Kyle Klausen

Assignment6\_2

CSD-325

06/22/25

Python revised code:

"""Forest Fire Sim, modified by Sue Sampson, based on a program by Al Sweigart A simulation of wildfires spreading in a forest. Press Ctrl-C to stop. Inspired by Nicky Case's Emoji Sim <http://ncase.me/simulating/model/> \*\* use spaces, not indentation to modify \*\* Tags: short, bext, simulation"""

Author: Kyle Klausen

Date: 6/19/25

Assignment: 6.2

import random, sys, time

try:

import bext

except ImportError:

print('This program requires the bext module, which you')

print('can install by following the instructions at')

print('<https://pypi.org/project/Bext/'>)

sys.exit()

# Set up the constants:

WIDTH = 79

HEIGHT = 22

TREE = 'A'

FIRE = '@'

EMPTY = ' '

WATER = '~' # New water feature

# Can edit these to adjust inputs for program

INITIAL\_TREE\_DENSITY = 0.20 # Amount of forest that starts with trees.

GROW\_CHANCE = 0.01 # Chance a blank space turns into a tree.

FIRE\_CHANCE = 0.01 # Chance a tree is hit by lightning & burns.

PAUSE\_LENGTH = 0.5

def main():

forest = createNewForest()

addLake(forest) # Add the lake once

bext.clear()

while True: # Main program loop.  
 displayForest(forest)  
  
 # Run a single simulation step:  
 nextForest = {'width': forest['width'],  
 'height': forest['height']}  
  
 for x in range(forest['width']):  
 for y in range(forest['height']):  
 if (x, y) in nextForest:  
 continue # Already handled  
  
 currentTile = forest[(x, y)]  
  
 if currentTile == WATER:  
 nextForest[(x, y)] = WATER # Lake stays unchanged  
 elif currentTile == EMPTY and random.random() <= GROW\_CHANCE:  
 nextForest[(x, y)] = TREE  
 elif currentTile == TREE and random.random() <= FIRE\_CHANCE:  
 nextForest[(x, y)] = FIRE  
 elif currentTile == FIRE:  
 for ix in range(-1, 2):  
 for iy in range(-1, 2):  
 neighbor = (x + ix, y + iy)  
 if forest.get(neighbor) == TREE:  
 nextForest[neighbor] = FIRE  
 nextForest[(x, y)] = EMPTY  
 else:  
 nextForest[(x, y)] = currentTile  
  
 forest = nextForest  
 time.sleep(PAUSE\_LENGTH)

def createNewForest():

"""Returns a dictionary for a new forest data structure."""

forest = {'width': WIDTH, 'height': HEIGHT}

for x in range(WIDTH):

for y in range(HEIGHT):

if (random.random() <= INITIAL\_TREE\_DENSITY):

forest[(x, y)] = TREE

else:

forest[(x, y)] = EMPTY

return forest

def addLake(forest):

"""Adds a lake to the center of the forest."""

centerX = WIDTH // 2

centerY = HEIGHT // 2

lakeWidth = 9

lakeHeight = 5

for x in range(centerX - lakeWidth // 2, centerX + lakeWidth // 2):  
 for y in range(centerY - lakeHeight // 2, centerY + lakeHeight // 2):  
 forest[(x, y)] = WATER

def displayForest(forest):

"""Display the forest data structure on the screen."""

bext.goto(0, 0)

for y in range(forest['height']):

for x in range(forest['width']):

tile = forest[(x, y)]

if tile == TREE:

bext.fg('green')

elif tile == FIRE:

bext.fg('red')

elif tile == WATER:

bext.fg('blue')

else: bext.fg('reset')

print(tile, end='')

print()

bext.fg('reset')

print('Grow chance: {}% '.format(GROW\_CHANCE \* 100), end='')

print('Lightning chance: {}% '.format(FIRE\_CHANCE \* 100), end='')

print('Press Ctrl-C to quit.')

if \_\_**name\_\_** == '\_\_**main\_\_**':

