**NetLOGO assignment assigned Wednesday, March 30, 2016** **Due: Friday, April 8th or before.**

**Send via email to jacobson@cs.uni.edu with SUBJECT: CS 1025 assign – with the attachments of the .nlogo file(s).**

***Note: You can answer Q#2 in email note or via paper copy or with an attached file.***

1. Create a NetLogo model. In your model create 16, 32, 48 or 64 turtles by using a SLIDER. Have the "**odd**" turtles take **oddSteps** steps forward. Have the "**even**" turtles take **evenSteps** steps forward. Note that **oddSteps** and **evenSteps** are SLIDERS. Save your NetLogo model and send it to jacobson@cs.uni.edu as an attachment. Have each of the **odd** turtles draw some type of Polygon, such as a hexagon or pentagon or heptagon or triangle. Have all of the **even** turtles draw a different polygon type of polygon than the odd ones did. Get creative with NetLOGO.

   *You certainly may add additional features to this, if you wish. Different color for the odd and for the even, different shapes than the default turtle shape, spiral of polygons, etc. Different pensizes so they leave different sized trails when they are dragging their tails.*

## remainder

**remainder** *number1 number2*

Reports the remainder when *number1* is divided by *number2*. This is equivalent to the following NetLogo code:

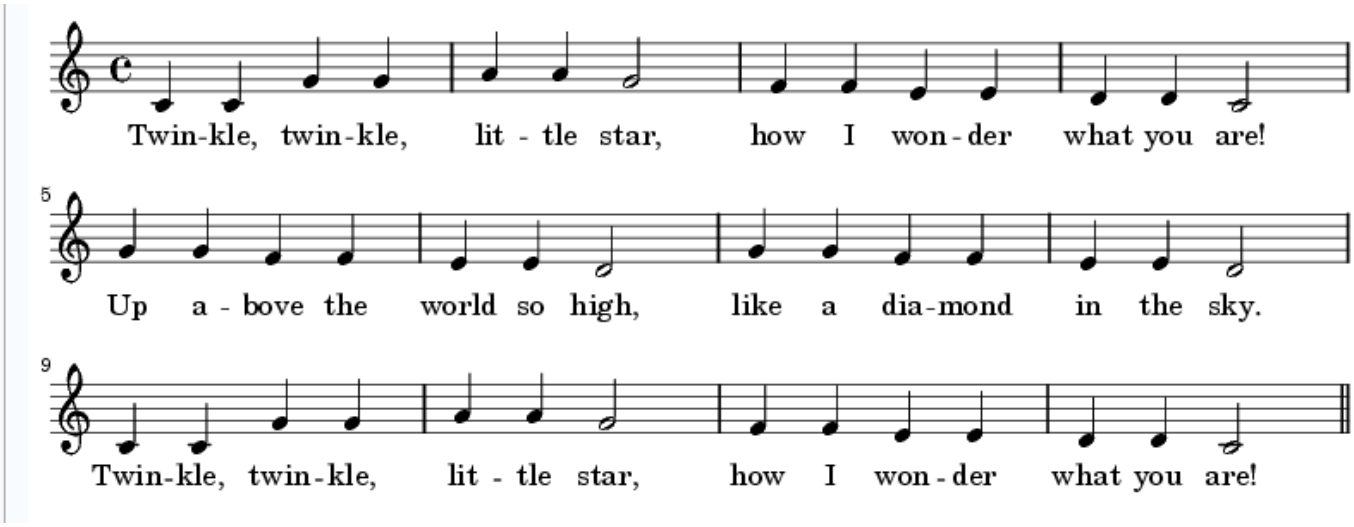
```
number1 - (int (number1 / number2)) * number2
```

```
show remainder 62 5
=> 2
show remainder -8 3
=> -2
```

2. In the NetLogo models library, open File->Models Library; Sample Models -> Chemistry & Physics -> GasLab -> **GasLab Maxwells Demon**.

   - Click on the Information tab and carefully read the sections on "**What is it?**" and "**How it works.**" Try out the model. Read again. Take some notes. Play with the model. Then answer these questions.
     - Who is "Maxwell's Demon," and what does the demon do?
     - How does the model detect when two balls "hit" each other?
   - Click on the Interface tab. Change the number of particles to 500, click "setup," and then click on "go."
     - In the middle interface, balls with numbers appear. What do these represent?
     - What does the color of the balls represent?
     - The balls are not always the same color as when the simulation started out. What makes them change their color?
     - Describe what happens when your models is allowed to run over a long period of time (you may want to increase the speed). *(SPEED **increase** or **decrease** from NORMAL SPEED in NETLOGO are very useful.)*

3. Create the NetLogo model that will play the song **Twinkle Twinkle Little Star** once when the user of your model clicks a button. Give the user choices for the length of a quarter note in the song to be any of the following: 0.1, 0.2, 0.3, 0.4, 0.5 or 0.6 for the various TEMPO choices. Send the NetLogo model file as an attachment to jacobson@cs.uni.edu on or before the due date deadline. Use any musical instrument that you like.




Using the list approach (using a **foreach** statement with two lists, one for note, one for duration of the note) is encouraged, but not required. See the http://www.cs.uni.edu/~jacobson/025/logo/bday.html page for more details.

