Hidden Treasure

# Learning Objectives

1. Create a working menu using Event Driven Programming
2. Identify, Describe and Explain the Event Driven Features of the menu
3. Create a working easy game mode
4. Identify, Describe and Explain the Event Driven Features of the easy game mode
5. Make improvements based on user feedback
6. Extend Further: Apply what you have learned to create a difficult game mode

# Introduction and User Requirements

You have been commissioned by a games developer to create a game called Hidden Treasure using Event Driven Programming. The program should be developed using c# and Visual Studio.

|  |  |  |
| --- | --- | --- |
| **The end goal – What it could/should look like** | | |
| **A working menu with working buttons linking to other forms** | **A working easy mode game form** | **A working difficult mode game form**  The Gold Star Programmer task:  This is an extension task so you need to decide on all the design features based on the provided user requirements |

## LO1\_BRONZE: Create a working menu using Event Driven Programming

|  |  |
| --- | --- |
| **User Requirements** | **Example design** |
| **Create a menu for the game that uses buttons to open other forms**   * There should be a main menu form with a suitable title (game name) * There should be a suitable image * There should be an easy mode option (button) that opens a new form that will become the easy mode game screen with further development * There should be a difficult mode option (button) that opens a new form that will become the difficult mode game screen with further development |  |

|  |
| --- |
| **Programming Guide**   1. Open visual studio and create a new windows forms application project. Remember you must save your project into your visual studio 2019 projects folder 2. Use Label, picture and button interface objects from the tool bar to design your form 3. Use the properties menu on the right of the screen to change text, fonts and importantly the name given to the objects   **Using the suggested property names will made programming easier later as these are the names used in this guide**  Text property of the form  Label – unnamed as it is for visual reasons only and will not be used in the program code  Picturebox – again unnamed as it will not be used in the code. Import the image provided or choose your own  Button – property name = btnEasy  Button – property name = btnDifficult |
| 1. Add a new form to the project and call it: **EasyMode**    1. Project --> Add New Windows Form --> (Name it EasyMode) 2. Add a new form to the project and call it: **DifficultMode**    1. Project --> Add New Windows Form --> (Name it DifficultMode)   **IMPORTANT**  You are now ready to program the instructions for the two buttons. Event Driven Programs trigger code instructions to run when events happen. Events must be “linked” to a method (procedure/function) for them to work. Events are usually created by double clicking on a user interface object  The Event method (Procedure/function) code should be created by Visual Studio. You should only be typing code that is highlighted in the examples below.   1. **Program the button: btnEasy**    1. Double click on your button – this should create the \_Click event for your button and open the code editor    2. Type the following instructions to open the EasyMode form when the button is clicked      1. **Program the button: btnDifficult**    1. Double click on your button – this should create the \_Click event for your button and open the code editor    2. Type the following instructions to open the DifficultMode form when the button is clicked |

# LO2: Identify, Describe and Explain the Event Driven Features of the menu

|  |  |  |  |
| --- | --- | --- | --- |
| Event Driven feature | Describe:  What is it?/What are they? | Example using examples: Give an example of this being used in your program and explain what it does/why it is needed | Can you think of other examples and why they might be used? |
| Events |  |  |  |
| Event Handlers |  |  |  |
| Triggers |  |  |  |

## LO3\_Silver: Create a working easy game mode

|  |  |
| --- | --- |
| **User Requirements** | **Example design** |
| **Create a menu for the game that uses buttons to open other forms**   * There should be an EasyMode game screen * There should be labels with instructions on how to play the game * There should be 9 buttons (These will all be connected to the same event handler so be careful not to double click) * There should be a label to display the number of clicks * There should be a label to display the time remaining * An appropriate message should be displayed if the player finds the treasure and the game should end * An appropriate message should be displayed if the player runs out of time (15 seconds) and the game should end |  |

