طرح درس
کارگاه اس‌دی‌ال

موضوعات

1. نصب کردن اس‌دی‌ال
2. باز کردن و بستن یک پنجره
3. نحوه کار رندر
4. ترسیم اشکال
5. ورودی گرفتن از کاربر (ایونت هندلینگ)

نصب کردن اس‌دی‌ال

با توجه به سیستم عامل خود یکی از دستورات زیر را دنبال کنید:

Linux:

```
sudo apt install libsdl2-dev libsdl2-gfx-dev
```

macOs:

اگر برو نصب نشده:

https://brew.sh/

و سپس:

```
brew install Sdl2
brew install Sdl2Gfx
```

Windows:

در این مرحله کار خاصی لازم نیست انجام دهید. کافیست ریپوی project را از گیتهاب sut-ce-fop-97 کلون کنید.

همانطور که در کارگاه گرافیک اشاره شد، سایت sdl wiki مناسب‌ترین منبع برای یادگرفتن SDL است. اما با توجه به مطالب بسیار گسترده این سایت و برای راحتی بیشتر شما، صفحاتی از آن که در این پروژه کاربرد دارد در ادامه آورده می‌شود.

برای باز کردن پنجره

به شدت توصیه می‌شود که به لینک زیر مراجعه کرده و مثال آن را به دقت بررسی کنید.

https://wiki.libsdl.org/SDL_CreateWindow

معرفی و نحوه کار Renderer

در SDL برای ترسیم اشکال فقط می‌توانیم رنگ یک پیکسل را عوض کنیم و این فرآیند برگشت‌ناپذیر است. مثلا وقتی یک پیکسل در ابتدا به رنگ قرمز است و ما آن را آبی می‌کنیم، تنها راه برگرداندن این حرکت، دوباره قرمز کردن آن پیکسل است و دستوری وجود ندارد که بتوان رنگ آبی را برداشت. به عبارت دیگر پیکسل‌ها حافظه ندارند.

حال رندرر برای ترسیم اشکال مختلف چه می‌کند؟

ابتدا کل پنجره را با یک رنگ می‌پوشاند (تمام پیکسل‌های صفحه‌ها را به رنگی که برای پاک کردن ست شده، درمی‌آورد).

از این به بعد، هر دستور ترسیمی که به رندرر داده‌شود آن را روی یک صفحه فرضی اعمال می‌کند. این صفحه فرضی buffer نام دارد. بافر یک حافظه موقتی است و برای آن است که تغییرات جزئی‌ای که در طول ترسیم رخ می‌دهد نمایش تصویر کنونی را خراب نکند.

حال فرض می‌کنیم ترسیمات برنامه تمام‌شده‌است و وقت آن است که آن‌ها را نمایش دهیم. رندرر با دستور SDL_RenderPresent تمام بافر را روی پنجره‌ای که مشخص شده، کپی می‌کند.

چرخه زیر در هر ثانیه بسته به FPS یا FramePerSecond بارها انجام می‌شود:

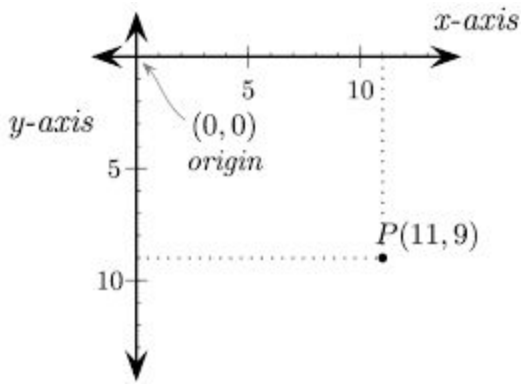
```
SDL_SetRenderDrawColor( r, g, b, a );
SDL_RenderClear( renderer );
// drawing here
SDL_RenderPresent( reneder );
SDL_Delay( 1000 / FPS );
```

نقاشی

سه رنگ اصلی در گرافیک کامپیوتری قرمز، آبی، سبز هستند. به علاوه میزان شفافیت یک رنگ با alpha یا a مشخص میشود.

بنابرین موقع نقاشی در کامپیوتر برای مشخص کردن هر رنگی به چهار عدد r, g, b, a احتیاج داریم که هرکدام بین ۰ تا ۲۵۵ تغییر میکنند.