**Much improved use of NetLogo to "sing" Happy Birthday**

```
extensions [sound]

to playHappyBirthday
    ( foreach [ 60 60 62 60 65 64   60 60 62 60 67 65   60 60 72 69 65 64 62   70 70 69 65 67 65 ]
              [ .3 .1 .4 .4 .4 .8   .3 .1 .4 .4 .4 .8   .3 .1 .4 .4 .4 .4 .4   .3 .1 .4 .4 .4 .8 ]
       [
           playNote ?1 ?2
       ]
    )
end

to playNote [ theNote theLength ]

   sound:start-note "TRUMPET" theNote 65
   wait theLength
   sound:stop-note "TRUMPET" theNote

end
```

**`playNote ?1 ?2`**

**?1** There are two lists in the foreach loop. The first list consists of musical notes for **theNote** such as 60 for C, 62 for D, 65 for F, 64 for E and 72 for C (the C an octave above middle C, i.e. 60 + 12 = 72***).*** ***60 60 62 60 65 64 etc.***

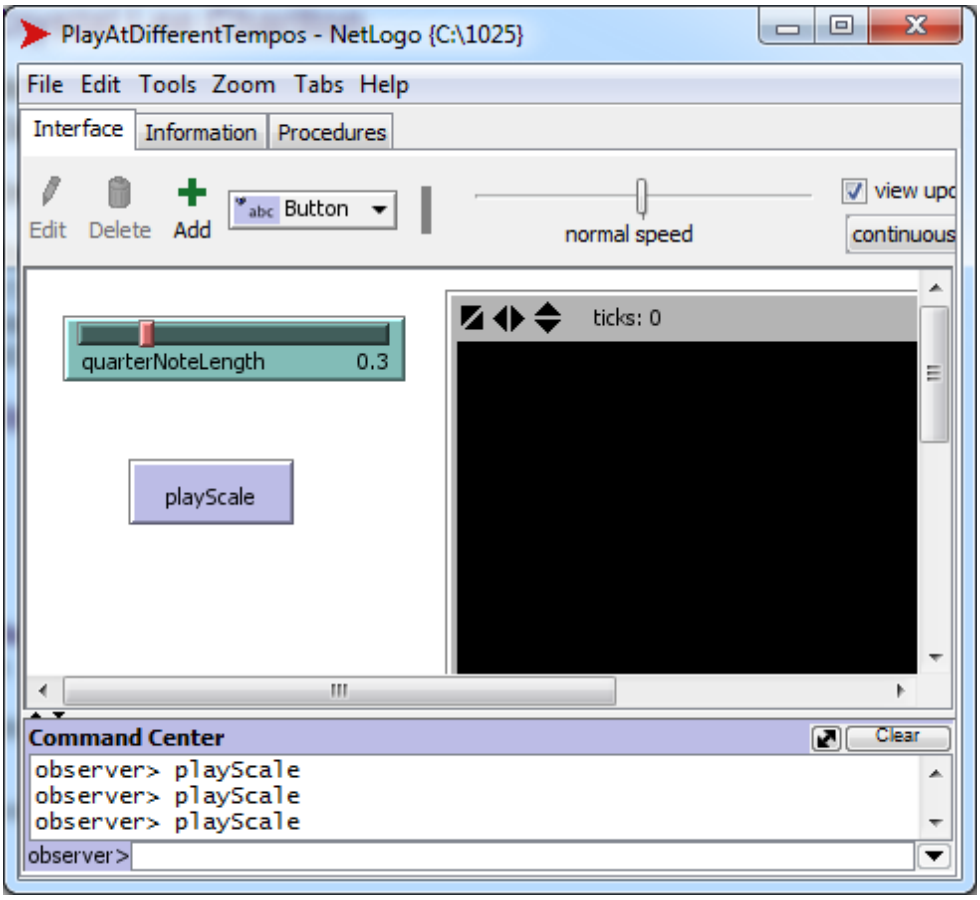
**?2** The second list is durations in seconds for **theLength** of the note. .3 is 3 tenths of a second. 0.4 is 0.4 second and theLength for a quarter note in Happy Birthday song***.*** ***.3 .1 .4 .4 etc.***

**Much improved use**

```
extensions [sound]

to playHappyBirthday
    ( foreach [ 60 60 62 60 65 64   60 60
                [ .3 .1 .4 .4 .4 .8   .3 .1
      [
          playNote ?1 ?2
      ]
    )
end

to playNote [ theNote theLength ]

   sound:start-note "TRUMPET" theNote 65
   wait theLength
   sound:stop-note "TRUMPET" theNote

end
```

```
extensions [sound]

to playScale
  ( foreach [ 60 62 64 64 65 67 69 71 71 72 60  64 67 72 ]
            [ 1  1  .5 .5  1  1  1 .5 .5  1  2   1  1  4 ]
          [

             playNote ?1 ?2
          ]
  )
end

to playNote [ theNote theLength ]

  sound:start-note "TRUMPET" theNote 65

  wait theLength * quarterNoteLength

  sound:stop-note "TRUMPET" theNote

end
```

Note that the Twinkle Twinkle Little Star song has only quarter notes and half notes. A quarter note gets 1 count and a half note gets two counts. There are 36 quarter notes in the song and 6 half notes in the song. 32 plus 2 times 6 = 48 counts or 48 beats in the song.




**FACE** for the spaces, **EGBDF** for the lines of the treble clef. Every Good Boy Does Fine = E G B D F. See also Moody Blues.

**FACE** rhymes with SPACE. There are 4 spaces within the 5 lines. F and A and C and E are the notes for the 4 spaces.