**CSC 545/645 Computer Speech, Music and Images**

**Exercise No. 10c, Week 13, Due April 18, 2021**

**Learn to read srt subtitle files in Processing**

**Goal**

Learn to extract information from Subrip text (srt) files

**Background**

Subrip is a free program that extracts subtitles and timings from video (Wikipedia). srt is the file format used by Subrip and is widely used as a standard for storing subtitles. For more information regarding the format, see <https://en.wikipedia.org/wiki/SubRip>.

**Procedure**

Write a Processing program to read and extract information from srt files. Start with the skeleton code download from Blackboard. You are given a setup() function that creates a SubtitlePlayer object. This object will read an srt file and print the subtitles, with their start and end times, to the console. Your goal is to complete the code for SubtitlePlayer.

First, open the srt file stored in the data folder. The format is simple and you can likely figure it out from examining the file. The two most important things to notice are that subtitles are separated by a blank line and each subtitle includes a string specifying the start time and end time, separated by an arrow (“-->”). Each subtitle is preceded by a string holding one integer. I believe that can indicate the sequence number of the subtitle but it’s not used in this file.

SubtitlePlayer includes some strings that will help you parse times – arrow is used to split the start time from the end time and the characters in tokens are used to split a time into hours, minutes, seconds, and milliseconds.

The code you are given assumes that the SubtitlePlayer constructor will load the strings from the file into an array of strings – an instance variable names *lines[].* If you choose to use my design, complete the constructor as follows:

1. Use a while loop to read each line in the file. For each line:
2. If the line contains an arrow
   1. Create a new *subtitle.* I suggest you use a StringList because you can add to it, line by line.
   2. Append the time line to the subtitle and increment the line index.
   3. While the length of the lines is > 0, append the line to the subtitle.
   4. Use a separate function to print the subtitle (pass the subtitle StringList to that function).
   5. Print a blank line to separate the subtitles.
   6. Increment the pointer to go to the next line
3. In the print subtitle function:
   1. Get the time line and print the start time followed by the end time; use Java’s *substring* function to separate the start time from the end time. Use the string *­splitTokens* method to separate the elements of the time into an array of strings. Use parseTime() (described below) to parse the times, based on the array of elements.
   2. Print each line of the subtitle.
4. Write the parseTime() function. parseTime(String timeArr) takes an array of strings; timeArr[0] is hours, timeArr[1] is minutes, timeArr[2] is seconds, and parseTime[3] is milliseconds. Return the time in milliseconds as an integer.

**Deliverables**

Submit your pde file on Blackboard by the due date.