

검색

千 Line 으로 비디오플레이어 만들기 2

멀티미디어 : 2009/02/14 22:44

[Google 크롬 다운로드](#)

타이핑에는 시간을 적게 웹 브라우저에는 더 많은 시간을

www.google.co.kr/chrome

AdChoices

[영어의 문장구조](#)

영어를 단어 뜻으로 대충 추측해서 이해하시죠? 과연 올바른 방법일까요?

www.nswenglish.com

AdChoices

ffmpeg 천줄로 비디오플레이어 만들기

<http://www.dranger.com/ffmpeg/ffmpeg.html>

<http://hybridego.net/entry/천줄로-비디오플레이어-만들기>

<http://hybridego.net/entry/千-Line-으로-비디오플레이어-만들기-1>

<http://hybridego.net/entry/千-Line-으로-비디오플레이어-만들기-2>

<http://hybridego.net/entry/千-Line-으로-비디오플레이어-만들기-3>

<http://hybridego.net/entry/千-Line-으로-비디오플레이어-만들기-4>

<http://hybridego.net/entry/千-Line-으로-비디오플레이어-만들기-5>

<http://hybridego.net/entry/千-Line으로-비디오플레이어-만들기-6>

<http://hybridego.net/entry/千-Line으로-비디오플레이어-만들기-7>

<http://hybridego.net/entry/千-Line으로-비디오플레이어-만들기-8>

<http://hybridego.net/entry/千-Line으로-비디오플레이어-만들기-마무리>

별로 중요하지 않은 부분중 생략된 부분이 있고 의역이 많이 있습니다.

군데군데 틀린 번역이 있을수 있습니다. (영어가 후달려서)

하지만 소스코드 설명 부분은 틀리지 않도록 노력했습니다.

위의 원문을 번역한 것입니다.

잘못된 부분이 있을지도 모르겠습니다.

혹시 잘못된 부분을 발견하시면 댓글로 알려주시면 감사하겠습니다.

Tutorial 02: Outputting to the Screen

 tutorial02.c

SDL and Video

우리는 스크린에 그림을 그리기 위해서 SDL 을 사용할 것입니다.

SDL 은 Simple Direct Layer 표준이고 멀티미디어, Cross-Platform 등을 위한 훌륭한 라이브러리 입니다.

SDL the official website 에서 개발 패키지를 다운 받으실 수 있습니다.

이번 Tutorial을 컴파일 하기 위해서는 SDL 라이브러리가 꼭 있어야 하니까 설치해 두셔야 합니다.

SDL은 스크린에 그림을 그리기 위한 많은 메소드를 갖고 있고, 이것으로 비디오를 출력할 수도 있습니다. 이것을 YUV 오버레이 라고 부릅니다.

YUV (technically not YUV but YCbCr) 는 RGB같은 RAW 이미지 데이터를 저장하는 방법입니다.

개략적으로 말하면 Y는 밝기, U와 V는 색상을 나타냅니다. (이것은 RGB보다 좀더 복잡한데 몇몇 색깔정보는 버려지고, Y요소 2개당 U,V 요소는 1개씩 들어가기 때문입니다.) SDL의 YUV 오버레이는 YUV 데이터의 RAW 배열을 갖고

있고 이것을 디스플레이 합니다. 4가지 종류의 YUV포맷이 사용 가능하지만 YV12가 가장 빠릅니다. YUV420P이라는 다른 YUV포맷이 있는데 이것은 YV12에서 U랑 V가 바뀐것 빼고는 똑같습니다.
Subsample 된 420 이라는 것은 4:2:0의 비율을 나타내는데 Y성분이 4개면 U,V성분은 1개가 있다는 뜻입니다.
이런 방법은 bandwidth를 저장하는데 매우 효율적입니다. 사람의 눈은 이런 변환을 잘 구분하지 못하기 때문이지요.
이름에 있는 P는 포맷이 planar 라는 뜻입니다. -이것은 YUV요소가 연속적 배열이라는 의미 입니다.
ffmpeg는 이미지를 YUV420P로 변환할 수 있고, 그것을 다른 비디오 포맷으로 쉽게 변환 할 수 있습니다.

이제 우리는 Tutorial1의 SaveFrame() 함수를 대체하고, 출력을 우리 화면에 해보도록 합니다.

