MIDDLE EAST TECHNICAL UNIVERSITY, NORTHERN CYPRUS CAMPUS
CNG315 Algorithms

Date handed out: 5 November 2020, Thursday
Date submission due: 19 November 2020, Thursday 23:59

Assignment 1: A Simple Eye-tracking Data Analyser

This assignment aims to help you practice the ***heap data structure*** and ***heap sort algorithm***. Your main task in this assignment is to develop a simple eye-tracking data analyser using C programming language.

Overview

Eye tracking is commonly used to understand how people interact with visual stimuli. When people interact with web pages, their eyes become relatively stable at certain points referred to as fixations. The series of these fixations represent their scan paths. Eye-tracking data analysis is typically conducted based on the areas of interest (AOIs) of visual stimuli, specifically which AOIs are commonly used and in which order. Figure 1 shows an example of a person's eye-tracking data on one page of the WordPress.com website, which is segmented into its AOIs, where the fixations are represented as circles and the longer fixations are represented with the larger circles.

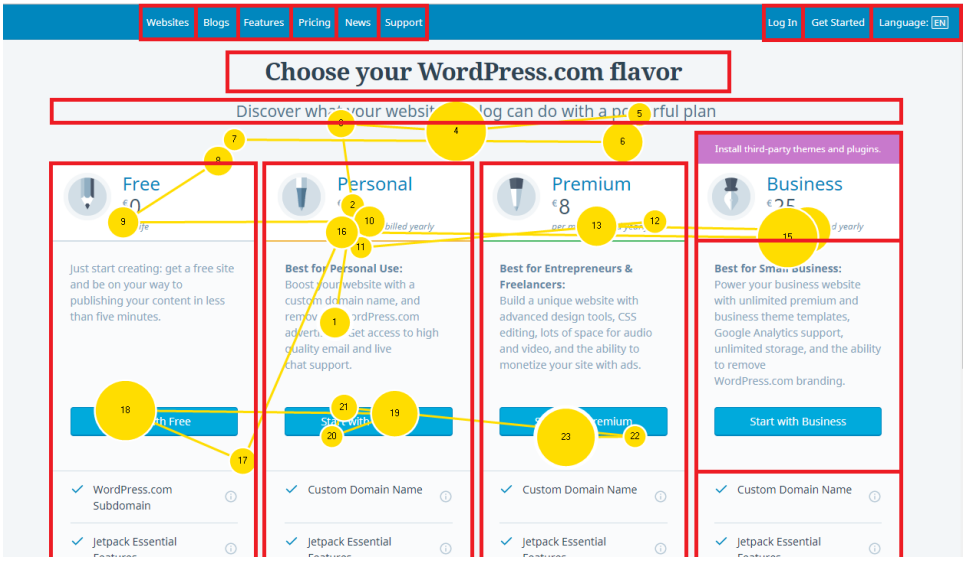

Figure 1. Eye-tracking data of a person on one page of the WordPress.com website.

In this assignment, you will need to develop a simple eye-tracking data analyser which supports the following functionalities:

1. Take a series of fixations of multiple people on a particular page (id, x, y, the duration for each fixation)
2. Take the details of the AOIs of the page (name, top-left-x, width, top-left-y, and height)
3. Compute the total number of fixations (i.e. fixation count) and the total duration of fixations (i.e. dwell time) for each AOI.
4. Sort the AOIs based on either fixation count or dwell time
5. Print the sorted AOIs with their fixation counts and dwell time

Process Model

The process model of this program is as follows (see Figure 2):

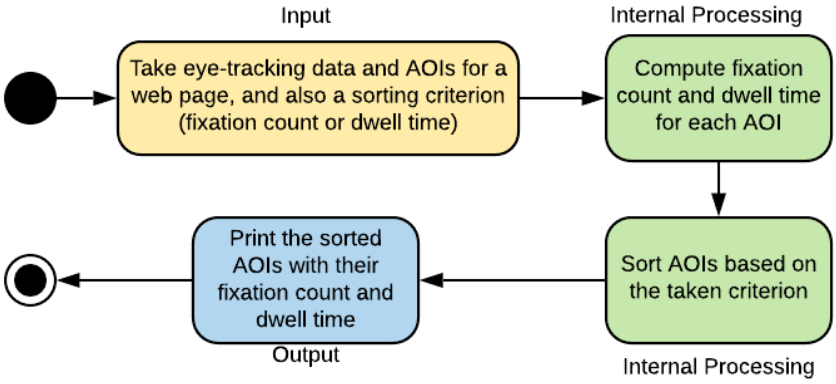

Figure 2. Process model

1. Input

This program will have three input:

- **Sorting criterion:** Sorting criterion can be either (1) fixation count or (2) dwell time.
- **AOIs:** There is a file to keep the details of AOIs in the following order separated by a space: AOI-name, top-left-x, width, top-left-y and height. AOI-name is a single character and other values are integer values. Each line represents a different AOI. You should not make any assumptions about the number of AOIs. Please note that the coordinate of the top-left point of a web page is (0, 0), and it works as shown in Figure 3:

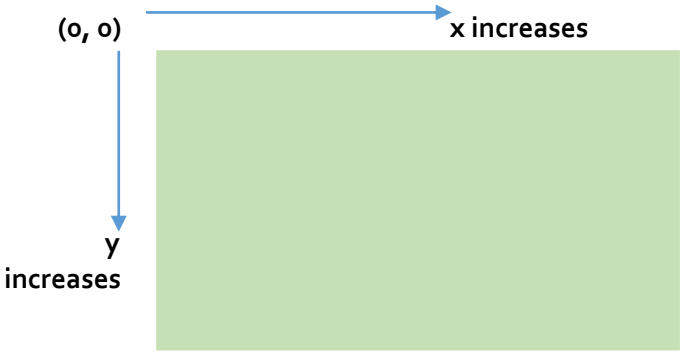

Figure 3. Coordination system

- **Eye-tracking data files:** There is a separate eye-tracking data file for each person to keep the details of his/her fixations in the following order separated by a space: fixation index, x, y, and fixation duration. All of these values are integers. Each line represents a different fixation. You should not make any assumptions about the number of fixations in each file. These files are located in a folder and named as 1.txt, 2.txt, 3.txt, etc. where these numbers represent the identifier of the people.

This program takes these inputs them as command-line arguments as follows:

```
eyeAnalyser sortingCriteria AOIsFile DataFolder ParticipantCount
```

The name of the program	The sorting criterion which is 1 for fixation count and 2 for dwell time	The path of the AOI file	The path of the folder of the eye-tracking data files	The number of people that we have data files

2. Internal Processing

Once the inputs are received from the user, the program will start with the preparation of the structure to keep the details of the AOIs. It will create a data structure for each AOI to keep its name, top-left-x, width, top-left-y, height, fixation count, and dwell time, and keep them in an unsorted array. An example of this array for six AOIs is illustrated below in Figure 4.

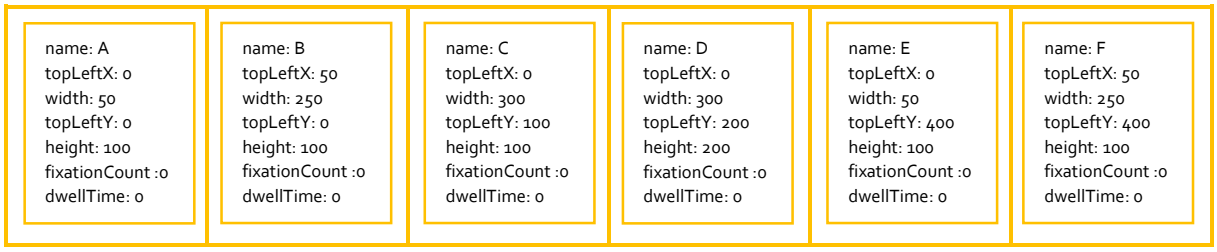

Figure 4. An example of AOI array for six AOIs

Once the AOI structure is created, then the following functions that you will implement as part of this assignment are called in the following order for internal processing:

- **computeFixationCount**: It takes the AOI array, the path of the data folder, and the number of people. It will then read the data files, find the corresponding AOI for each fixation, and update its fixation count.
- **computeDwellTime**: It takes the AOI array, the path of the data folder, and the number of people. It will then read the data files, find the corresponding AOI for each fixation, and update its dwell time.
- **heapSort**: It takes the AOI array and the sorting criteria which is either 1 (fixation count) or 2 (dwell time). It firstly applies the Build Heap algorithm to create a max-heap and then sorts the AOIs based on the heap sort algorithm.

3. Output

This program will print the sorted AOIs with their fixation count and dwell time. An example of six AOIs is given below, sorted according to their fixation counts.

AOI	Fixation-count	Dwell-time
C	56	8452 ms
D	43	8645 ms
B	42	5012 ms
A	20	3034 ms
F	5	987 ms
E	3	989 ms

For this operation, you will need to implement a function called **printAOIs** that will take the sorted AOI array and print AOIs in the format demonstrated above.

Incremental Development

You are expected to follow an incremental development approach. Each time you complete a function or module, you are expected to test it to make sure that it works properly. You can find a sample set of data on ODTUCLASS to test your program.

Professionalism and Ethics

You are expected to complete the assignments on your own. Sharing your work with others, uploading the assignment to online websites to seek solutions, and/or presenting someone else's work as your own work will be considered as cheating.

Rules

- You need to write your program by using **C programming.**
- You need to name your file with **your student id**, e.g. 1234567.c, and submit it to ODTUCLASS.
- **Code quality, modularity, efficiency, maintainability, and appropriate comments** will be part of the grading.

Grading Policy

The assignment will be graded as follows:

Grading Item	**Mark (out of 100)**
Input	10
AOI array	10
computeFixationCount	10
computeDwellTime	10
Heapify	15
buildHeap	10
heapSort	15
printAOIs	10
Main Function	10