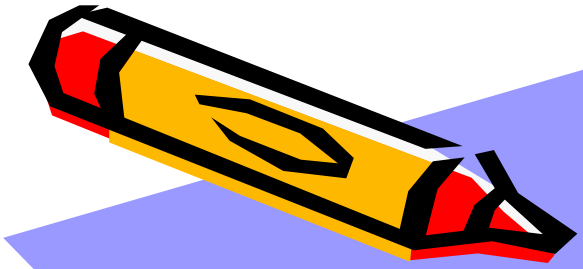


2006년 2학기 윈도우 게임 프로그래밍

제4강 게임 프레임워크 (Game Framework)

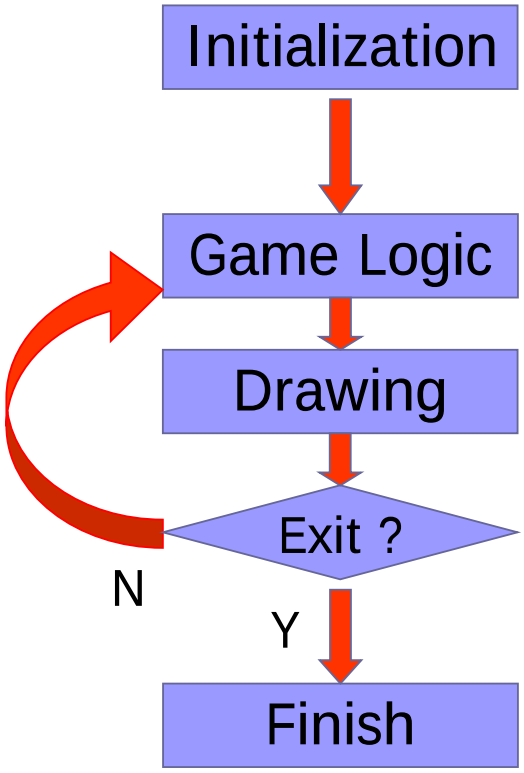
이대현

한국산업기술대학교

오늘의 학습 내용

- 게임 루프
- 게임 상태
- 게임 프레임워크

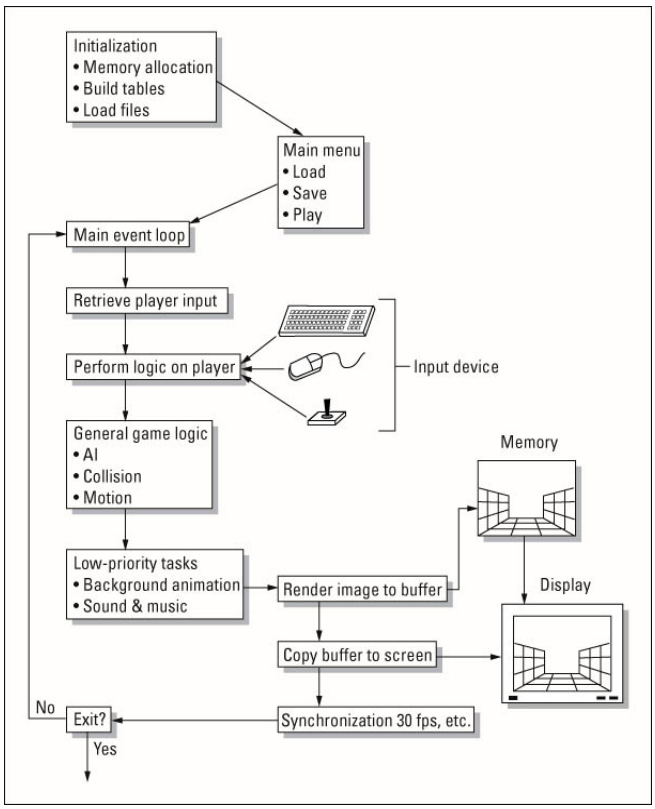
게임 루프



게임 루프의 예

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

    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;
            }
        }
    }