아! 그전에 SDL Library를 어떻게 사용해야 하는지를 먼저 봐야 겠군요

일단 라이브러리를 Include하고 SDL을 초기화해볼까요?

```
view plain copy to clipboard print ?
01. #include <SDL.h>
02. #include <SDL_thread.h>
03.
04. if(SDL_Init(SDL_INIT_VIDEO | SDL_INIT_AUDIO | SDL_INIT_TIMER)) {
05.     fprintf(stderr, "Could not initialize SDL - %s\n", SDL_GetError());
06.     exit(1);
07. }
```

SDL_Init()는 우리가 SDL을 사용하려고 할때 꼭 써야하는 함수 입니다.

SDL_GetError()는 디버깅 함수이구요

Creating a Display

이젠 이미지를 디스플레이 할곳이 필요한데 이 SDL에서는 이 장소를 Surface라고 부릅니다.

```
view plain copy to clipboard print ?
01. SDL_Surface *screen;
02.
03. screen = SDL_SetVideoMode(pCodecCtx->width, pCodecCtx->height, 0, 0);
04. if(!screen) {
05.     fprintf(stderr, "SDL: could not set video mode - exiting\n");
06.     exit(1);
07. }
```

화면의 가로, 세로를 설정합니다. 그 다음 옵션은 스크린의 depth 인데요 0으로 하면 현재 디스플레이와 같은 depth로 설정하는 것입니다.

이번엔 우리가 만든 screen에 YUV overlay를 만들어 video를 올릴것입니다.

```
view plain copy to clipboard print ?
01. SDL_Overlay *bmp;
02.
03. bmp = SDL_CreateYUVOverlay(pCodecCtx->width, pCodecCtx->height,
04.                             SDL_YV12_OVERLAY, screen);
```

가로, 세로, YV12를, screen 에~

Displaying the Image

이젠 이미지를 디스플레이 하기만 하면 됩니다. 마지막 프레임까지 쭉~

우리는 RGB프레임을 위해 갖고 있던것들을 제거할 수 있습니다. 그리고 SaveFrame()과 디스플레이 코드를 바꿀것입니다.

이미지를 디스플레이 하기 위해서 AVPicture struct 를 만들고 TUV에 쓸 데이터 포인터와 라인사이즈를 설정합니다.

```
view plain copy to clipboard print ?
01.  if(frameFinished) {
02.      SDL_LockYUVOverlay(bmp);
03.
04.      AVPicture pict;
05.      pict.data[0] = bmp->pixels[0];
06.      pict.data[1] = bmp->pixels[2];
07.      pict.data[2] = bmp->pixels[1];
08.
09.      pict.linesize[0] = bmp->pitches[0];
10.      pict.linesize[1] = bmp->pitches[2];
11.      pict.linesize[2] = bmp->pitches[1];
12.
13.
14.      img_convert(&pict, PIX_FMT_YUV420P,
15.                  (AVPicture *)pFrame, pCodecCtx->pix_fmt,
16.                  pCodecCtx->width, pCodecCtx->height);
17.
18.      SDL_UnlockYUVOverlay(bmp);
19.  }
```

SDL_LockYUVOverlay(), img_convert(), SDL_UnlockYUVOverlay()

overlay를 써야하기 때문에 일단 lock을 걸어줍니다. lock을 거는것을 습관하 하세요 그래야 나중에 뭇탈이 없습니다.

전에 봤던 AVPicture struct는 하나의 data pointer를 갖고 있는데 이것은 4개의 pointer의 array 입니다.

YUV420P를 다루기 때문에 우리는 3개의 channel을 갖고 있고 그래서 4개중에 3개만 세팅 합니다.

다른 포맷들은 아마 Alpha channel까지 해서 네개의 포인터를 갖고 있을지도 모릅니다.

linesize는 사운드 같은것 입니다.

YUV overlay에 있는 비슷한 구조들은 다양한 pixel과 pitch 입니다. ("pitches" 는 주어진 데이터 라인의 넓이를 알기 위해 사용되는 SDL용어 입니다.)

pict에 쓸때 overlay에 pict.data의 세개 배열을 지정한다. 우리는 사실 overlay에 기록하는 것입니다.

