Peyton Wolf

4/20/2025

Cst-250 Programming in C# ||

Mark Smithers

Grand Canyon University

Activity 5

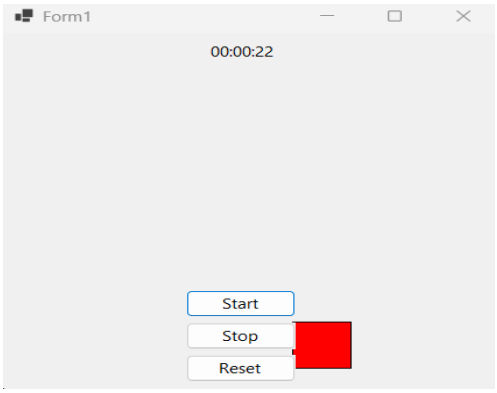
## ***Loom Videos***

- Timer: https://www.loom.com/share/1f5b21f35cd648b7844a0bb034d043c3?sid=8d82a54d-9d7f-444f-afa6-1aa8b63bc985
- Whack-A-Mole:Part1: https://www.loom.com/share/cfd02e7a18234b95a3e982ed08ed3715?sid=c046d170-34f4-4c2d-bb52-38fe116ec20d
  Part2: https://www.loom.com/share/10d02b7385af466f9f13c44ec7d19bd3?sid=df40b004-c073-4475-a7dc-184b608eef09

## ***GitHub Link***

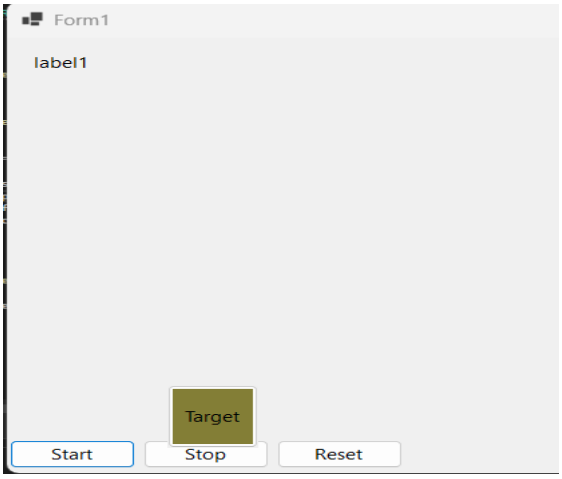
https://github.com/KnoxHighStax/CST250/tree/main/Activity5

## ***The Timer Control***




In this screenshot we are demonstrating the creation of a timer event where we are just displaying a timer for a user to keep track of how long they play the game for! First the application will start, and the game will populate with its various buttons, label and custom FrmGame_Paint element, being the square. Whenever the user is ready to start playing the game they will first click on the “Start” button and the game form will call on the FrmGame_Paint method to draw the game onto the board for the user to be able to click on! Once the Square has been clicked it will wait some time and then redisplay is for the user to be able to click on again. This process will continue recursively until the user decides to click on the “Stop” button. From here the square will stop moving and the timer at the top of the form will stop ticking to give up the total time that you were playing this game for.

## ***Creating a Whack-A-Mole Game***

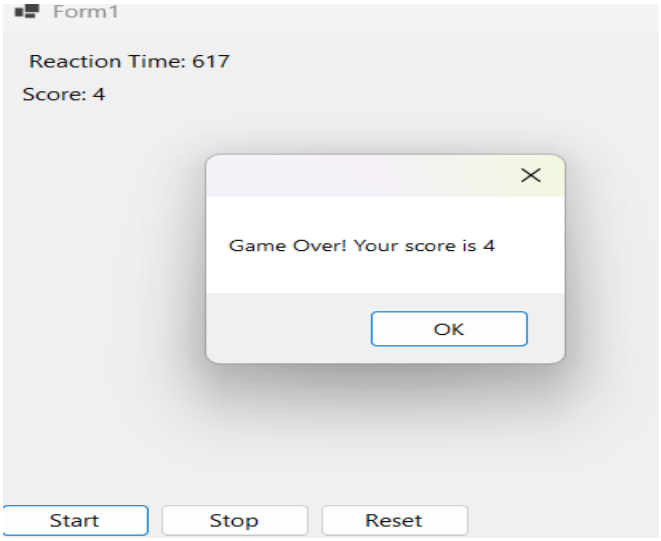



In this screenshot we have created a game form where the user can click on the “Start” button and the “Target” button will move randomly around the form until it has been clicked on by the user. After a few seconds then the “Target” button will re-display itself on the game form to be clicked again. This will continue recursively until the user clicks on the “Stop” button

## ***Adding Score Calculations***

In this screenshot we have added the ability to track how long it takes the user to react to the “Target” button displaying and the user clicking on it so that the user can help track how fast their movements are. As the player successfully clicks on the “Target” button the “Score” label will update with the total amount of successful click completed by the user.