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

실제 게임 루프



실습 과제 #3

- 게임 시작 화면과 메뉴 선택 화면의 구성
  - 480x272 싸이즈 16bit 컬러 모드로 화면 설정
  - 프로그램이 시작되면, 게임 시작 화면 출력
    - Space 를 누르면 메뉴 선택 화면으로 전환
    - ESC 을 누르면 프로그램 종료
  - 메뉴 선택 화면에서
    - ESC 를 누르면 게임 시작 화면으로 전환

intro.bmp

menu.bmp

게임 상태(Game State)의 이해 (1)

- 게임 상태란?
  - 게임 프로그램 실행 중의 어떤 특정 위치(또는 모드).
  - 사용자 입력(키보드 또는 마우스 입력)에 대한 대응 방식은 게임의 상태에 따라 달라짐.

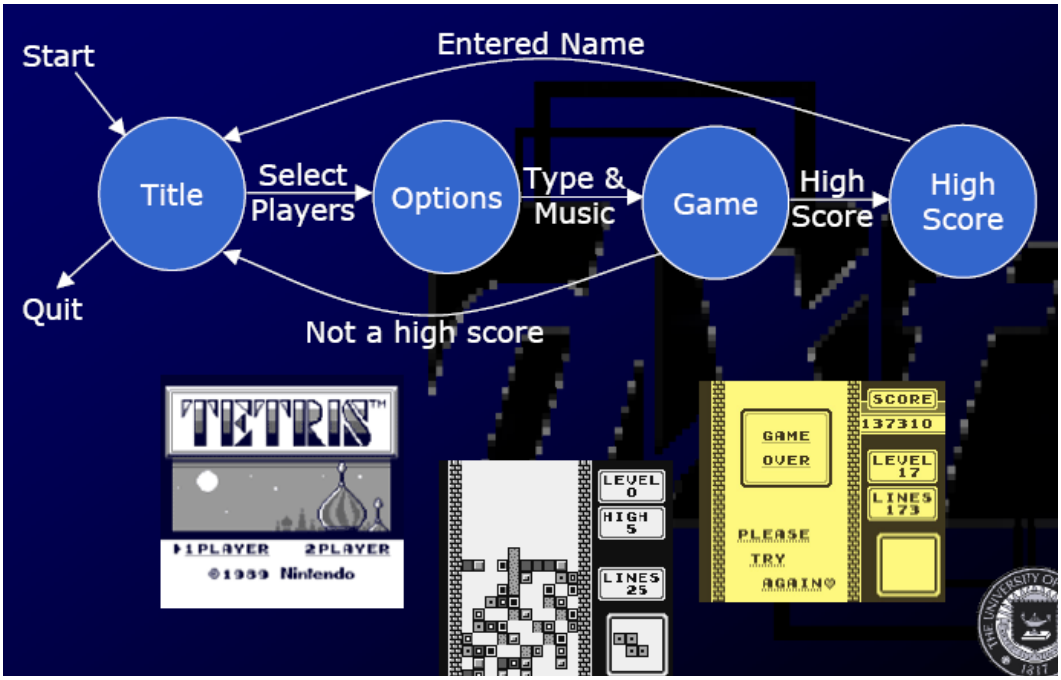
- 맵 선택 상태.
- 방향키는 맵 선택을 처리.

- 게임 메인 플레이 상태.
- 방향키는 캐릭터의 이동을 처리.

게임 상태(Game State)의 이해 (2)

- 게임 프로그램은 게임 상태의 집합으로 구현됨.

  예) 테트리스 게임

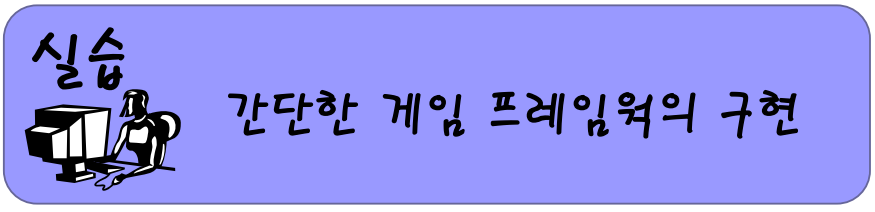


게임 상태(Game State)의 이해 (3)

- 플레이 모드 내에서도 게임 상태의 세분화가 가능
  - 가능하면 작은 단위의 게임 상태로 세분화할 수록 개발 및 디버깅이 용이.
    - 예) 테니스 경기에서 서브 상태와 스트로크 상태의 구분.

