game
    - lblScore shows the current score
    - lblTime shows the amount of time the player has been playing
    - ~~lblMethodKeys shows number of targets destroyed by keys~~
    - ~~lblMethodClick shows number of targets destroyed by clicks~~
    - lblTargetsDestroyed shows the number of targets destroyed
  + End Screen
    - lblPlayAgain asks the player whether they want to play again
    - lblFinished informs the player of their score
* Picture Boxes
  + Menu
    - picTitle shows the title of the game
    - picStart starts game when clicked
    - picHelp displays information/instructions when clicked
      * Opens a child form
    - picExit closes application when clicked
  + In Game
    - New picture boxes will be created when a target is missed
    - New picture boxes will be created to display targets
  + End Screen
    - picYes restarts the game when clicked
    - picNo returns you to the menu when clicked
* Buttons
  + No buttons – Picture boxes are used instead
* Menus
  + mnuProgram
    - mnuNewGame
      * starts a new game and resets score, time, and misses
    - mnuInstructions
      * displays the instructions in a child form
    - mnuExit
      * Closes the application
* Timers
  + timerMain
    - keeps track of and updates time (located in a label), it will also increase timerTargets interval when it hits certain points (10 seconds, 20 seconds, 30 seconds) and will increase the target speed as well
  + timerTargets
    - timer which creates the targets

*Variables:*

* intScore; static variable set to 0 which changes when targets are destroyed
* intTime; static variable set to 0 which keeps track of time
* intMisses; static variable which keeps track of the number of targets missed
* chrKeyPressed; variable which determines which key is pressed
* chrKeyRequired; variable which determines the key required to destroy a target
* blnTargetClicked; variable returns true or false depending on whether you right click or left click a target and which one is required
* intInterval; variable which determines the amount of time between one target appearing to the next, it decreases as time increases
* ~~intTargetDrop~~ (intSpeed); determines how fast the target moves down
* intLoop; to use in loops
* intMethod; a random variable which determines the method on the target

*Constants:*

* intSCOREPERTARGET = 150, whenever a target is successfully destroyed, the player will gain this many points
* intMAXMISSES = 5, the maximum number of targets a player can miss

*Loops:*

* Do…Loop; to repeatedly create targets until the player hits 5 misses
* For…Next; to create new picture boxes for a red cross based on the number of targets which the player misses

*Functions:*

* RndInt(): To randomize the type of target (q,w,e,r,d,f,left,right)

*Sub Procedures:*

* MissTarget takes intCrosses(intMisses) and creates a new picture box for the total number of misses
* MoveTarget takes picTarget[number] and creates a target at a random location near the top of the client rectangle (located in timer) and uses intTargetDrop to move targets down
* TargetInterval takes intTime and changes the interval of TimerTargets after certain points (10seconds, 20seconds, etc.)

*Parameters:*

* intCrosses(intMisses)
* picTargets(8)
* intTargetDrop
* intLowNum
* intHighNum

*Images:*

* pictures of a targets which will be modified to display different methods
  + 8 different pictures with either the letters q, w, e, r, d, f or the words left or right on them

*Arrays:*

* picCross(intMisses); array which creates picture boxes based on the number of times a target is missed (declared as picture box)
* picTargets(8); array which creates targets based on method(1-8), declared as picture box

*Child Forms:*

* Instructions; to explain the rules of the game
* Menu; a window with start/instructions/exit and a title