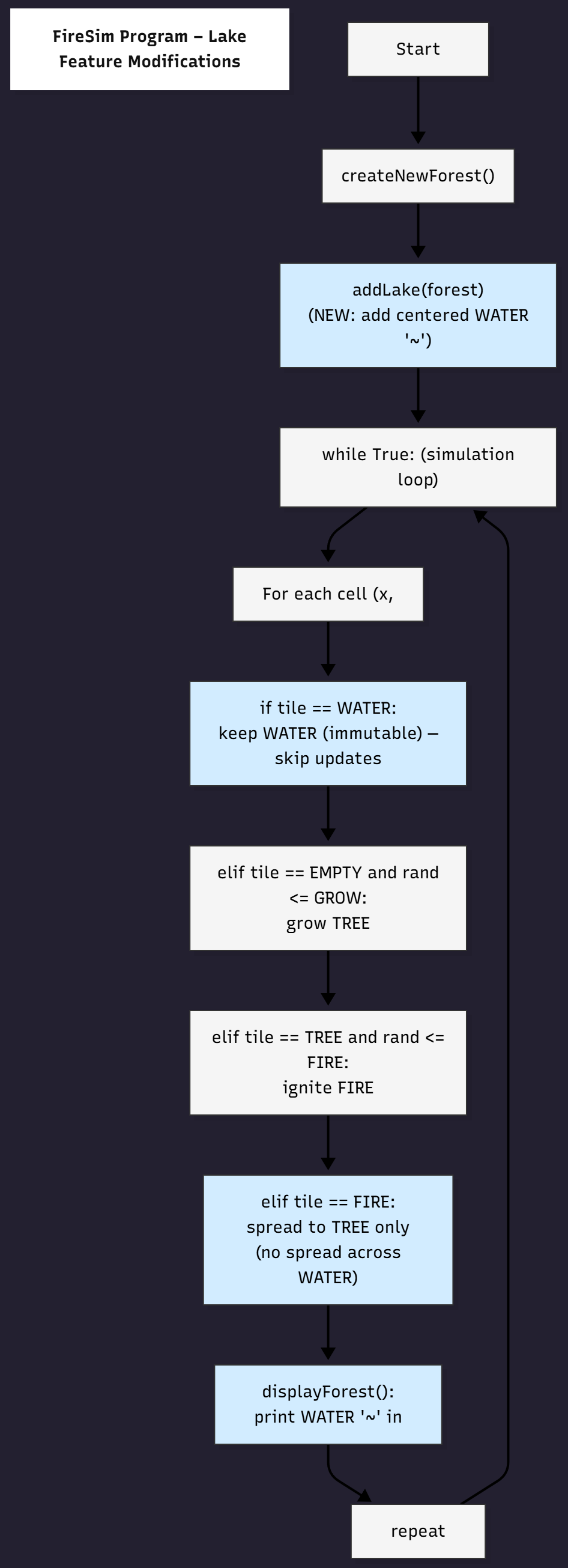
Aaron Provins, Damitrious Rivera, Michael Shalaby, Nikhil Paranjape

Module-6directory



"""Forest Fire Simulation with Random Lake Feature and File Logging  
Created for educational lab use. Includes a random central lake (firebreak),  
intro message, and saves each simulation layout to a text file.  
"""  
  
import random, sys, time, datetime  
  
try:  
 import bext  
except ImportError:  
 print('This program requires the "bext" module.')  
 print('Install it by typing this in Command Prompt:')  
 print(' pip install bext')  
 sys.exit()  
  
# Set up constants:  
WIDTH = 79  
HEIGHT = 22  
  
TREE = 'A'  
FIRE = '@'  
EMPTY = ' '  
WATER = '~' # Lake character  
  
# Simulation settings:  
INITIAL\_TREE\_DENSITY = 0.20  
GROW\_CHANCE = 0.01  
FIRE\_CHANCE = 0.01  
PAUSE\_LENGTH = 0.5  
  
  
def main():  
 # Display intro before starting simulation  
 bext.clear()  
 bext.fg('cyan')  
 print("=" \* 60)  
 print("🌲 FOREST FIRE SIMULATION 🌲".center(60))  
 print("=" \* 60)  
 print("\nThis simulation models how fire spreads through a forest.")  
 print("A blue '~' represents a lake — fire cannot cross it.")  
 print("Green 'A' are trees, red '@' are burning trees, and spaces are empty ground.")  
 print("\nPress Ctrl-C at any time to stop the simulation.")  
 print("\nGenerating random forest and lake...")  
 time.sleep(3)  
  
 forest = createNewForest()  
 bext.clear()  
 saveForestToFile(forest)  
  
 while True:  
 displayForest(forest)  
 nextForest = {'width': forest['width'], 'height': forest['height']}  
  
 for x in range(forest['width']):  
 for y in range(forest['height']):  
 current = forest[(x, y)]  
  
 # Keep the lake intact  
 if current == WATER:  
 nextForest[(x, y)] = WATER  
 continue  
  
 # Growth of new trees  
 if current == EMPTY and random.random() <= GROW\_CHANCE:  
 nextForest[(x, y)] = TREE  
 # Lightning strikes trees  
 elif current == TREE and random.random() <= FIRE\_CHANCE:  
 nextForest[(x, y)] = FIRE  
 # Burning trees spread fire  
 elif current == FIRE:  
 for ix in range(-1, 2):  
 for iy in range(-1, 2):  
 neighbor = forest.get((x + ix, y + iy))  
 if neighbor == TREE:  
 nextForest[(x + ix, y + iy)] = FIRE  
 nextForest[(x, y)] = EMPTY  
 else:  
 nextForest[(x, y)] = current  
  
 forest = nextForest  
 time.sleep(PAUSE\_LENGTH)  
  
  
def createNewForest():  
 """Return a new forest data structure with a random lake."""  
 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  
  
 # Randomize lake size  
 lake\_width = random.randint(WIDTH // 8, WIDTH // 4)  
 lake\_height = random.randint(HEIGHT // 5, HEIGHT // 3)  
  
 # Center position for lake  
 start\_x = WIDTH // 2 - lake\_width // 2  
 start\_y = HEIGHT // 2 - lake\_height // 2  
  
 # Create the lake area  
 for x in range(start\_x, start\_x + lake\_width):  
 for y in range(start\_y, start\_y + lake\_height):  
 if random.random() > 0.15: # adds rough edges  
 forest[(x, y)] = WATER  
  
 return forest  
  
  
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(f'Grow chance: {GROW\_CHANCE \* 100:.1f}% ', end='')  
 print(f'Lightning chance: {FIRE\_CHANCE \* 100:.1f}% ', end='')  
 print('Press Ctrl-C to quit.')  
  
  
def saveForestToFile(forest):  
 """Save the starting layout of the forest to a text file."""  
 filename = f"forest\_log\_{datetime.datetime.now().strftime('%Y%m%d\_%H%M%S')}.txt"  
 with open(filename, "w") as f:  
 for y in range(forest['height']):  
 line = ''.join(forest[(x, y)] for x in range(forest['width']))  
 f.write(line + "\n")  
 print(f"\nInitial forest layout saved to: {filename}")  
 time.sleep(2)  
  
  
# Run the simulation  
if \_\_name\_\_ == '\_\_main\_\_':  
 try:  
 main()  
 except KeyboardInterrupt:  
 bext.fg('reset')  
 print("\nSimulation ended. Stay safe out there. 🌳🔥")  
 sys.exit()

A screenshot of a computer

AI-generated content may be incorrect.