***Penalty Click***

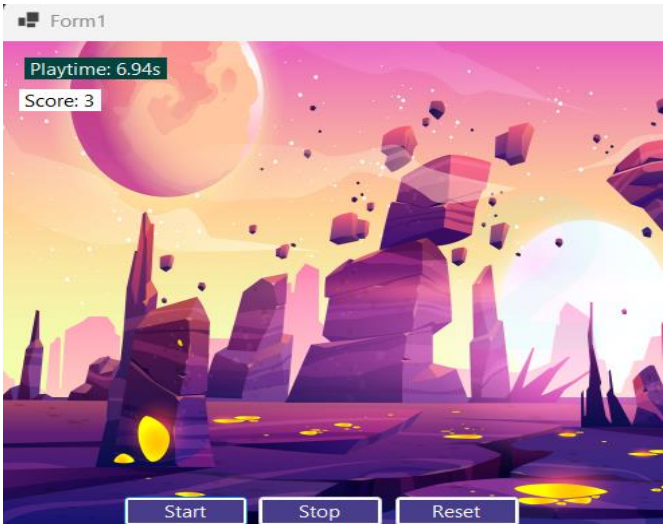



In this screenshot we are demonstrating how we have added penalties for missing clicks! If at any point while the user is playing the game, try to click on the “Target” button. If at any point they miss the “Target” button, then the game will end and a “MessageBox” will be displayed for the user letting them know that the game has ended and letting them know their final score for this round!

***Creating Levels***

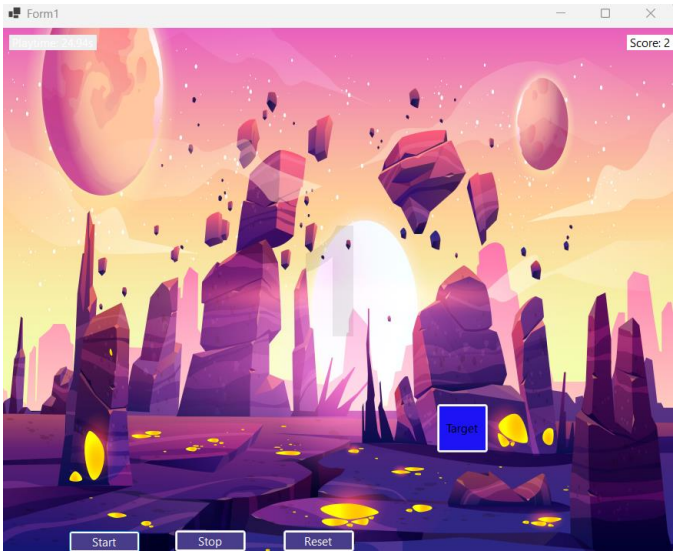

In this screenshot we are demonstrating how we have added “Levels” to our game. As the user is playing the game and continuously clicks on “Target” button without missing then they can start progressing through the “Levels’. For this game when the user clicks on the “Target” button 3 times successfully then the “LargeNumber” in the background will progress up to the next level and update the background color to match the level changing.

