

2006년 2학기 윈도우 게임 프로그래밍

제5강 색상키와 알파블렌딩

이대현

한국산업기술대학교



한국산업기술대학교

오늘의 학습 내용

- 색상키 (color key)
- 알파 블렌딩 (alpha blending)

색상키 (Color Key)

- 배경 위에 이미지를 뿌릴 때의 문제점
 - 캐릭터 이외의 부분까지 배경 위에 나타남.



색상키가 적용되지 않음.



색상키가 적용됨.

■ 색상키

- 사각형으로 표현된 캐릭터에서 캐릭터의 형태만 출력하고 필요하지 않은 영역은 화면에 출력하지 않게 할 때 사용하는 색.
- 캐릭터 몸체 부분에는 사용되지 않는 색을 사용해야 한다.

실습



Lecture05-01: 색상키의 구현

Lecture05-01 프로젝트의 구성

- C++ 소스 파일들
 - gameengine.cpp
 - main.cpp
 - introstate.cpp
 - playstate.cpp – 실습 시간에 작성.
- C++ 헤더 파일들
 - gameengine.h
 - gamestate.h
 - introstate.h
 - playstate.h

playstate.cpp (1)

```
void CPlayState::Init()
{
    SDL_Surface* temp = SDL_LoadBMP("character.bmp");
    player = SDL_DisplayFormat(temp);
    SDL_FreeSurface(temp);

    Uint32 colorkey = SDL_MapRGB(player->format, 255, 0, 0);
    SDL_SetColorKey(player, SDL_RLEACCEL|SDL_SRCCOLORKEY, colorkey);

    temp = SDL_LoadBMP("background.bmp");
    bg = SDL_DisplayFormat(temp);
    SDL_FreeSurface(temp);

    radius = 30;
    degree = 0.0f;
    mx = 480 / 2;
    my = 272 / 2;
    px = 0;
    py = 0;
    SDL_WarpMouse(mx, my);
    SDL_ShowCursor(true);
}
```

playstate.cpp (2)

```
void CPlayState::Draw(CGameEngine* game)
{
    SDL_Rect dstrect;
    dstrect.x = px - player->w / 2;
    dstrect.y = py - player->h / 2;

    SDL_BlitSurface(bg, NULL, game->screen, NULL);
    SDL_BlitSurface(player, 0, game->screen, &dstrect);

    SDL_Flip(game->screen);
}
```

표면의 포맷을 최적화

```
SDL_Surface
*SDL_DisplayFormat(SDL_Surface *surface);
```

```
void CPlayState::Init()
{
    SDL_Surface* temp = SDL_LoadBMP("character.bmp");

    player = SDL_DisplayFormat(temp);

    SDL_FreeSurface(temp);
}
```

- *temp* 표면은, 현재 비디오모드에 맞는 형식으로 변환하여, *player*라는 이름의 표면을 생성.
- 로딩된 비트맵 표면을 현재 비디오모드의 픽셀 형식에 맞게 변환하여, 블리팅의 속도를 높임.

색상키의 지정

```
Uint32 SDL_MapRGB(SDL_PixelFormat *fmt,
                  Uint8 r, Uint8 g, Uint8 b);

int SDL_SetColorKey(SDL_Surface *surface,
                   Uint32 flag, Uint32 key);
```

- RGB 값으로부터, 색상키에 해당하는 색상값을 계산함.
- `SDL_MapRGB`를 이용하여 캐릭터 이미지의 픽셀 형식에 맞는 색상값을 계산함.

```
Uint32 colorkey = SDL_MapRGB(player->format, 255, 0, 0);

SDL_SetColorKey(player, SDL_RLEACCEL | SDL_SRCCOLORKEY, colorkey);
```

- 캐릭터 이미지의 색상키를 지정함.
- `flag`를 `SDL_SRCCOLORKEY`로 설정.
- `SDL_RLEACCEL`을 설정하면, 좀더 빠른 색상키 처리가 가능함.

배경과 캐릭터의 렌더링

```
void CPlayState::Draw(CGameEngine* game)
{
    SDL_Rect dstrect;
    dstrect.x = px - player->w / 2;
    dstrect.y = py - player->h / 2;

    SDL_BlitSurface(bg, NULL, game->screen, NULL);
    SDL_BlitSurface(player, 0, game->screen, &dstrect);

    SDL_Flip(game->screen);
}
```

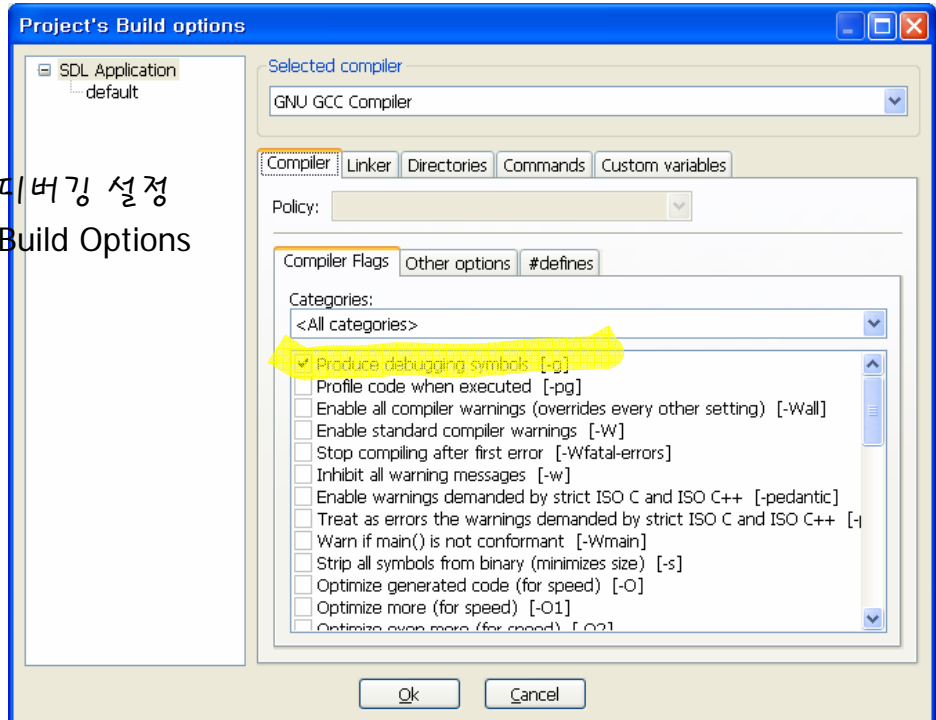
- 계속해서, 배경을 그리고, 그 위에 캐릭터를 뿌린다.

Tip: SDL 프로그램 디버깅 (1)

■ 소스레벨 디버깅 환경 설정

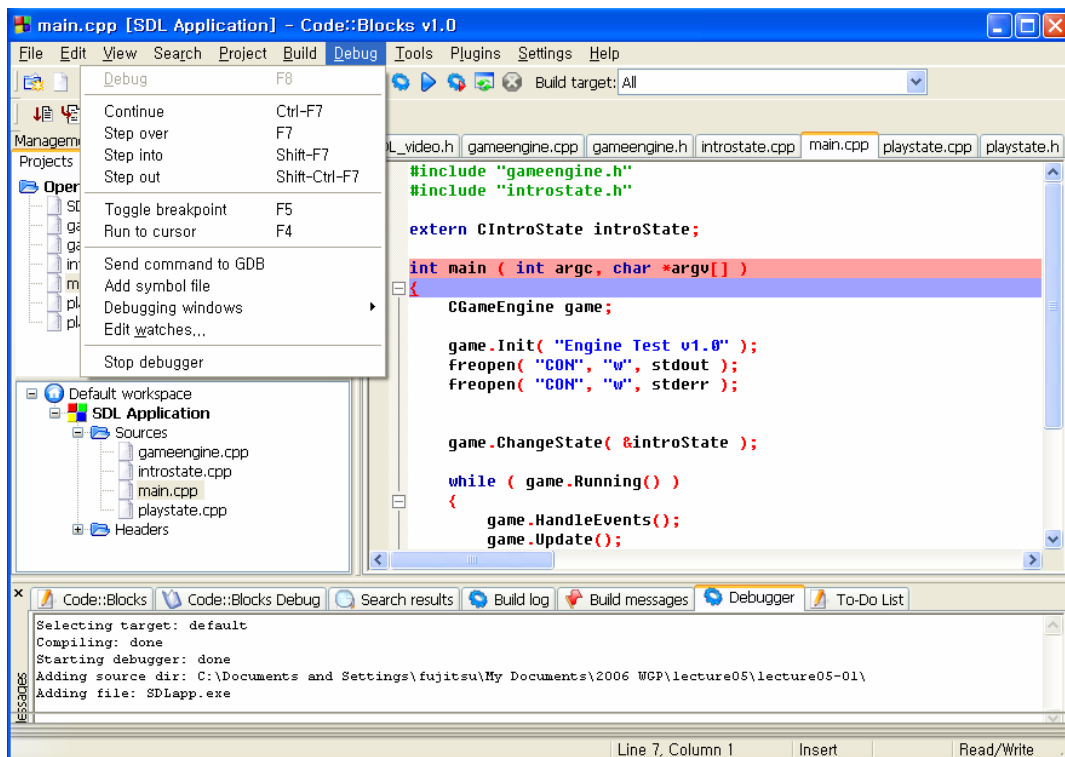
- 먼저, 프로젝트 폴더의 파일 경로가 반드시 영어로 되어 있어야 한다.

- 컴파일 옵션에 디버깅 설정
 - Project → Build Options



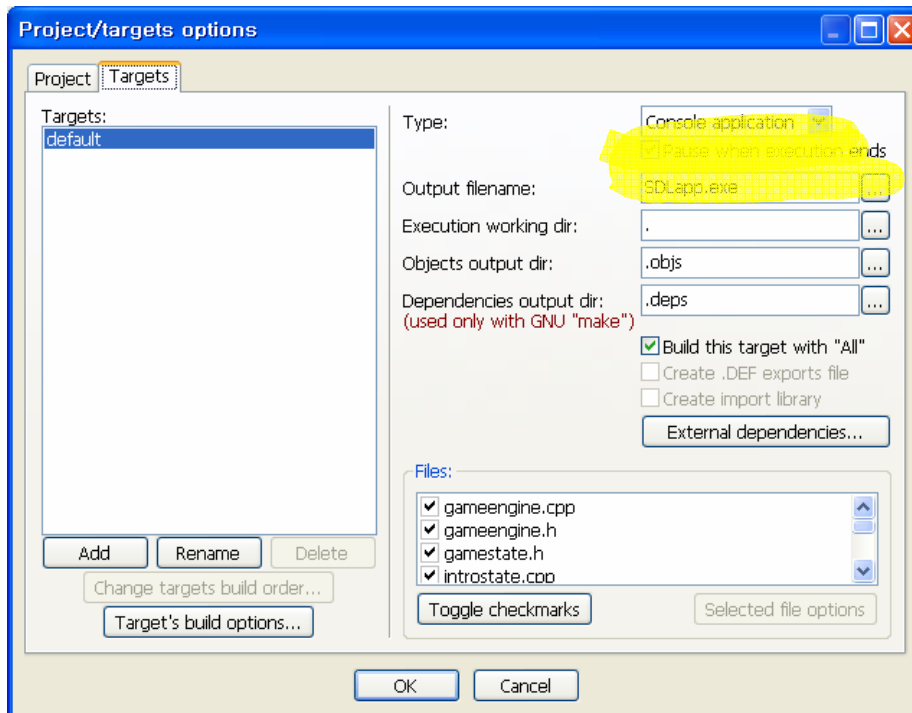
Tip: SDL 프로그램 디버깅 (2)

■ 디버거 실행



Tip: SDL 프로그램 디버깅 (3)

- 콘솔 윈도우를 활용한 디버깅 환경 설정
 - Project → Properties → Targets에서 Console application 설정



Tip: SDL 프로그램 디버깅 (4)

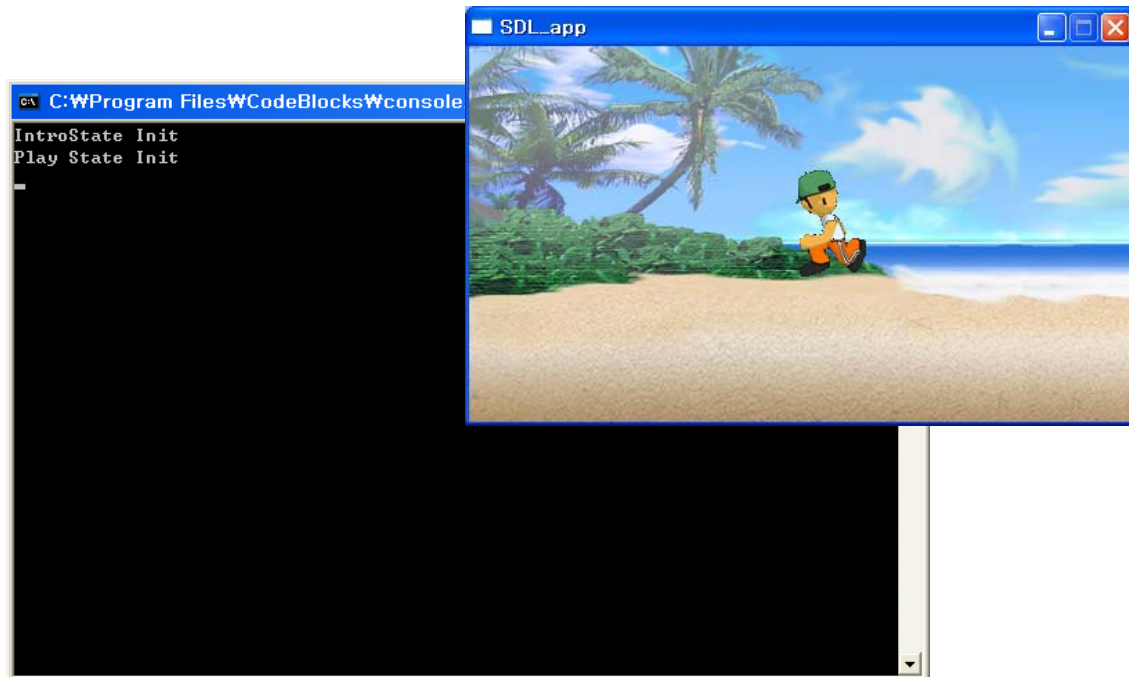
- SDL 라이브러리 초기화 직후에, stdout 및 stderr 출력 처리

```
int main ( int argc, char *argv[] )  
{  
    CGameEngine game;  
  
    game.Init( "Engine Test v1.0" );  
  
    freopen( "CON", "w", stdout );  
    freopen( "CON", "w", stderr );  
}
```

Tip: SDL 프로그램 디버깅 (5)

■ 콘솔 윈도우를 활용한 디버깅

- 프로그램 코드 상에서 printf 문을 사용하여 각종 정보를 출력하면서 디버깅 가능.



알파 블렌딩(Alpha Blending)

■ 알파 블렌딩이란?

- 이미지의 색깔을 반투명하게 하여, 배경과 이미지가 겹쳐지게 하는 것.
- 반투명도를 나타내는 값을 알파라고 함.
- 알파의 범위: 0-255(0이면 완전 투명, 255이면 불투명)



알파값: 50



알파값: 100



알파 블렌딩의 구현

playstate.cpp

```
void CPlayState::Init()
{
    SDL_Surface* temp = SDL_LoadBMP("character.bmp");
    player = SDL_DisplayFormat(temp);
    SDL_FreeSurface(temp);
    Uint32 colorkey = SDL_MapRGB(player->format, 255, 0, 0);
    SDL_SetColorKey(player, SDL_RLEACCEL|SDL_SRCCOLORKEY, colorkey);

    SDL_SetAlpha(player, SDL_RLEACCEL | SDL_SRCALPHA, 50);

    temp = SDL_LoadBMP("background.bmp");
    bg = SDL_DisplayFormat(temp);
    SDL_FreeSurface(temp);

    radius = 30;
    degree = 0.0f;
    mx = 480 / 2;
    my = 272 / 2;
    px = 0;
    py = 0;
    SDL_WarpMouse(mx, my);
    SDL_ShowCursor(true);
}
```

알파값의 지정

```
int SDL_SetAlpha(SDL_Surface *surface,  
                 Uint32 flags, Uint8 alpha);
```

```
SDL_SetAlpha(player, SDL_RLEACCEL | SDL_SRCALPHA, 50);
```

- *player* 캐릭터 이미지에 대해서, 알파값을 50으로 설정.
- 플래그로 *SDL_SRCALPHA*를 지정함.

실습 과제 #4



- 캐릭터 영혼의 움직임.
 - 방향키를 사용하여, 캐릭터를 상하좌우로 이동함.
 - 'g' 키를 누르면, 캐릭터는 움직임을 멈추고 캐릭터 영혼이 나오게 됨. 이후로는 영혼이 방향키에 따라서 상하좌우로 이동함. 다시 'g'를 누르면, 영혼은 사라지고, 원래 캐릭터가 다시 움직이게 됨. 영혼은 알파값은 50~200 까지 계속적으로 변함.
 - 참고: 키보드의 연속입력 활성화 방법
 - `SDL_EnableKeyRepeat(SDL_DEFAULT_REPEAT_DELAY, SDL_DEFAULT_REPEAT_INTERVAL);` 은 SDL 초기화 때 호출함.



프로젝트 발표 #1

- 일시: 9월 21일 (목) 오전 11시
- 발표 시간: 팀당 10분 발표 5분 질의응답.
- 발표 요령: 게임 기획서 파워포인트 자료를 구두 발표
- 평가 포인트
 - 족구 게임 재미 요소가 얼마나 발굴되었는가?
 - 팀원별로 업무 분담이 효과적으로 이루어졌는가?
 - 게임 개발 계획이 충실하게 이루어졌는가?
- 게임기획서 사전 제출
 - 메일 제목: 2006 WGP 게임기획서 팀이름
 - 파워포인트로 작성하여 압축하지 말고 그대로 첨부 파일로 보낼 것.
 - 제출 기한: 9월 20일 정오 12시까지.