Team Project #2

PHASE 1

Build a Bouncing Ball Simulation Program

This project constructs a prototypical mechanical system simulator in which an object elastically scatters, at right angles, from a bounding surface. In this phase, the object effectively executes one-dimensional motion. The only user interaction is to start and stop the motion, control the object’s speed and to reset the elapsed timer. Subsequent project phases will introduce more complex geometry, more user interaction, database functionality and full project documentation.

This project will be tracked using a GANTT chart, created using Microsoft Project. The first phase of the project will be documented with an interim project report.

General Procedure:

1. After surveying the project specification, use Microsoft Project to create a GANTT chart to plan the work. This will be done in consultation with the instructor.
2. Develop and test the code, per the specifications and plan.
3. Write the interim project report for phase 1.

Project Specifications

1. This project is built using C#.
2. This is a GUI, Windows Forms Application.
3. Create an image of a rectangular enclosure, at least 500 pixels on a side. Use a picture box to display this image on the form.
4. Create another image (sprite) of a disk (or ball, if you prefer). Control the alpha channel to make the corner pixels transparent, showing only a circular central portion (the disk). Use a picture box to display this in the form, on top of the image of the enclosure.
5. At the bottom of the form, place a Start button. When pressed, its text becomes “Stop”. When that is pressed, the text changes back to “Start”.
6. Also along the bottom of the form, place a text box and another button. The button’s text is “Update Speed”. This text box will be used to input a number signifying the speed of the simulator.
7. In a third area along the bottom of the form, place two labels. The first label’s text is “Hits”. The second label, beside the first, will be used to display the number of times that the simulated ball hits and bounces off of the bottom edge of the rectangular enclosure.
8. In a fourth area along the bottom of the form, place another label. This will be used to display the elapsed time, in the form hh:mm:ss.ss . For example 10:27:32.41, 10 hours, 27 minutes and 32.41 seconds. Beside this label, place a button, labeled “Reset”.
9. With the above structure in place, construct a simulator that operates as follows:
   1. Initial conditions:
      1. The ball is located in the center of the enclosure, not moving.
      2. The Start/Stop button shows “Start”.
      3. The Speed value is set at 5. It can accept user input between 1 and 100.
      4. The Hits value is initialized to 0 (zero).
      5. The elapsed time is initialized to zero: “0:00.00”.
   2. When ball is stopped and the user presses the Start/Stop button:
      1. The text of the Start/Stop button changes to “Stop”.
      2. The elapsed time starts to increase and is displayed on the form in the designated label. Display to hundreds of a second.
      3. The ball moves either upward or downward (with a 50-50 chance of going either way), at a speed (represented as 5 in the text box), such that it takes 0.5 second to travel from its central position to the boundary of the enclosure.
      4. When the ball reaches the boundary of the enclosure, it reverses direction, traveling at the same speed in the opposite direction. It stays within the enclosure.
      5. Whenever the ball bounces off of the bottom edge of the enclosure, the Hits value is incremented and its display is updated.
   3. The user changes the speed of the ball’s movement by entering a value (between 1 and 100-values outside this range are rejected) in the text box and pressing the Update Speed button.
      1. When the user does this, the value input is checked by the software. If the value is not in the range of [1,100], the input value is cleared from the text box and a warning message is displayed on the form.
      2. When an acceptable value is input and the Update Speed button is pressed, the ball immediately starts to move at the new speed.
      3. The speed of the ball is linearly proportional to the speed value supplied by the user. For a user input value of 5, the ball traverses the enclosure in 1 second. For an input value of 100, the ball traverses the enclosure in 1/20 second. For an input value of 1, the ball traverses the enclosure in 5 seconds, etc.
   4. When the ball is moving and the user presses the Stop button:
      1. The ball stops moving, wherever it is.
      2. The text of the Start/Stop button changes from “Stop” to “Start”.
      3. The timer for the elapsed time stops. The display shows the elapsed time at this moment.
   5. When the user presses the Reset button, the elapsed time is reset to zero, but the timer continues to run, if it currently is running, increasing the displayed elapsed time.
10. Document the source code to fully describe its structure and operation.
11. The interim project report will contain the following sections:
    1. Section 1: Overall description of the program: what it does, what controls the user has, a description of the appearance of the user interface.
    2. Section 2: Description of the architecture of the program. What are the major parts or modules of the code? Write this so that someone unfamiliar with the project and application will understand it. You may assume that the reader understands basic software concepts. The purpose of this is to orient someone who needs to work with this, possibly to make upgrades in the future (this will happen in phase 2). This person needs to see the big picture of how the system is built.
    3. Section 3: Technical reference describing the complexities inherent in this system. This is the underlying details of how the system works. This supplements the code documentation and should be sufficient to guide someone having to work on or modify the internals of the system (this will happen in phase 2).
    4. Section 4: User’s instructions showing how to run the program and what its behavior is with the operation of its various controls.