همچنین مختصات در گرافیک کامپیوتری به این صورت است:



همچنین زاویه در کتابخانه اس‌دی‌ال دقیقا برعکس زاویه مثلثاتی و به درجه است.

تمامی توابع ترسیم‌کننده اس‌دی‌ال در فایل SDL2_gfxPrimitives.h موجودند. لینکی به این فایل در زیر موجود است.

http://www.ferzkopp.net/Software/SDL2_gfx/Docs/html/_s_d_l2__gfx_primitives_8h.html

پیشنهاد می‌شود تابع‌های مختلف را ببینید و کار کردن با هر یک را (برای کشیدن دایره و مستطیل و خط و ...) بیاموزید.

هندل کردن ایونت‌ها

ایونت چیست؟ هر اتفاق قابل فهم برای کامپیوتر یک ایونت است. مثل فشرده شدن، تایپ شدن یا رها شدن یک کلید از کیبرد یا کلیک یک دکمه از موس، یا بسته، مینیمایز، مکسیمایز یا حتی جابجا کردن پنجره باز روی صفحه.

سیستم ایونت‌هندلینگ در اس‌دی‌ال به این صورت است:

یک صف داریم که هر ایونتی که رخ می‌دهد در انتهای این صف قرار می‌گیرد و اس‌دی‌ال به ما اجازه می‌دهد هربار یک ایونت از سر صف برداریم.

```
SDL_Event e;
SDL_PollEvent(&e);
```

تابع بالا اگر ایونتی موجود باشد یک برمیگرداند و آن ایونت را در e می‌ریزد. و اگر ایونتی موجود نباشد، صفر برمیگرداند و به محتوای e کاری ندارد.

ایونت‌ها انواع مختلفی دارند اما در این داک ما فقط به ایونت‌های مربوط به صفحه کلید و پنجره می‌پردازیم.

ایونت‌های صفحه کلید

ایونت‌های صفحه کلید دارای یک keycode هستند که مشخص می‌کند ایونت مربوط به چه کلیدی است.

برای دیدن کی‌کدهای مختلف این صفحه را ببینید.

https://wiki.libsdl.org/SDL_Keycode

با فشرده یا رها شدن هر کلید کیبورد یک ایونت ایجاد می‌شود که با استفاده از دستور e.type می‌توان نوع آن را دریافت کرد و سپس با دستور e.key.keysym.sym می‌توان keycode آن را گرفت.

```
SDL_Event e;
while ( SDL_PollEvent(&e) ) {
    swtich ( e.type ) {
        case SDL_WINDOWEVENT:
            If ( e.window.event ==
SDL_WINDOWEVENT_CLOSE )
                // alt + f4 or tiny close button on
title bar pressed
                quit(0); // should destroy window
before exit(0);
            break;
        case SDL_KEYDOWN:
        case SDL_KEYUP:
                // a key pressed or a key released
                Int keycode = e.key.keysym.sym;
            break;
    }
}
```

شما می‌توانید قسمت‌های SDL_KEYDOWN, SDL_KEYUP را از هم جدا کنید تا مثلا فقط زمانی که کلیدی فشرده می‌شود اتفاقی بیفتد و با رهاکردن آن اتفاقی نیفتد (که معمولا هم همینطور است).

مثلا پدال گاز را مانند یک دکمه در نظر بگیرید که هرچقدر آن را نگه داریم سرعت خودرو زیاد می‌شود، و با رها کردن آن دکمه سرعت دیگر افزایش نمی‌یابد که کاهش هم می‌یابد. یعنی با هر بار خواندن ایونت از نوع SDL_KEYDOWN سرعت خودرو را افزایش می‌دهیم اما وقتی اولین ایونت از نوع SDL_KEYUP دریافت کردیم نه تنها دیگر افزایش سرعت نداریم بلکه نرخ ثابتی (شما بخوانید شتاب ترمز) داریم که با آن سرعت کاهش می‌یابد تا زمانی که ایونتی از نوع SDL_KEYDOWN بخوانیم یا سرعت خودرو به صفر برسد.

سخن آخر

علاوه بر نکاتی که در این داک پوشش داده شد و در کارگاه‌ها مطرح شد، کتابخانه اس‌دی‌ال قابلیت‌های دیگری مانند پخش صوت و نمایش تصویر هم دارد. علاقه‌مندان می‌توانند سرچ کنند :))

موفق باشید