- 서브 상태.
- O 버튼은 서브 동작.

- 스크로크 상태.
- O 버튼은 스트로크 동작.



프로젝트의 구성

- C++ 소스 파일들
  - gameengine.cpp
  - introstate.cpp
  - main.cpp – 강의 시간 중 실습 시 직접 작성.
  - menustate.cpp

- C++ 헤더 파일들
  - gameengine.h
  - gamestate.h
  - introstate.h – 강의 시간 중 실습 시 직접 작성.
  - menustate.h

main.cpp

```
#include "gameengine.h"
#include "introstate.h"

extern CIntroState introState;

int main ( int argc, char *argv[] )
{
   CGameEngine game;
   game.Init( "Engine Test v1.0" );

   game.ChangeState( &introState );
   while ( game.Running() )
   {
        game.HandleEvents();
        game.Update();
        game.Draw();
   }

   game.Cleanup();

   return 0;
}
```

introstate.cpp (1)

```
#include <SDL/SDL.h>
#include "gameengine.h"
#include "gamestate.h"
#include "introstate.h"
#include "menustate.h"

CIntroState introState;
extern CMenuState menuState;

void CIntroState::Init()
{
   bg = SDL_LoadBMP("intro.bmp");
}

void CIntroState::Cleanup()
{
   SDL_FreeSurface(bg);
}

void CIntroState::Pause()
{
}

void CIntroState::Resume()
{
}
```

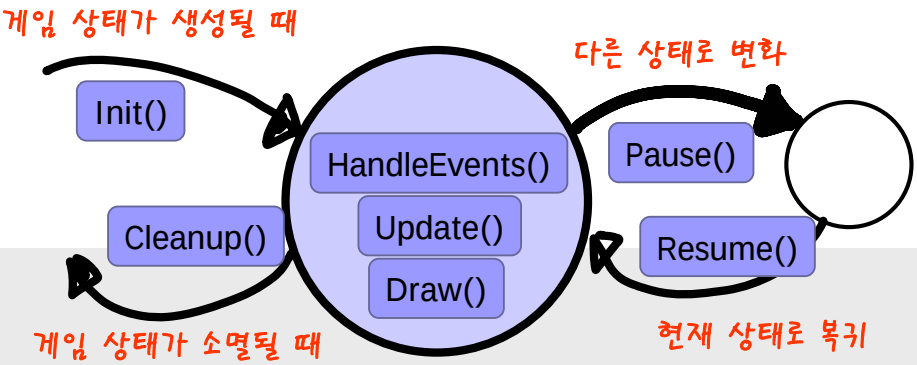

introstate.cpp (2)

```
void CIntroState::HandleEvents(CGameEngine* game)
{
  SDL_Event event;

  if (SDL_PollEvent(&event)) {
    switch (event.type) {
      case SDL_QUIT:
        game->Quit();
        break;
      case SDL_KEYDOWN:
        switch (event.key.keysym.sym) {
          case SDLK_SPACE:
            game->ChangeState( &menuState );
            break;
          case SDLK_ESCAPE:
            game->Quit();
            break;
        }
        break;
    }
  }
}
```

introstate.cpp (3)

```
void CIntroState::Update(CGameEngine* game)
{
}

void CIntroState::Draw(CGameEngine* game)
{
    SDL_BlitSurface(bg, NULL, game->screen, NULL);
    SDL_Flip(game->screen);
}
```


게임 상태의 구현: CGameState 클래스




```
class CGameState
{
public:
    CGameState() { }
    virtual void Init() = 0;
    virtual void Cleanup() = 0;
    virtual void Pause() = 0;
    virtual void Resume() = 0;

    virtual void HandleEvents(CGameEngine* game) = 0;
    virtual void Update(CGameEngine* game) = 0;
    virtual void Draw(CGameEngine* game) = 0;

    void ChangeState(CGameEngine* game, CGameState* state)
    {
          game->ChangeState(state);
    }
};
```

