[ 게임 그래픽 프로그래밍 ]

**3D 게임 프로그래밍**

인하대학교 미래인재개발원 문화컨텐츠개발

김예슬

**3D 게임 개발을 위한 개발 문서**

[ class GameManager ]

|  |  |
| --- | --- |
| static CGameManager\* GetInstance() | |
| 인자 | X |
| 반환 값 | CGameManager |
| 설명 | 게임 매니저의 인스턴스 반환 |

|  |  |
| --- | --- |
| bool GetDevMode() | |
| 인자 | X |
| 반환 값 | bool |
| 설명 | 개발 모드 확인 변수 반환 |

|  |  |
| --- | --- |
| bool GetUImode() | |
| 인자 | X |
| 반환 값 | bool |
| 설명 | 게임 UI 모드 확인 변수 반환 |

|  |  |
| --- | --- |
| void ReceiveInput(UINT message, WPARAM wParam, LPARAM lParam) | |
| 인자 | message, wParam, lParam : WndProc()의 인자와 동일 |
| 반환 값 | X |
| 설명 | 전달받은 입력 메시지를 처리 |

|  |  |
| --- | --- |
| string GetName() | |
| 인자 | X |
| 반환 값 | string |
| 설명 | 해당 객체의 이름 반환 |

**구동 가능한 3D 게임 프로그램 소스**

#include "stdafx.h"

#include "GameManager.h"

CGameManager::CGameManager()

{

m\_strName = "GameManager";

m\_isDevMode = false;

m\_isUIMode = false;

m\_isGridMap = false;

SetClipCursor(0);

ShowCursor(false);

}

CGameManager\* CGameManager::GetInstance()

{

static CGameManager instance;

return &instance;

}

bool CGameManager::GetDevMode()

{

return m\_isDevMode;

}

bool CGameManager::GetUImode()

{

return m\_isUIMode;

}

bool CGameManager::GetGridMapMode()

{

return m\_isGridMap;

}

void CGameManager::ReceiveEvent(ST\_EVENT eventMsg)

{

RECT rc;

POINT p1, p2;

if (eventMsg.eventType == EventType::eInputEvent)

{

switch (eventMsg.message)

{

case WM\_KEYDOWN:

{

if (VK\_CONTROL == eventMsg.wParam && !m\_isUIModeIn)

{

m\_isUIMode = !m\_isUIMode;

m\_isUIModeIn = !m\_isUIModeIn;

if (m\_isUIMode)

{

SetClipCursor(-15);

ShowCursor(true);

}

else

{

SetClipCursor(0);

ShowCursor(false);

}

}

if (VK\_TAB == eventMsg.wParam && !m\_isDevMoveIn)

{

m\_isDevMode = !m\_isDevMode;

m\_isDevMoveIn = !m\_isDevMoveIn;

}

if (VK\_SCROLL == eventMsg.wParam && !m\_isGridMapIn)

{

m\_isGridMap = !m\_isGridMap;

m\_isGridMapIn = !m\_isGridMapIn;

}

}

break;

case WM\_KEYUP:

{

if (VK\_CONTROL == eventMsg.wParam)

m\_isUIModeIn = false;

if (VK\_TAB == eventMsg.wParam)

m\_isDevMoveIn = false;

if (VK\_SCROLL == eventMsg.wParam)

m\_isGridMapIn = false;

}

break;

}

}

}

string CGameManager::GetName()

{

return m\_strName;

}

void CGameManager::SetClipCursor(int ySize)

{

RECT rc;

POINT p1, p2;

GetClientRect(g\_hWnd, &rc);

p1.x = rc.left;

p1.y = rc.top + ySize; // x\_Btn

p2.x = rc.right;

p2.y = rc.bottom;

ClientToScreen(g\_hWnd, &p1);

ClientToScreen(g\_hWnd, &p2);

rc.left = p1.x;

rc.top = p1.y;

rc.right = p2.x;

rc.bottom = p2.y;

ClipCursor(&rc);

}