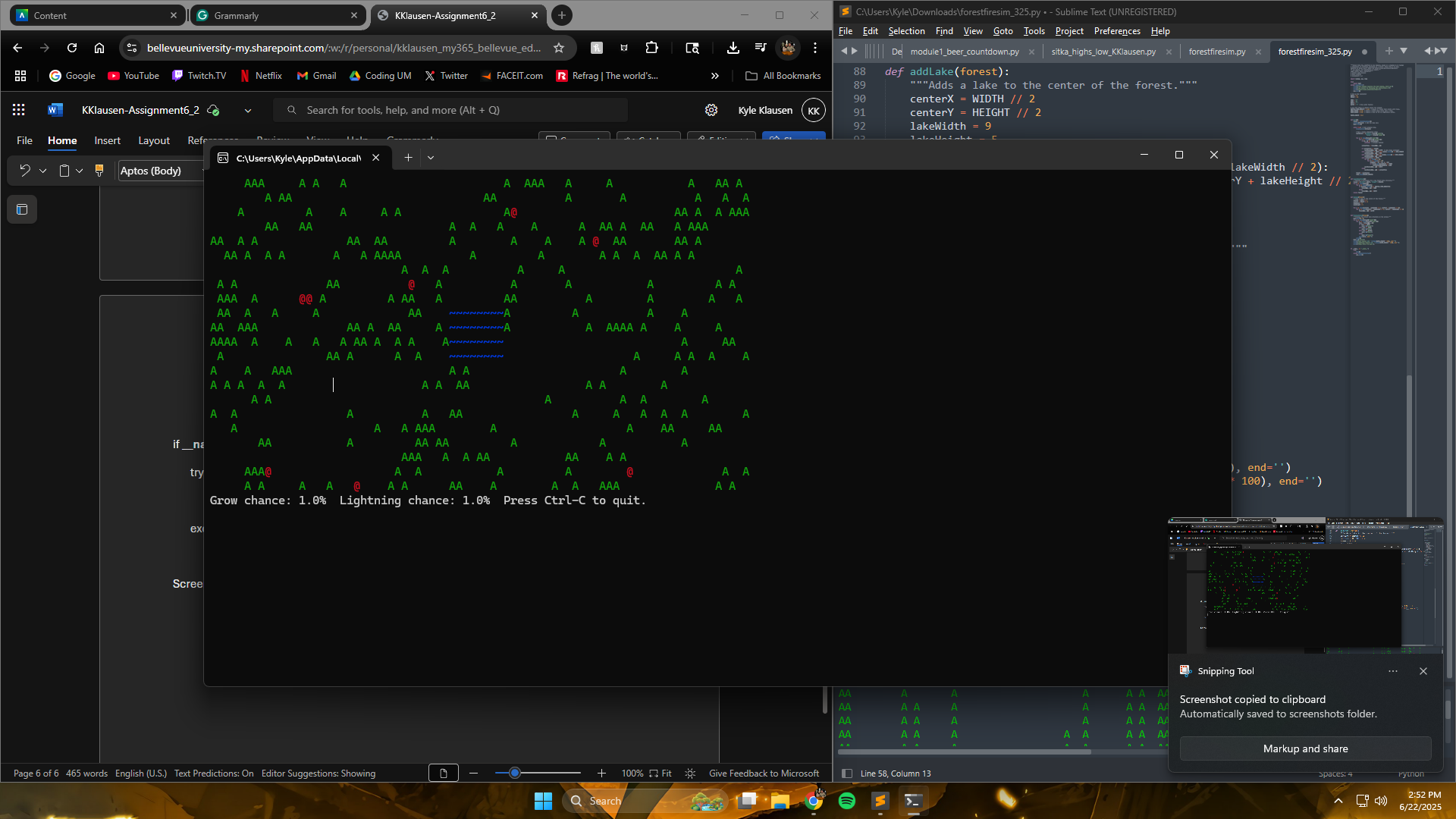
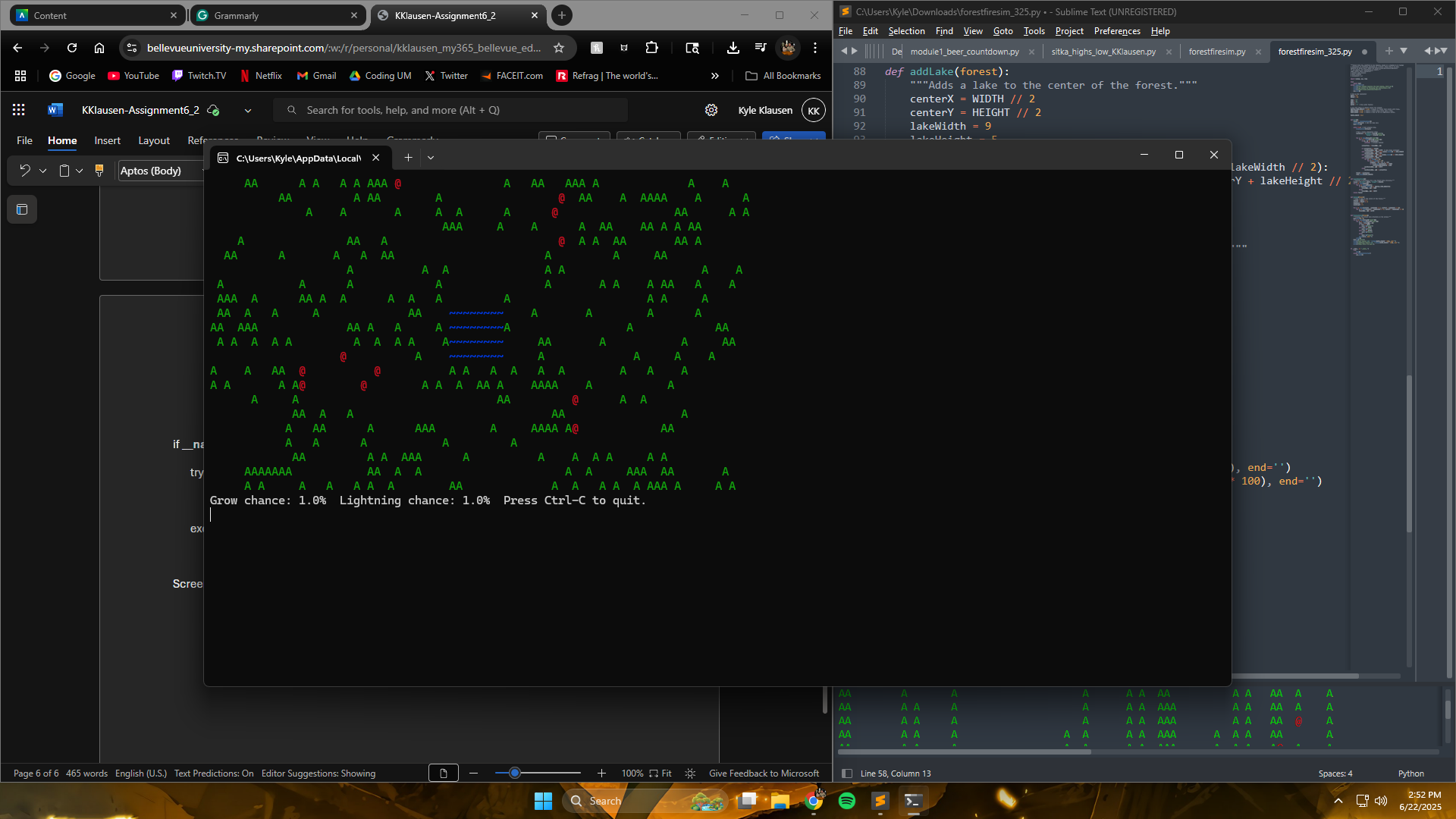
try:

main()

except KeyboardInterrupt:

sys.exit()

Screenshots of sucessful run:



Revised Flowchart to show revised code:

