

2006년 2학기 윈도우 게임 프로그래밍

제3강 키보드 및 마우스 입력 처리

이대현
한국산업기술대학교



한국산업기술대학교

오늘의 학습 내용

- 키보드 입력 처리
- 마우스 입력 처리

캐릭터 원운동 애니메이션



```
#include <SDL/SDL.h>
#include <math.h>

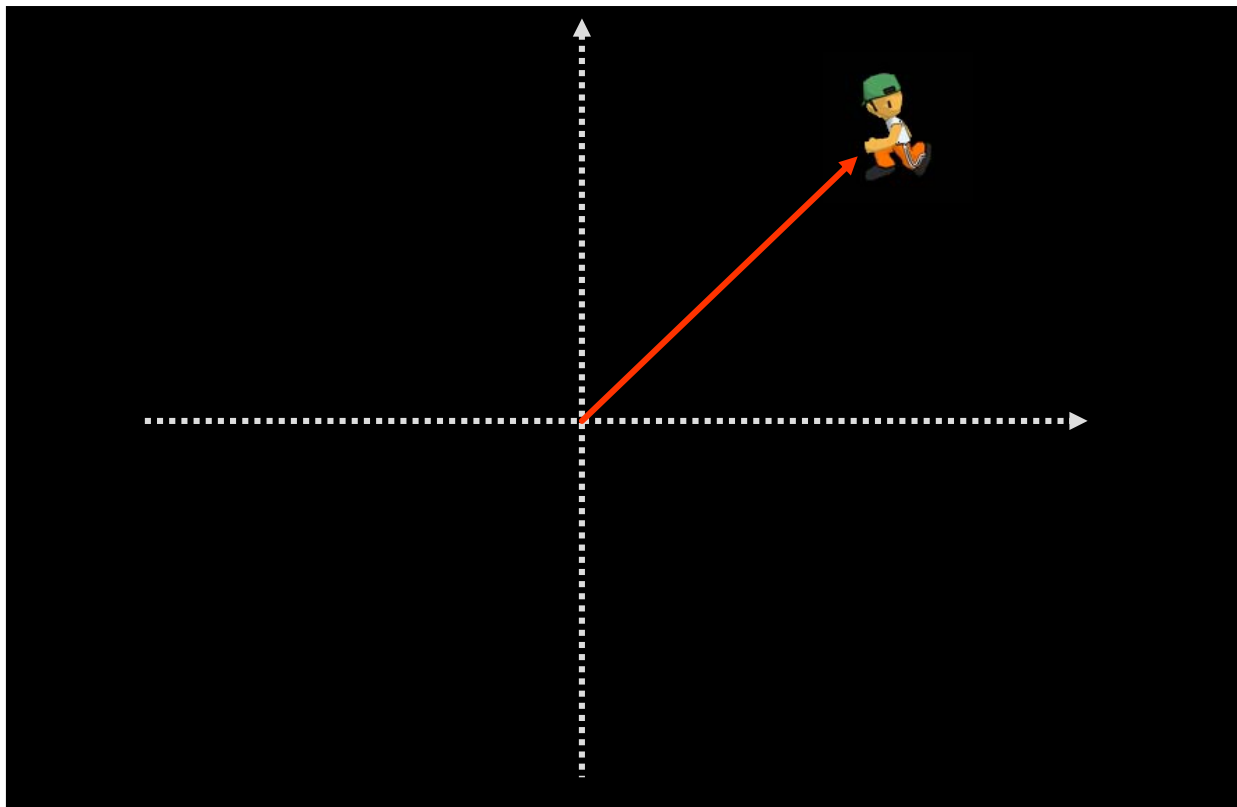
int main(int argc, char** argv)
{
    SDL_Init(SDL_INIT_EVERYTHING);
    SDL_Surface* screen =
        SDL_SetVideoMode(0, 0, 16, SDL_HWSURFACE | SDL_DOUBLEBUF | SDL_FULLSCREEN);
    SDL_Surface* bmp = SDL_LoadBMP("character.bmp");

    int centerx = screen->w / 2;
    int centery = screen->h / 2;
    int radius = 300;

    for (float degree = 0.0f; ; degree += 0.01) {
        int bmpcenterx = centerx + cos(degree) * radius;
        int bmpcentery = centery - sin(degree) * radius;
        SDL_Rect dstrect;
        dstrect.x = bmpcenterx - bmp->w / 2;
        dstrect.y = bmpcentery - bmp->h / 2;
        SDL_BlitSurface(bmp, 0, screen, &dstrect);
        SDL_Flip(screen);
        if (degree > 2 * 3.141592)
            degree = 0.0f;
    }

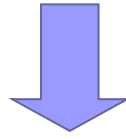
    SDL_FreeSurface(bmp);
    SDL_FreeSurface(screen);
    SDL_Quit();
}
```

캐릭터 좌표의 계산

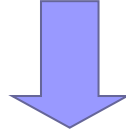


키보드 및 마우스 입력 처리 과정

Step1: 입력 이벤트를 폴링한다. (SDL_PollEvent 함수 이용)



Step2: 이벤트의 종류를 구분한다. (event.type 을 이용)



Step3: 실제 입력값을 구한다. (event.key 또는 event.motion 을 이용)

실습



ESC 키를 통한 프로그램 종료

```

// .....중략.....
int radius = 300;

bool quit = false;
float degree = 0.0f;
while (!quit) {
    SDL_Event event;
    if (SDL_PollEvent(&event)) {
        if (event.type==SDL_KEYDOWN && event.key.keysym.sym == SDLK_ESCAPE)
            quit = true;
    }
    int bmpcenterx = centerx + cos(degree) * radius;
    int bmpcentery = centery - sin(degree) * radius;
    SDL_Rect dstrect;
    dstrect.x = bmpcenterx - bmp->w / 2;
    dstrect.y = bmpcentery - bmp->h / 2;

    SDL_BlitSurface bmp, 0, screen, &dstrect);
    SDL_Flip(screen);

    if (degree > 2 * 3.141592)
        degree = 0.0f;
    else
        degree += 0.01f;
}
// .....후략.....

```

SDL_Event 공용체(Union)

- SDL 이벤트를 다루기 위한 구조체.
- 발생한 이벤트의 정보를 담고 있음.

```

typedef union{
    Uint8 type;
    SDL_ActiveEvent active;
    SDL_KeyboardEvent key;
    SDL_MouseMotionEvent motion;
    SDL_MouseButtonEvent button;
    SDL_JoyAxisEvent jaxis;
    SDL_JoyBallEvent jball;
    SDL_JoyHatEvent jhat;
    SDL_JoyButtonEvent jbutton;
    SDL_ResizeEvent resize;
    SDL_ExposeEvent expose;
    SDL_QuitEvent quit;
    SDL_UserEvent user;
    SDL_SysWMEvent syswm;
} SDL_Event;

```

이벤트 폴링

```
int SDL_PollEvent(SDL_Event *event);
```

- 이벤트를 폴링하여, *event*에 정보를 저장함.

```
SDL_Event event;
```

```
if (SDL_PollEvent(&event)) {
```

- 폴링된 입력이벤트의 종류를 검사.
- *SDL_KEYDOWN*, *SDL_KEYUP*, *SDL_MOUSEMOTION* 등등.

```
if (event.type == SDL_KEYDOWN &&  
    event.key.keysym.sym == SDLK_ESCAPE)  
    quit = true;  
}
```

- 입력된 키 값을 조사.
- *event.key.keysym.sym* 변수에 담겨 있음.
- *SDLK_UP*, *SDLK_DOWN*, *SDLK_LEFT* 등등.



실습

마우스로 캐릭터 이동하기

```
// 앞부분 생략 - 초기화, 스크린 표면 생성 및 이미지 로딩

bool quit = false;
while (!quit) {
    SDL_Event event;
    if (SDL_PollEvent(&event)) {
        switch (event.type) {
            case SDL_KEYDOWN:
                if (event.key.keysym.sym == SDLK_ESCAPE)
                    quit = true;
                break;
            case SDL_MOUSEMOTION:
                SDL_Rect dstrect;
                dstrect.x = event.motion.x - bmp->w / 2;
                dstrect.y = event.motion.y - bmp->h + 20;
                SDL_BlitSurface(bmp, 0, screen, &dstrect);
                SDL_Flip(screen);
                break;
        }
    }
}

// 후략 - SDL 리소스 해제 및 종료
```

마우스 이동 처리

```
case SDL_KEYDOWN:
    if (event.key.keysym.sym == SDLK_ESCAPE)
        quit = true;
    break;

case SDL_MOUSEMOTION:
    SDL_Rect dstrect;
    dstrect.x = event.motion.x - bmp->w / 2;
    dstrect.y = event.motion.y - bmp->h + 20;
    SDL_BlitSurface(bmp, 0, screen, &dstrect);
    SDL_Flip(screen);
    break;
```

- 입력 이벤트의 종류가 마우스의 이동 입력.
- 버튼의 경우는 `SDL_MOUSEBUTTONDOWN`, `SDL_MOUSEBUTTONUP`

- 마우스의 위치는 `event.motion.x/y`에 저장.



■ 캐릭터의 회전운동 애니메이션 컨트롤

- Full screen 모드로 모드 설정.
- 프로그램이 시작되면 캐릭터가 화면 정중앙을 중심으로 회전운동을 함.
 - 초기 회전 반지름 100
- 회전원의 중심이동: 마우스 및 상하좌우키를 이용하여 이동.
 - SDLK_UP, SDLK_DOWN, SDLK_LEFT, SDLK_RIGHT 이용
- 회전원의 확대 축소: 'a' → 확대, 'd' → 축소
 - SDLK_a, SDLK_d 이용.
 - 반지름의 범위: 20 ~ 300