EECS 22: Advanced C Programming
Assignment 5

Tim Schmidt
schmidtt@uci.edu

11/14/2017

Outlines

- General Information
- Double Linked List Traversal
- Command-Line Argument Parsing
- Load/Save Dependencies
- Spotlight
- Watermark
- Important

Assignment 5

- A command line driven movie program [100 pts]
- Deadline: 2017/12/06, Wednesday, 6:00 pm
- Goal
  - learn to process command line parameters
  - learn to create and operate on doubly linked lists
- Extra Credit
  - Zoom operation on movie [10 pts]

Double Linked List Traversal

`SaveMovie()` from MovieLab.c:

```
218     count = 0;
219     curr = movie->Frames->First;
220     while (curr != NULL) {
221         for (y = 0; y < curr->YUVImage->Height; y++) {
222             for (x = 0; x < curr->YUVImage->Width; x++) {
223                 fputc(GetPixelY(curr->YUVImage, x, y), file);
224             }
225         }
226
227         for (y = 0; y < curr->YUVImage->Height; y += 2) {
228             for (x = 0; x < curr->YUVImage->Width; x += 2) {
229                 fputc(GetPixelU(curr->YUVImage, x, y), file);
230             }
231         }
232
233         for (y = 0; y < curr->YUVImage->Height; y += 2) {
234             for (x = 0; x < curr->YUVImage->Width; x += 2) {
235                 fputc(GetPixelV(curr->YUVImage, x, y), file);
236             }
237         }
238
239         curr = curr->Next;
240         count++;
241     }
```

Initialization

Stop at the end of the linked list

Operations on the current entry

Move to the next entry

Double Linked List Traversal

- Variations:
  - What if you need to add an entry at the end?
    - `AppendRGBImage`
    - `AppendYUVImage`
  - What if you need to delete some entries?
    - `CropImageList`: at either end
    - `FastImageList`: in the middle
  - What if you need to reverse the order?
    - `ReverseImageList`
- Hints:
  - Use extra pointers (in addition to `curr`)
  - Use additional variables to keep track of the position

Command-Line Argument Parsing

`Main()` from MovieLab.c:

```
while (x < argc) {
    if (strcmp(argv[x], "-i") == 0) {
        if (x < argc - 1) {
            fin = argv[x + 1];
        } else {
            printf("Missing argument for the input file name!\n");
            return 5;
        }
        x += 2;
        continue;
    }
    /* to be implemented */
    x++;
}
```

Why? #1

Why? #2

Why? #3

Why? #4

Why? #5

Why? #6

Command-Line Argument Parsing

- Answers:
  1. The "`-i`" option always requires a second argument for the input file name after it.
  2. Save the pointer for the input string
  3. There is not input file and we report this error
  4. Increment by two because we processed to parameters
  5. Completed this set of arguments and we skip all other options
  6. Skip any unrecognized option.

Load/Save Dependencies

LoadMovie (Implement by students)

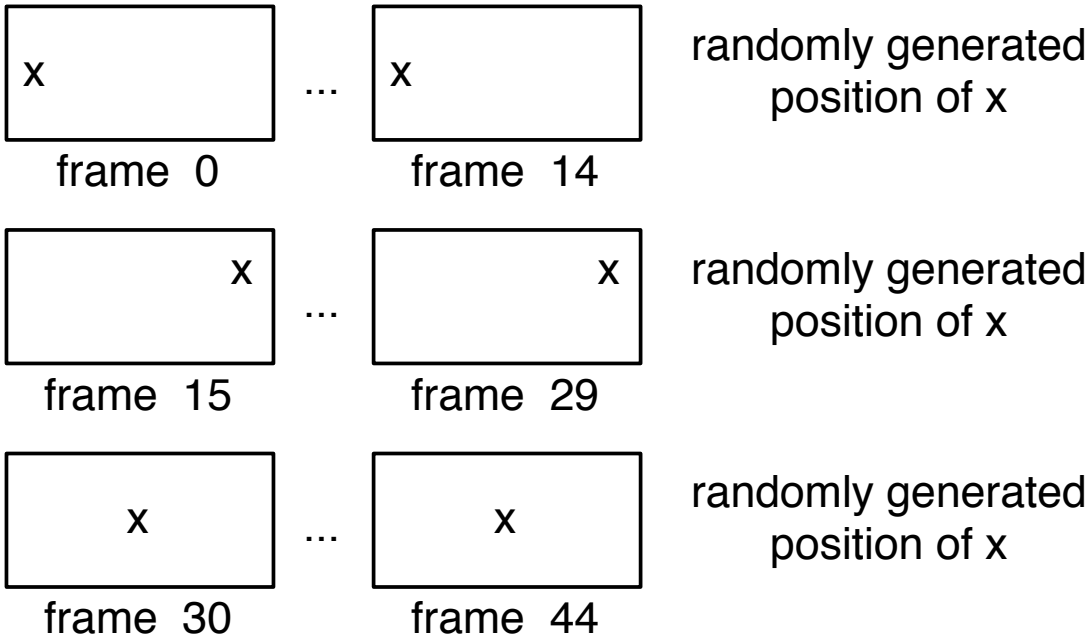
- LoadOneFrame (Implement by TAs)
  - CreateYUVImage (Implement by students)