***Custom Graphics***



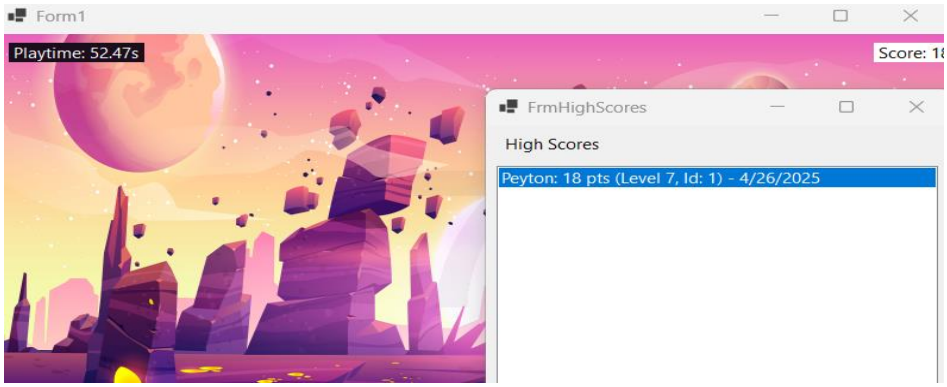

In this screenshot we are just demonstrating how we have updated our game form with a fun background image to be the main environment of our game. Instead of having some boring blank white background or just a boring color we can add fun background displays to make it more appealing for the user to interact with.

***Account for Reform Size***



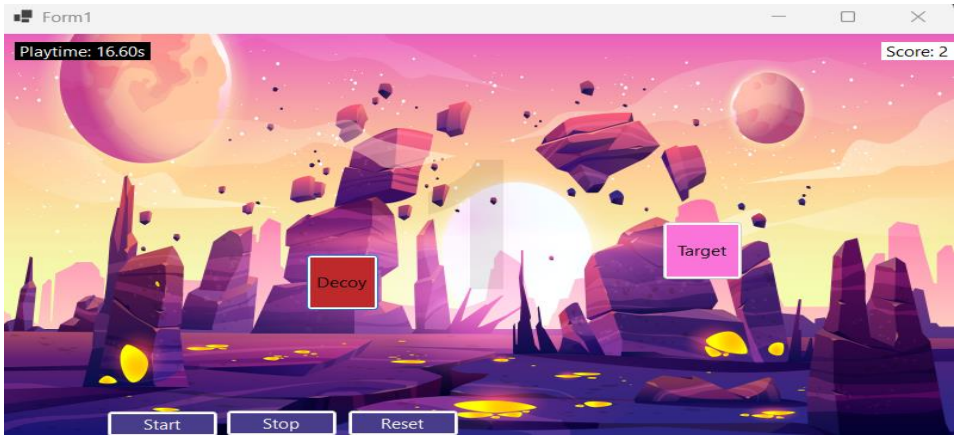

In this screenshot we are demonstrating how we have added the ability for the user to resize the form, and the buttons and labels will move with the form being resized. Before the user attempted to resize and make the form bigger the buttons would stay in their spots and be in the middle of the form and if the user was to make the form smaller then how the default generates then the buttons would be out of visual view. By locking the button's locations, they will move with the form being resized, avoiding all of these issues from occurring.

***High Score Hall of Fame***



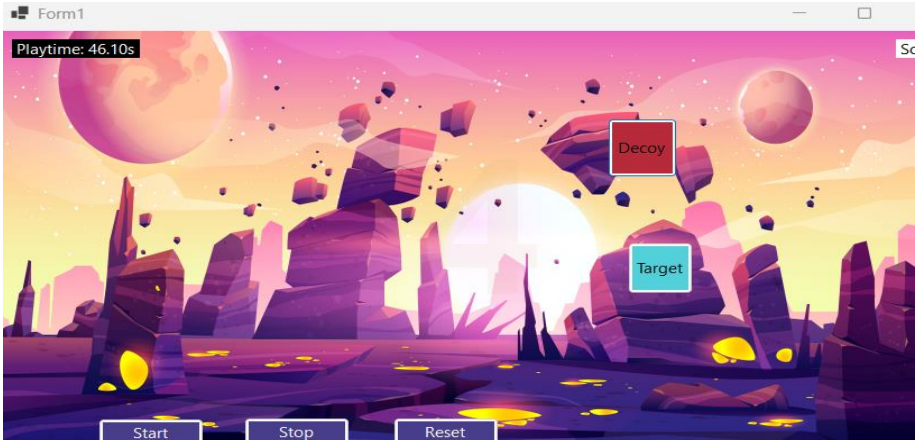

In this screenshot we have added a new form to track the High scores of a specific game or if multiple people want turns playing to see who this did better is how that can be easily tracked. When the game comes to an end it will ask the user for their name and once the user inputs their name a new form will display (FrmHighScores) where the High scores will be displayed for this game session. We will display in the listbox a string with their name, the total points that were scored, the level the user got to, their user Id to quickly identify them and the date of when this high score was made.