게임 관리자의 구현: CGameEngine 클래스

```
class CGameEngine
{
public:

    void Init(const char* title, int width=480, int height=272,
              int bpp=16, bool fullscreen=false);
    void Cleanup();

    void ChangeState(CGameState* state);
    void PushState(CGameState* state);
    void PopState();

    void HandleEvents();
    void Update();
    void Draw();

    bool Running() { return m_running; }
    void Quit() { m_running = false; }

    SDL_Surface* screen;

private:
    vector<CGameState*> states;
    bool m_running;
    bool m_fullscreen;
};
```

- 게임 상태의 이동.
- 현재 상태를 소멸시킴.

- 게임 상태의 이동.
- 현재 상태를 메모리에 유지한 채로 다음 상태로 이동함.

- 이전 상태로 이동.
- 현재 상태는 소멸됨.

- 현재 상태에 대한 게임 루프 처리.

- 상태들의 보관 및 관리.
- STL의 벡터 클래스를 이용.

- 게임 루프의 계속 실행 여부.

introstate.cpp의 분석 (1)

```
#include <SDL/SDL.h>
#include "gameengine.h"
#include "gamestate.h"
#include "introstate.h"
#include "menustate.h"

CIntroState introState;
extern CMenuState menuState;

void CIntroState::Init()
{
    bg = SDL_LoadBMP("intro.bmp");
}

void CIntroState::Cleanup()
{
    SDL_FreeSurface(bg);
}

void CIntroState::Draw(CGameEngine* game)
{
    SDL_BlitSurface(bg, NULL, game->screen, NULL);
    SDL_Flip(game->screen);
}
```

- introState 객체의 생성.
- 외부의 menuState 객체의 사용 선언.

- introState가 생성되면, 이미지 로딩.

- introState가 소멸되면, 이미지도 제거.

- draw 메소드는 바탕화면만 그려주는 것으로 끝.

introstate.cpp의 분석 (2)

```
void CIntroState::HandleEvents(CGameEngine* game)
{
  SDL_Event event;

  if (SDL_PollEvent(&event)) {
    switch (event.type) {
      case SDL_QUIT:
        game->Quit();
        break;
      case SDL_KEYDOWN:
        switch (event.key.keysym.sym) {
          case SDLK_SPACE:
            game->ChangeState( &menuState );
            break;
          case SDLK_ESCAPE:
            game->Quit();
            break;
        }
        break;
    }
  }
}
```

- 윈도우 창의 X 표를 클릭했을 때 발생하는 이벤트.
- 스페이스키가 눌리면 menuState로 이동.
- ESC키가 눌리면 게임 종료.

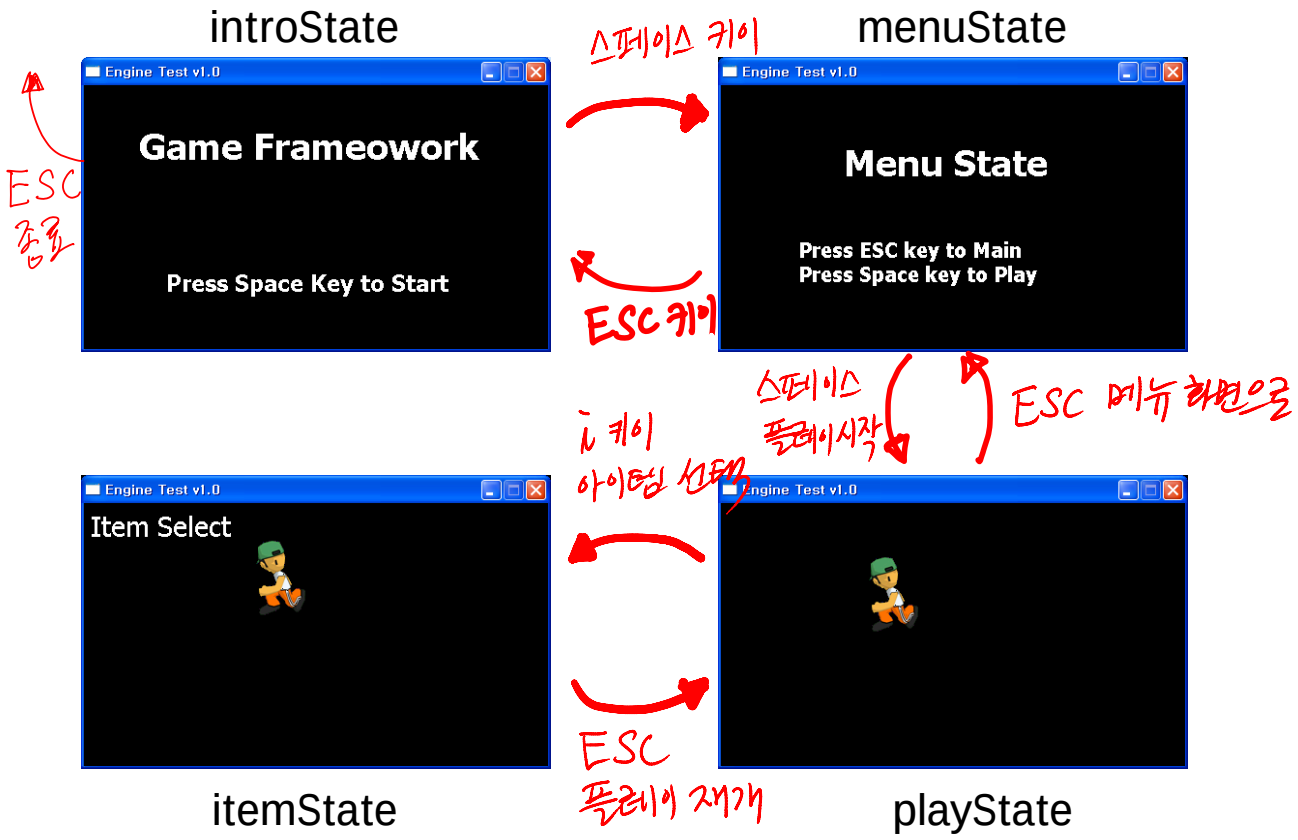
main.cpp의 분석

```
#include "gameengine.h"
#include "introstate.h"

extern CIntroState introState;

int main ( int argc, char *argv[] )
{
   CGameEngine game;
   game.Init( "Engine Test v1.0" );

   game.ChangeState( &introState );

   while ( game.Running() )
   {
        game.HandleEvents();
        game.Update();
        game.Draw();
   }

   game.Cleanup();

   return 0;
}
```

- 게임 관리자 초기화.
- SDL 표면 생성 등등.
- 게임의 시작 상태를 introState로 지정.
- 현재 상태에서 게임 루프를 수행함.
- 내부적으로, 상태의 변화가 일어날 수 있음.

과제 (1)

- 게임 프레임워 안에서 캐릭터의 회전 운동 애니메이션 컨트롤 구현
  - 480x272 화면으로 설정.
  - 모두 4개의 게임 상태로 구성
    - introState
    - menuState
    - playState
    - itemState – item.bmp 의 사용
  - playState의 처리
    - 프로그램이 시작되면 캐릭터가 화면 정중앙을 중심으로 회전운동을 함.
      - 초기 회전 반지름 50
    - 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 ~ 100



참고 함수들

마우스의 위치 조정

```
void SDL_WarpMouse(Uint16 x, Uint16 y);
```

마우스 커서의 화면 표시

```
int SDL_ShowCursor(int toggle);
```

true: 표시
false: 제거

사각형 색칠

```
int SDL_FillRect(SDL_Surface *dst, SDL_Rect
*dstrect, Uint32 color);
```

사각형 좌표
NULL이면 전체화면

색상지정
0 → 검은색

대상표면

- 제출 방법
  - e-mail: kpu.homework@gmail.com
  - 메일 제목: 2006 WGP 과제(1) 학번 이름
  - 프로젝트를 하나의 폴더 안에 만든 후, 폴더를 zip으로 압축해서 메일에 첨부.
  - 제출 기한: 이번주 금요일, 기한 넘기면 50% 감점
  - copy 제출시 0점 (3회시 F)