[ 게임 데이터베이스 프로그래밍 ]

**게임 데이터 유지보수**

인하대학교 미래인재개발원 문화콘텐츠프로그래밍

김예슬

**1) 데이터 구조 설계**

- 게임 맵 툴과 게임 내에서 실행될 맵 구조

우상단 / 좌상단 / 우하단 / 좌하단 위치 값

각 타일의 속성 값

플레이어 리젠 위치의 경우 1번 받아와야 함

각각의 값은 특정 알파벳 단어로 암호화가 되어 있어야 하고, 복호화를 통해 원래 값을 알 수 있어야 함

**2) 데이터 생성 및 수정**

- 게임 맵 툴 에서 이루어짐

- 맵 툴에서 이미지 버튼을 클릭해서 각각의 타일과 플레이어 리젠 위치를 설정할 수

있음

- 버튼 클릭 시 현재 선택된 타일이 무엇인지 표시됨

- 좌클릭으로 타일 배치, 우클릭으로 타일 제거가 가능하며 드래그 기능 지원으로 보다

빠르게 데이터를 생성할 수 있음

- 배치한 타일들이 마음에 들지 않을 때, 리셋버튼으로 모든 타일을 한 번에 지울 수

있음

- 새로운 맵을 만들어서 저장 할 수 있음

- 기존에 존재하는 맵을 불러와서 수정한 후 저장할 수 있음

- 맵을 다 찍고, 저장하려 할 때 플레이어 리젠 위치가 설정되어 있지 않으면 저장되지

않음

- .map이라는 확장자를 사용하여 저장함

**3) 데이터 적용**

- 인 게임 내에서 적용됨

- 맵 툴에서 생성된 데이터를 불러온 후, 인 게임에서 실행됨

- 맵 툴의 데이터가 크지 않기 때문에, 다음 스테이지를 넘어가는 순간에 로드함

- 맵 데이터를 읽어온 후, 맵에 오브젝트들이 존재할 경우 오브젝트 로드가 시작됨

(맵 데이터 자체에 오브젝트 데이터가 함께 있음)

- 스테이지 클어 전 플레이어가 메인 화면을 가거나 사망할 경우, 해당 스테이지가

재 실행되는 구조

- 모든 스테이지 클리어한 후 재 실행 혹은 메인 화면 후 실행될 경우 맨 처음 스테이지부터 시작됨

**4) 데이터 복원**

- 버전 관리 시스템을 이용한 데이터 관리 및 복원

**5) 관련 소스코드**

void Map::ReadMapData()

{

string fileName;

ifstream mapFile;

if (dGameManager->GetNowStage() < 10)

fileName = "map\_0";

else

fileName = "map\_";

fileName += to\_string(dGameManager->GetNowStage()) + ".dat";

fileName = "./Map/" + fileName;

mapFile.open(fileName, ios::in | ios::binary);

if (mapFile.is\_open())

{

TileMap tileMap;

tileMap.pos = { 0,0,0,0 };

tileMap.type = 0;

mapFile.read((char\*)&resenSpot.x, sizeof(int));

mapFile.read((char\*)&resenSpot.y, sizeof(int));

resenSpot.x = (resenSpot.x / dKeyCode) - dKeyCode;

resenSpot.y = (resenSpot.y / dKeyCode) - dKeyCode;

while (!mapFile.eof())

{

mapFile.read((char\*)&tileMap.type, sizeof(int));

mapFile.read((char\*)&tileMap.pos.left, sizeof(int));

mapFile.read((char\*)&tileMap.pos.top, sizeof(int));

mapFile.read((char\*)&tileMap.pos.right, sizeof(int));

mapFile.read((char\*)&tileMap.pos.bottom, sizeof(int));

tileMap.type = (tileMap.type / dKeyCode) - dKeyCode;

tileMap.pos.left = (tileMap.pos.left / dKeyCode) - dKeyCode;

tileMap.pos.top = (tileMap.pos.top / dKeyCode) - dKeyCode;

tileMap.pos.right = (tileMap.pos.right / dKeyCode) - dKeyCode;

tileMap.pos.bottom = (tileMap.pos.bottom / dKeyCode) - dKeyCode;

if (tileMap.type <= 0)

break;

mapPos.push\_back(tileMap);

}

dSoundSys->PlayChangeStageSound();

}

else

{

dGameManager->SetNowStage(-1);

dGameManager->SetFocusLv(0);

dGameManager->SetNowScene(eResultScene);

}

mapFile.close();

resetPos = mapPos;

isNextStage = false;

}