## ***Decoy***



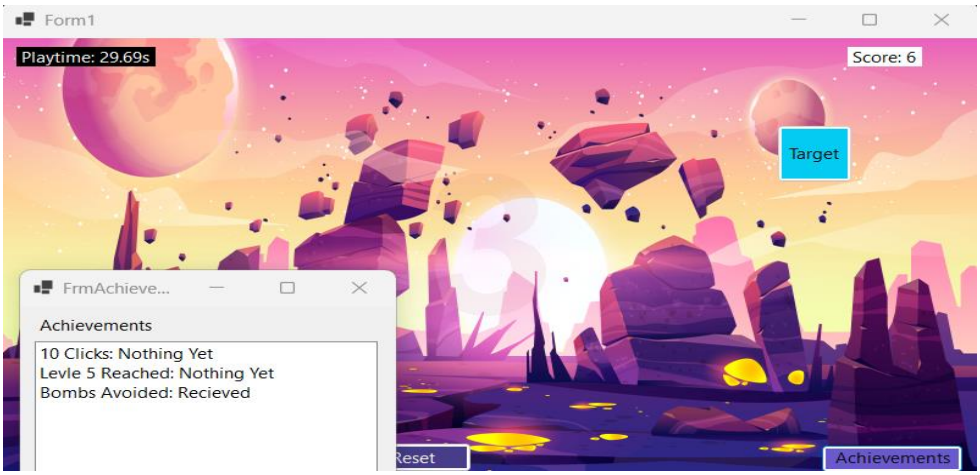

In this screenshot we are demonstrating how we have added a new button like “Target”, but this one will be our “Decoy” or bomb (Don't want to hit). This will serve as a way to try to trick the user, at this point of the project instead of if the user clicks and misses the target but if the user also clicks on the “Decoy” by mistake then it will trigger the game to lower the score of the user, so now there are losing points.

## ***Increasing Difficulty***




In this screenshot we are demonstrating how we have added the ability for our “Target” button to decrease a small amount in its total size each time the user has a successful click on that button. This will over time make things alot more difficult for the user; as they approach a score of 10, 20 and 30 the button “Target will be much smaller than are “Decoy” since the decoy won't change sizes and is also the starting size of the “target” button.

## ***Rewards***




In this screenshot we are demonstrating the reward system that we have added for this application. Here we have created a new form called “FrmAchievements” where the user will be able to see the achievements that are available for the they to gain while playing! We have also added a button on our main game form (Achievements) that when clicked will display this new form of the achievements. Once the user completes an achievement the “FrmAchievements” will update that the user has “Received” the achievement.

1. What was challenging?
   I would have to say that the most challenging aspect of this project would have been managing the game’s difficulty progression and ensuring smooth interactions between different components. For instance, dynamically adjusting the target size and movement speed while maintaining the game balance required careful tuning. Another challenging aspect was implementing the bomb avoidance mechanic, ensuring the bomb’s position didn’t overlap with the target and tracking player penalties accurately.
2. What did you learn?
   I would have to say with this project I have learned a lot about how to structure a WinForms applications with multiple forms and class; this will help to keep our code more modular. I gained a ton of experience with game mechanics like timers(“timeMoveTarget_Tick”) dynamic UI updates, and event handling (“btnTarget_Click”). I also explored graphics rendering with “DrawLargeNumbers” and user input validation with “FrmNameEntry”. With this project we have also reinforced concepts like OOP and LINQ for sorting high scores.
3. How would you improve on the project?
   To help improve this project I would try to focus on both gameplay and code structure. First, I would add more visuals and maybe some auditory feedback, like animations for successful clicks or sound effects whenever a level-up occurs. We

could also expand the game mechanics by adding some sort of power-ups or multiple moving targets/bombs, this could help to increase the complexity and replay value of the game.

4. How can you use what you learned on the job?
   With these skills that we have developed in this project, they could translate very well to professional software development fields. Experience with WinForms provides a foundation for other GUI frameworks like WPF or MAUI, which are typically used in enterprise desktop applications. The Game logic of timer-based events, collision detection and state management can be applicable to simulation tools or interactive training software. The techniques for handling and displaying data mirror real-world tasks in CRUD applications and the object-oriented principles, like encapsulating achievements and scores in dedicated classes help to demonstrate scalable code organization for larger projects.