COSC2429 – Introduction to Programming

Assignment 3 – Group Project

Let's send people to Mars (hypothetically). We need to build a program for the spaceship that will bring our astronauts to Mars. This includes multiple subprograms:

1. Basic info: The subprogram allows astronauts to keep track of the current journey to Mars. It must be able to do these things (at least, you can add more):

- Inquire the info on current status of the spaceship (current velocity, distance from Earth, distance from Mars, estimated time of arrival, current fuel level, current fuel burn rate, current general health of the spaceship, and current general health of the crew members).

- Even though the numbers are hypothetical, you have to make them sensible. For example, if I run this subprogram 1 minute later, distance from Earth must increase, while distance from Mars, estimated time of arrival and current fuel level must decrease.

Hint: The equation is: distance = velocity*time. You should find a way to keep the record of previous status and do some calculations.

2. Daily journal: The subprogram allows astronauts to write anything to the database (findings, ideas, etc.). It must be able to do these things (at least, you can add more):

- Keep the record of the current date/time, user's name and content of the journal entry.

- Allow the user to read but not UPDATE previous entries.

Hint: For simplicity, we don't have any user/password management here. You also don't have to connect it to a database, text files will be fine. You can do database if you want (no extra points).

Creativity part:

If your group has 3 members, you have to think of 1 more subprogram of your choice to write.

If your group has 4 members, you have to think of 2 more subprograms of your choice to write.

Be creative! Think outside of the box but inside the spaceship (**What will the astronauts need?**).

HD point: You have to connect all subprograms into 1 big program

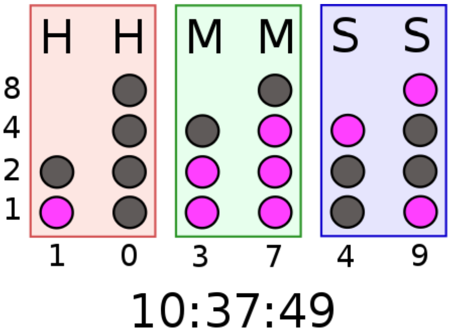
You don't have to build a GUI for the program, we can do it in command line. If you want to make it harder for your team by building a GUI, feel free to do so (no extra point).

In total, you will have about **5 minutes to demo your programs** in the lab on week 12. Feel free to contact me if you need help. Goodluck have fun!

If you are stuck on being creative, you can choose to do this instead (This will replace 1 subprogram only so if you are a team of 4, you still have to think of 1 more subprogram):

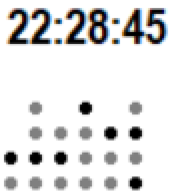
Binary Clock

A binary clock displays the time of day in binary format. Modern binary clocks have six columns of lights; two for each of the hours, minutes and seconds. The photo below shows a binary clock displaying the time "12:15:45":



For example, to show the time "10:37:49": In the second picture, the binary values increase from the bottom to the top row. Lights on the bottom row have a value of 1, lights on the row above have a value of 2, then 4 on the row above that, and finally a value of 8 on the top row. Any 24-hour time can be shown by switching on a certain combination of lights.

Now you have to draw it in turtle. You also have to include a normal digital clock to check your binary clock's correctness. The results should be looking similar to this:



Hints:

- Use multiple turtles
- Use turtle.write() to write out the time on the digital clock
- Use turtle.dot() to draw the dots on the binary clock
- You can set different colors, size, font, position of clocks, as long as they are visible to human eyes.

Now you have a program to draw a static binary clock. Can we make it a working dynamic clock? This means we want to program to continuously get the new time and update both the binary clock and the digital clock accordingly. The result should be like in the video on Canvas Assignment 3 details page.

Hints:

- Use while and for loop (of course)
- Use multiple turtles (again)