비슷하게, overlay로부터 직접 linesize 정보를 얻기도 합니다. 바꾼 포맷을 PIX_FMT_YUV420P로 바꾸고, 방금 전처럼 img_convert 를 사용합니다.

Drawing the Image

이제 그러면 됩니다. 이 함수에는 비디오의 가로, 세로 사이즈를 지정하면 크기를 변환해준다. SDL은 graphic processor를 이용해 빠른 Scaling 을 해줍니다.

```
view plain copy to clipboard print ?
01.  SDL_Rect rect;
02.
03.  if(frameFinished) {
04.
05.
06.      img_convert(&pict, PIX_FMT_YUV420P,
07.                  (AVPicture *)pFrame, pCodecCtx->pix_fmt,
08.                  pCodecCtx->width, pCodecCtx->height);
09.
10.      SDL_UnlockYUVOverlay(bmp);
11.      rect.x = 0;
12.      rect.y = 0;
```

```

13.     rect.w = pCodecCtx->width;
14.     rect.h = pCodecCtx->height;
15.     SDL_DisplayYUVOverlay(bmp, &rect);
16. }

```

Now our video is displayed!

이번에는 SDL의 다른 기능을 살펴봅시다.

SDL은 event system 입니다. 여러분이 버튼을 누르거나 마우스를 움직이거나 어떤 Signal을 줄때 event가 일어납니다.

여러분의 프로그램에서 이 event들을 감지할 수 있습니다. 그리고 여러분의 프로그램에서 event를 발생시킬 수도 있습니다.

이런 기능은 SDL로 Multithread 프로그램을 돌릴때 유용하게 쓸수 있습니다.

지금 이 프로그램에서는 packet processing이 끝나자마자 이벤트를 감시할 것입니다.

SDL_QUIT 이벤트를 이용해서 우리 프로그램을 종료 하도록 합시다.

```

view plain copy to clipboard print ?
01.  SDL_Event      event;
02.
03.      av_free_packet(&packet);
04.      SDL_PollEvent(&event);
05.      switch(event.type) {
06.      case SDL_QUIT:
07.          SDL_Quit();
08.          exit(0);
09.          break;
10.      default:
11.          break;
12.      }

```

이제 컴파일을 해봅시다.

여러분이 리눅스를 쓰신다면 다음이 SDL lib 을 사용한 가장 좋은 컴파일 방법입니다.

```

gcc -o tutorial02 tutorial02.c -lavutil -lavformat -lavcodec -lz -lm `
`sdl-config --cflags --libs`

```

sdl-config는 gcc에다가 SDL libraries 를 include 할수 있는 적당한 flag를 출력해 줍니다.

여러분이 만약 여러분 시스템의 컴파일을 위해 약간 다른 설정을 해야 한다면 SDL Documentation을 참고 하세요.

자! 이제 컴파일을 하고 실행을 시켜 봅시다.

어떤일이 일어나나요? 비디오가 미친거 같죠? ㅋㅋ 사실 우리는 비디오 파일에서 추출할 수 있는 한 최대속도로 비디오를 디스플레이 합니다. 우리는 지금 언제 비디오를 디스플레이 해야하는지를 coding 하지 않았어요. sync가 전혀 안되고 있습니다.

우리는 video syncing을 해야하고요.... 아차! 일단 Sound 플레이를 해야겠군요!



千 Line 으로 비디오플레이어 만들기 2 조회수 704

- 9 아이유 열애설 상대 누구? '주변사람이 만류할 정...
- 12 착한남자, 가장 잔인했던 1초 문채원의 키스가 두...

크리에이티브 커먼즈 라이선스



이 저작물은 크리에이티브 커먼즈 코리아 저작자표시-비영리-변경금지 2.0 대한민국 라이선스에 따라 이용하실 수 있습니다.

'멀티미디어' 카테고리의 다른 글

千 Line 으로 비디오플레이어 만들기 3 (0)	2009/02/14
<u>千 Line 으로 비디오플레이어 만들기 2 (0)</u>	2009/02/14
千 Line 으로 비디오플레이어 만들기 1 (0)	2009/02/14
千 Line 으로 비디오플레이어 만들기 서문 (0)	2009/02/04

Posted by Real_G

이전 1 ... 644 645 646 647 648 649 650 651 652 ... 1603 다음