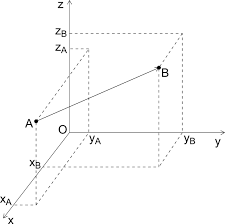
**WPF C# APPLICATION DEVELOPMENT EXERCISE**

**Background description**

Consider to have a machine that moves in a 3 dimensional Cartesian space (for example a 3D printer): every point of the volume of the machine is described with the X, Y, Z coordinates of a reference system that will be call "MACHINE SYSTEM" or "MAC".

Coordinate X,Y,Z of point B



Coordinate X,Y,Z of point A

MAC Reference System

If the machine moves in a random position within its volume, then the coordinates of the point P representing the position of the machine at a given time, would continuously change.

To move from point a P1 of coordinates X1,Y1,Z1 to point a P2 of coordinates X2,Y2,Z2 , the coordinates of the various axes would vary as follows:

* X would change from X1 to X2
* Y would change from Y1 to Y2
* Z would change from Z1 to Z2

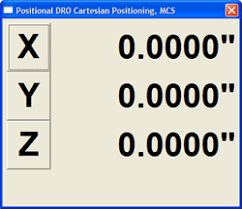
Graphical user interface

Description automatically generatedIn real life, the variation from one position to another will be regulated by the speed set by the movement of the machine.

In conclusion, the X coordinate of the current position will gradually change from X1 to X2 in the time period determined by the speed of movement and so will the Y and Z coordinates.

On the machine tools, the position of the head in real time, is displayed on a device called DRO (Digital Read Out) where the X,Y,Z values are shown.

<https://en.wikipedia.org/wiki/Digital_read_out>

**Scope of the exercise**

The exercise consists of creating a graphical component that simulates the DRO.

The aesthetic form can be very simple, see example on the side to get a graphic idea of how the control might looks like.

This component should be preferably built in C# WPF or Windows Form.

In order to have the DRO working, it is necessary to create, in C# language, a machine simulator that allows to change the position of the virtual machine in real time.

The virtual machine will "move continuously" from a random position to another random position and the DRO will update accordingly.

The connection between UI and random data generation must be made using the binding capabilities of WPF .

**Description of the Simulator**

The simulator must be a separate DLL from the main application and also shall be written in C#.

The DLL is called by the DRO user interface above described.

The speed of the virtual machine is set to a parameter set in a configuration file with a default speed of 100 mm per second.

The size of the X,Y and Z axis is also defined in a configuration file.

The simulator, at its initialization, start form the 0,0,0 position and will generate a new random position within the range of the axes: this random position (P1) need to be reached by the axis, once the position is reached, then a new random position is generated, reached and the loop will continue.

Steps:

1. Start from 0,0,0 for X,Y,Z axis (P1).
2. An additional random position (P2) is created and represents the next target.
3. To simulate the real motion, the machine will progressively updates the positions from P1 to P2 and, under timer (250 milliseconds) updates the intermediate positions from P1 to P2.

The implementation of a real kinematics is not required, it is enough to divide the vector P1-P2 into a certain number of steps necessary to reach the position P2 from P1 based on the travel time required to travel the P1-P2 distance at the selected speed.

1. Once the P2 position is reached, P1 will become P2 and the cycle starts again at point 1.

This loop will allow the simulator to generate motion positions that will update the DRO.

**Data Saving**

To simulate a data saving on a database, it is required that the current location (wherever the machine is at a certain time) is saved on a database (XML or DB file or any other preferred form of DB).

The saving is done by timer every "X" seconds (the duration of the timer is configurable in configuration file).

**User Interaction**

The interface must also have two buttons:

* When pressing the first button, the XYZ coordinates of the machine position at that time must be saved in a text file (it must be possible to choose the path and file name in the configuration file)
* When the second button is pressed, the current database must be saved in a text file. Also in this case it must be possible to select the path and file name in the configuration file;

The text file format must consist of one or more lines, each containing X, Y, and Z.