|  |
| --- |
| **Programming Guide**   1. Open your project in visual studio if it is not already open 2. Find your EasyMode form design 3. Use Label, picture and button interface objects from the tool bar to design your form 4. Use the properties menu on the right of the screen to change text, fonts and importantly the name given to the objects   **Using the suggested property names will made programming easier later as these are the names used in this guide**    Text property of the form  3 Labels – unnamed as they are for visual reasons only and will not be used in the program code  9 Buttons – property name = left as default (1-9) as these are meaningful enough for the scenario  Label – property name = timeLabel  Label – property name = clicksLabel |
| **IMPORTANT**  The Event method (Procedure/function) code should be created by Visual Studio. You should only be typing code that is highlighted in the examples below.   1. **Create the form load event – this is needed to set all of the default values for the form. This will allow the game to be played multiple times without having to end the whole program**    1. Double click on the form background – visual studio will open the code editor and create the form load event    2. Create the global variables for the EasyMode form:      * 1. Program form load (this might have jumped to the next page):        1. **Create a single event and event handler for the 9 buttons**    1. Use your cursor to select all of the buttons at ones:      * 1. Select the event view from the property menu to the right of your screen      * 1. Create a click event called **ButtonClick.** Type ButtonClick and press the enter key on the keyboard this will create an event handler that will run when any of the 9 buttons are clicked and open the code editor  1. **Program the ButtonClick Event Handler**    1. These instructions need to disable the button that was clicked, check whether the treasure was behind it and run the win instructions if they player has guessed correctly: |
| **Adding a Timer**  Remember: The Event method (Procedure/function) code should be created by Visual Studio. You should only be typing code that is highlighted in the examples below   1. **Open your EasyMode form design and add a timer to it from the tool box. (you can search using the search box at the top of the tool box)**      1. **Set the interval property to 1000 (1000 = 1 second)** |
| **EXTEND: Adding a Timer**  This is more challenging because you will need to amend other event handler methods (procedures/functions)   1. **Double click on the timer – visual studio with create the timer \_tick event and open the code editor** 2. **Create a global variable to store the time remaining**      1. **Set up the times in the Form Load Event Handler**      1. **Stop the timer when the game is won – ButtonClick Event Handler**      1. **Finally Program the Timer1\_Tick Event Handler** |

# LO4: Identify, Describe and Explain the Event Driven Features of the EasyGame Mode

|  |  |  |  |
| --- | --- | --- | --- |
| Event Driven feature | List three examples used in the EasyMode form: | Explain the purpose of each event/event handler that you listed | Can you think of other examples and why they might be used? |
| Events/Event Handlers |  |  |  |

|  |  |
| --- | --- |
| Identify the global variables used in the program and their datatypes |  |
| Explain the difference between a global and local variable |  |
| Explain why local variables are often preferred by programmers |  |

# LO5: Make improvements based on user feedback

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | **Name of tester:** | | **Does the menu form work as expected?** | | **Does the EasyMode correctly detect “a win” and show an appropriate message?** | | **Does the EasyMode end when the timer runs out?** | | **Can you suggest any improvements?** | | **Review of user testing – Explain whether you are going to make any improvements based on the user testing** | |

## Extend Further: Apply what you have learned to create a difficult game mode

|  |
| --- |
| **User Requirements** |
| **Use the tick boxed to monitor and track your progress**  **Create a DifficultMode Game Screen**   * There should be a DifficultMode game screen * There should be labels with instructions on how to play the game * There should be 16 buttons (These will need to be connected to the same event handler so be careful not to double click) * Treasure should be placed at random “behind” one of the buttons * There should be a label to display the number of clicks * There should be a label to display the time remaining * An appropriate message should be displayed if the player finds the treasure and the game should end * The game should end after 15 seconds and an appropriate message should be displayed * A bomb should be placed at random behind on of the buttons    + This should not be same button as where the treasure is placed   + If the button with the bomb is selected then the game should end and an appropriate message should be displayed |