SaveMovie (Implement by students)

- SaveOneFrame (Implement by TAs)

Watermark

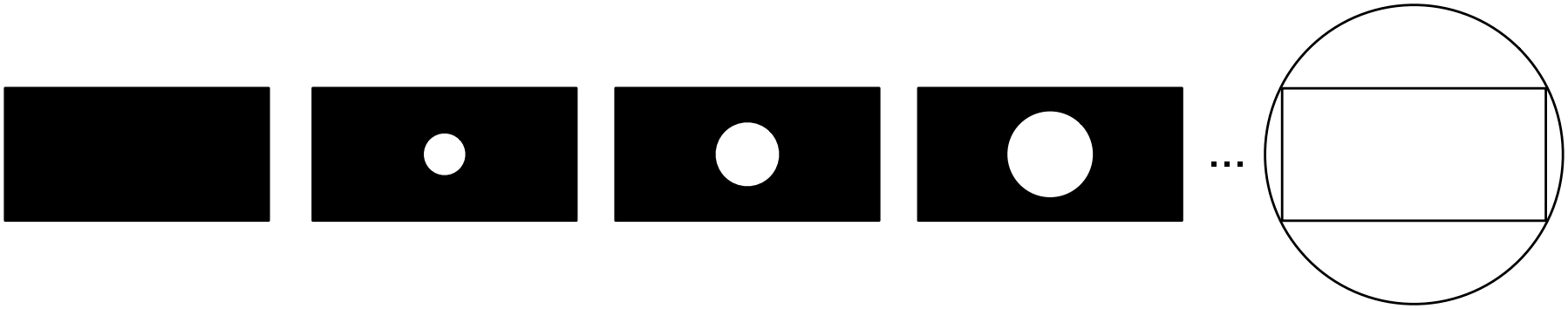
- Adds a watermark to each frame
  - Watermark position is randomly generated
  - Watermark position changes every 15 frames
- Example



Spotlight

- Adds a fade in and fade out spotlight to the movie
  - First 40 frames fade in
  - Last 20 frames fade out

- Fade in for a movie with 480x270 pixel



Frame: 0
Radius: 0

Frame: 1
Radius: 7.06...

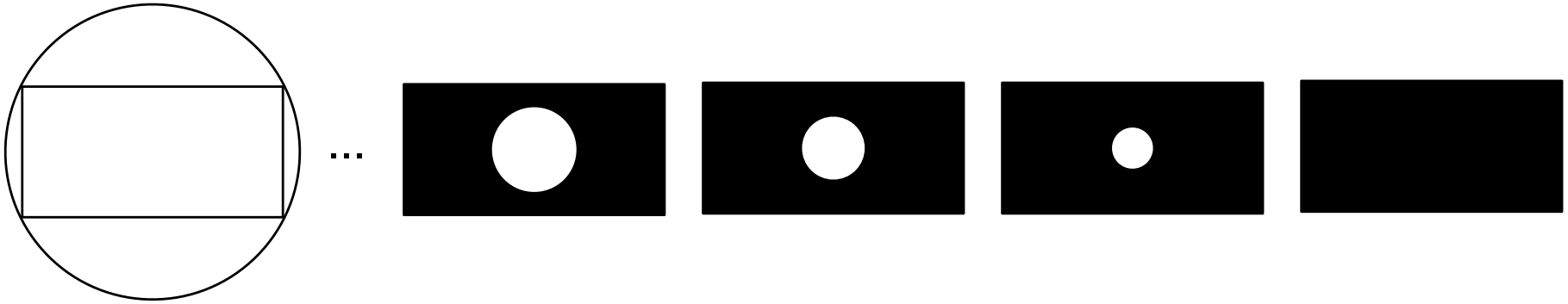
Frame: 2
Radius: 14.12...

Frame: 3
Radius: 21.18...

Frame: 39
Radius: 275.36...

Spotlight

- Fade out for a movie with 480x270 pixel



Frame: 126
Radius: 275.36...

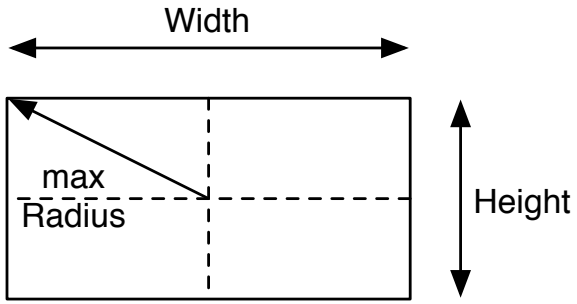
Frame: 142
Radius: 43.47...

Frame: 143
Radius: 28.98...

Frame: 144
Radius: 14.49...

Frame: 145
Radius: 0

- Computing the maximal radius



Important

- Example movies are provided
  - `~eecs22/public/hw5/demo/NAME.yuv`