[ 게임 그래픽 프로그래밍 ]

**2D 게임 프로그래밍**

인하대학교 미래인재개발원 문화컨텐츠개발

김예슬

**2D 게임 개발을 위한 개발 문서**

[ GameManager ]

|  |  |
| --- | --- |
| 함수 | GameManager() |
| 해설 | 기본 생성자 |

|  |  |
| --- | --- |
| 함수 | ~GameManager() |
| 해설 | 기본 소멸자 |

|  |  |
| --- | --- |
| 함수 | static GameManager\* GetInstance() |
| 해설 | static 변수 반환 함수 |

|  |  |
| --- | --- |
| 함수 | void CalcScreenSize(HWND hWnd) |
| 해설 | 윈도우 창 크기 측정 함수. 내부 변수에 값 저장 |

|  |  |
| --- | --- |
| 함수 | void ReadSaveData() |
| 해설 | 세이브 데이터 불러오는 함수. (현재 스테이지와 포커스 레벨)  세이브 데이터가 있을 경우 해당 값을, 없을 경우 초기값을 내부 변수에 저장 |

|  |  |
| --- | --- |
| 함수 | void WriteSaveData() |
| 해설 | 게임이 종료될 때, 세이브 데이터를 저장하는 함수.  (현재 스테이지와 포커스 레벨) |

|  |  |
| --- | --- |
| 함수 | void SetIsPause() |
| 해설 | 현재 일시정지 상태가 아니면 일시정지 설정, 아니라면 일시정지를 해제하는  함수, 일시 정지 상태에 따라 BGM이 재생 유무가 결정됨 |

|  |  |
| --- | --- |
| 함수 | void SetNowScene(int scene) |
| 매개변수 | int scene : 현재 씬 넘버 |
| 해설 | 내부변수에 현재 씬 넘버를 저장하는 함수. |

|  |  |
| --- | --- |
| 함수 | void SetIsPlayerLive(bool live) |
| 매개변수 | bool live : 플레이어 생존 여부 |
| 해설 | 내부 변수에 플레이어 생존 여부 값 저장 함수 |

|  |  |
| --- | --- |
| 함수 | void SetIsPlayerLive(bool live) |
| 매개변수 | bool live : 플레이어 생존 여부 |
| 해설 | 내부 변수에 플레이어 생존 여부 값 저장 함수 |

|  |  |
| --- | --- |
| 함수 | bool GetIsPause() |
| 해설 | 일시정지 상태 값 반환 함수. |

|  |  |
| --- | --- |
| 함수 | bool GetNowScene() |
| 해설 | 현재 씬 넘버 값 반환 함수 |

|  |  |
| --- | --- |
| 함수 | RECT GetScreenSize() |
| 해설 | 화면 크기 값 반환 함수 |

|  |  |
| --- | --- |
| 함수 | void SetDrawRect(bool set) |
| 매개변수 | bool set : 모드 설정 on/off |
| 해설 | 디버그 모드 설정 함수. true일 경우 물체의 히트박스를 출력함. |

|  |  |
| --- | --- |
| 함수 | void SetNowStage(int set) |
| 매개변수 | int set : 현재 스테이지 넘버 |
| 해설 | 내부 변수에 현재 스테이지 넘버를 저장 함수 |

|  |  |
| --- | --- |
| 함수 | void SetFocusLv(int set) |
| 매개변수 | int set : 현재 포커스 레벨 |
| 해설 | 내부 변수에 현재 포커스 레벨 저장 함수 |

|  |  |
| --- | --- |
| 함수 | bool GetDrawRect() |
| 해설 | 디버그 모드 설정 값 반환 함수 |

|  |  |
| --- | --- |
| 함수 | void GetNowStage() |
| 해설 | 현재 스테이지 넘버 반환 함수 |

|  |  |
| --- | --- |
| 함수 | void GetFocusLv() |
| 해설 | 현재 포커스 레벨 값 반환 함수 |

**구동 가능한 2D 게임 프로그램 소스**

#include "stdafx.h"

#include "GameManager.h"

#include "SoundSystem.h"

#include <fstream>

#define dSoundSys SoundSystem::GetInstance()

#define dKeyCode 'k'

using namespace std;

GameManager::GameManager()

{

nowScene = eMainScene;

isPause = false;

isPlayerLive = true;

isDrawRect = false;

ReadSaveData();

}

GameManager::~GameManager()

{

WriteSaveData();

}

GameManager\* GameManager::GetInstance()

{

static GameManager gameManager;

return &gameManager;

}

void GameManager::CalcScreenSize(HWND hWnd)

{

GetClientRect(hWnd, &screenSize);

}

void GameManager::ReadSaveData()

{

ifstream readFile;

readFile.open("saveData.dat", ios::in | ios::binary);

if (readFile.is\_open())

{

// 자료 읽어오기

readFile.read((char\*)&nowStage, sizeof(int));

readFile.read((char\*)&nowFocusLv, sizeof(int));

// decode

nowStage = (nowStage / dKeyCode) - dKeyCode;

nowFocusLv = (nowFocusLv / dKeyCode) - dKeyCode;

}

else

{

nowStage = -1;

nowFocusLv = 0;

}

readFile.close();

}

void GameManager::WriteSaveData()

{

ofstream saveFile;

saveFile.open("saveData.dat", ios::out | ios::binary);

// encode

nowStage = (nowStage + dKeyCode) \* dKeyCode;

nowFocusLv = (nowFocusLv + dKeyCode) \* dKeyCode;

saveFile.write((char\*)&nowStage, sizeof(int));

saveFile.write((char\*)&nowFocusLv, sizeof(int));

saveFile.close();

}

void GameManager::SetIsPause()

{

if (isPause == false)

isPause = true;

else

isPause = false;

dSoundSys->SetIsPause(isPause);

}

void GameManager::SetIsPlayerLive(bool live)

{

isPlayerLive = live;

}

void GameManager::SetNowScene(int scene)

{

nowScene = scene;

}

bool GameManager::GetIsPause()

{

return isPause;

}

bool GameManager::GetIsPlayerLive()

{

return isPlayerLive;

}

int GameManager::GetNowScene()

{

return nowScene;

}

RECT GameManager::GetScreenSize()

{

return screenSize;

}

void GameManager::SetDrawRect(bool set)

{

if (set == true)

isDrawRect = true;

else

isDrawRect = false;

}

void GameManager::SetNowStage(int set)

{

nowStage = set;

}

void GameManager::SetFocusLv(int set)

{

nowFocusLv = set;

}

bool GameManager::GetDrawRect()

{

return isDrawRect;

}

int GameManager::GetNowStage()

{

return nowStage;

}

int GameManager::GetFocusLv()

{

return nowFocusLv;

}