An experiment to build Turtle Geometry with basic lists and functions

Constructor and selectors

A turtle is defined simply as a list of three things - x and y coordinates and an angle. A turtle is created by a constructor and it is accessed by some selectors.

In[1]:=
```
MakeTurtle[x_, y_, angle_] := {x, y, angle}
```

In[2]:=
```
TurtleX[{x_, y_, angle_}] := x
```

In[3]:=
```
TurtleY[{x_, y_, angle_}] := y
```

In[4]:=
```
TurtleAngle[{x_, y_, angle_}] := angle
```

Tests of the constructor and selectors

Most part of turtle graphics can be drawn by just lines, even the arcs. So we need functions to get x, y out of a turtle and create a line.

In[49]:=
```
TurtleLineCoords[turtles_] := {TurtleX[#], TurtleY[#]} & /@ turtles
```

Now lets test how these functions work.

In[15]:=
```
t1 = MakeTurtle[10, 14, 30]
```

Out[15]=
```
{10, 14, 30}
```

In[16]:=
```
t2 = MakeTurtle[20, 10, 35]
```

Out[16]=
```
{20, 10, 35}
```

In[19]:=
```
{TurtleX[t2], TurtleY[t2], TurtleAngle[t2]}
```

Out[19]=
```
{20, 10, 35}
```

In[21]:=
```
{TurtleX[t1], TurtleY[t1], TurtleAngle[t1]}
```

Out[21]=
```
{10, 14, 30}
```

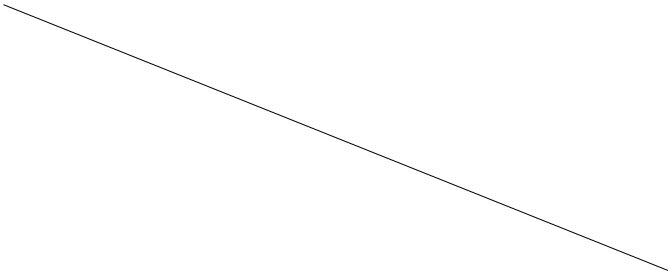
In[22]:=
```
TurtleLineCoords[{t1, t2}]
```

Out[22]=
```
{{10, 14}, {20, 10}}
```

Now lets try to draw a line between t1 and t2.

In[23]:=
```
Graphics[Line[TurtleLineCoords[{t1, t2}]]]
```

Out[23]=



Drawing primitives

Now we shall define the transformation functions which will depend less on the representation of turtle and use the above functions. First the forward function.

In[65]:=
```
TurtleFwd[turtle_, len_] :=
 MakeTurtle[TurtleX[turtle] + Sin[TurtleAngle[turtle] Degree] * len,
  TurtleY[turtle] + Cos[TurtleAngle[turtle] Degree] * len, TurtleAngle[turtle]]
```

Lets test it now.

In[66]:=
```
tf = TurtleFwd[MakeTurtle[0, 0, 0], 10]
```

Out[66]=
```
{0, 10, 0}
```

In[67]:=
```
Graphics[
 Line[TurtleLineCoords[{MakeTurtle[0, 0, 0], TurtleFwd[MakeTurtle[0, 0, 0], 10]}]]]
```

Out[67]=

Lets modify the forward function to make it a higher order function.

In[69]:=
```
TFwd[len_] := TurtleFwd[#, len] &
```

Now we will implement the other primitives.

In[71]:=
```
TurtleRight[turtle_, angle_] :=
 MakeTurtle[TurtleX[turtle], TurtleY[turtle], TurtleAngle[turtle] + angle]
```

In[72]:=
```
TRight[angle_] := TurtleRight[#, angle] &
```

In[73]:=
```
TurtleLeft[turtle_, angle_] :=
 MakeTurtle[TurtleX[turtle], TurtleY[turtle], TurtleAngle[turtle] - angle]

TLeft[angle_] := TurtleLeft[#, angle] &
```

The sequencer (a.k.a. the fold)

In[120]:=
```
TSeq[turtleFns_] :=
 Fold[Function[{trtlist, tf}, Append[trtlist, tf[Last[trtlist]]]], #, turtleFns] &
```

With the above sequencer and the higher order functions a list of primitive modifiers can be successively applied to form turtles all the way to the last step.

In[125]:=
```
TSeq[{TFwd[10], TRight[45], TFwd[20]}][{MakeTurtle[0, 0, 0]}]
```

Out[125]= $\left\{\{0, 0, 0\}, \{0, 10, 0\}, \{0, 10, 45\}, \left\{10\sqrt{2}, 10 + 10\sqrt{2}, 45\right\}\right\}$

In[126]:=
```
TurtleLineCoords[%]
```

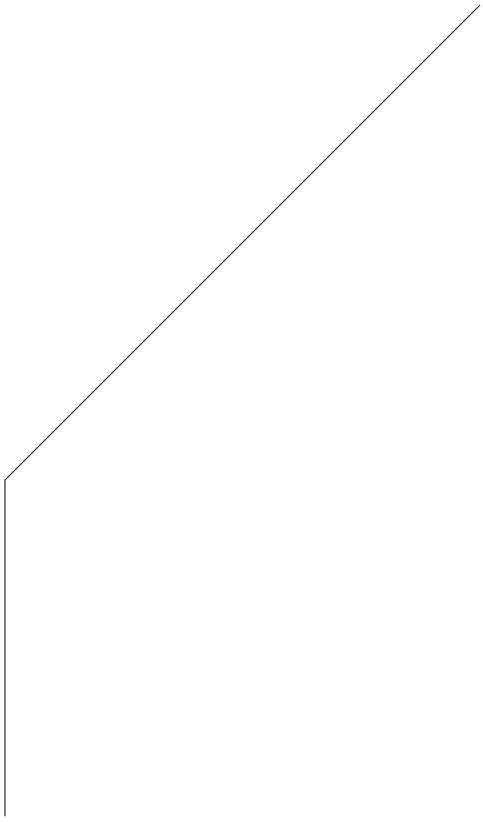
Out[126]= $\left\{\{0, 0\}, \{0, 10\}, \{0, 10\}, \left\{10\sqrt{2}, 10 + 10\sqrt{2}\right\}\right\}$

In[127]:=
```
Line[%]
```

Out[127]= $\text{Line}\left[\left\{\{0, 0\}, \{0, 10\}, \{0, 10\}, \left\{10\sqrt{2}, 10 + 10\sqrt{2}\right\}\right\}\right]$

In[128]:=
```
Graphics[%]
```

Out[128]=



Finally, we need a convenience function to begin drawing on a list of primitives like TFwd and TRight.

Turtle

In[131]:=
```
Turtle[stepslist_] :=
 Graphics[Line[TurtleLineCoords[TSeq[stepslist][{MakeTurtle[0, 0, 0]}]]]]
```

In[138]:=
```
Turtle[{TFwd[5], TLeft[45], TFwd[5], TRight[135], TFwd[15],
  TRight[135], TFwd[5], TLeft[45], TFwd[5], TRight[90], TFwd[8]}]
```

Out[138]=