C++프로그래밍및실습

리소스 관리 모듈

진척 보고서 #2

제출일자: 2023.12.03

제출자명: 김기태

제출자학번: 185482

1. 프로젝트 목표

1) 배경 및 필요성

개인적으로 렌파이라는 툴을 이용한 비주얼 노벨 개발 작업을 진행 중에 있다. 작업을 진행할 때 관리해야 할 리소스가 너무나도 많아 불편한 점이 많이 있었기 때문에 이를 보완하기 위한 실사용 목적으로 개발한다. 실제로 현재 개발중인 버전 기준으로 리소스 정의 코드 + 시나리오 스크립트가 함께 들어 있어서 만 줄가량을 뛰어넘는 스크립트 문서의 분량 때문에 그때그때 필요한 부분을 제때 찾기에 불편하고 시간이 소요되었기 때문에 스크립트 문서를 분할하여 각 리소스관리 부분을 담당하는 텍스트 파일을 따로 만들 필요성이 생겼기 때문이다. 또한개발 중인 리소스의 용량이 4GB를 넘는 문제 때문에 완성 후 배포 시에 필요 없는 더미 데이터들을 식별하여 삭제하여서 용량을 최소화시켜야만 할 필요가 생겼기 때문이기도 하다.

2) 프로젝트 목표

첫째로는 외부 파일에 리소스 정의에 사용된 코드들을 자동적으로 저장해내는 것이다. 렌파이의 스크립트 기반인 'rpy' 확장자의 텍스트 파일로 저장하도록 한다. 따라서 보유 리소스의 내용이 변동되어도 단 한 번의 툴 사용만으로 모든 리소스의 정의 내역이 현재 파일과 정확히 연동되도록 할 것이다.

둘째로는 외부 텍스트 파일을 읽어 들여서 해당 텍스트 파일에서 사용된 파일과 사용되지 않은 파일을 찾아내서 명단화하는 것이다. 여기서 읽기에 사용할 파일 은 script.rpy로 이 파일의 텍스트를 읽어 들여서 만일 사용한 적이 있는 리소스 라면 사용된 리소스의 목록을 담은 텍스트 파일에 분류하고, 사용한 적이 없는 리소스면 사용되지 않은 리소스 목록을 담은 텍스트 파일로 분류한다.

3) 차별점

기존에 리소스 정의 및 정리 용도로 사용하던 파이썬 기반 코드들은 사용자가 직접 스크립트 파일에 복사 붙여넣기를 해 주어야만 한다는 단점이 있었다. 따라서 기존 코드와는 달리 **C++로 재설계**함과 동시에 외부 파일에 결과값을 저장할 수 있도록 하며, 어떤 환경에서도 실행되도록 절대 경로를 상대경로로 변경하고, 기존 코드들이 배경음악, 사운드 이펙트, 스탠딩 그래픽, 배경 그래픽 등으로 나뉘어져 있었던 것을 한 파일 안에 통합환경으로 구현할 계획이다.

2. 기능 계획

1) 리소스 정의 및 외부 파일에 자동 저장

- 각자 서식에 맞는 리소스 정의를 개별 파일에 저장하고, 리소스 정의 파일을 각자 파일에 맞는 audio.rpy, scg.rpy, bg.rpy등의 파일에 저장한다. 종류별로 따로 따로 저장함으로써 스크립트의 가독성을 높이고 관리를 더 쉽게 할 수 있다.

2) 사용되지 않은 리소스 식별 기능

사용되지 않은 리소스를 스크립트 파일에서 동일 문자열의 존재 여부를 확인하여 단 하나라도 존재하면 존재하는 리소스 목록에 등재하고, 존재하지 않으면 존재하지 않는 리소스 목록에 등재하여 txt파일로 내보내는 것이다. 이는 테스트 버전 내보내기 및 개발 완료 단계에서 더미 데이터의 최소화를 통한 저장공간 절약에 도움이 될 것이다.

3. 진척사항

1) 기능 구현

(1) 사용되지 않은 리소스 식별 기능

- 입출력

입력값은 game폴더 안에 있는 script.rpy이며, 출력값은 현재 상태로는 콘솔에 출력한다.

후일 콘솔뿐만 아니라 txt파일로 출력하도록 할 예정이다.

- 설명

script.rpy를 읽어 와서, 그 파일의 내용에서 특정한 문구 이후의 내용을 한 줄 추출한다. 예를 들면 play music Clair_de_Lune fadein 0.5라는 음악 재생을 의미하는 스크립트 명령 어가 script.rpy에 저장되어 있다고 치자. 이때 "play music"을 기준으로 어떤 줄을 추출해야 하는지를 식별한다. 그 뒤 play music 뒤에 있는 Clair_de_Lune fadein 0.5라는 문자열에서 가장 처음 오는 공백을 기준으로 앞부분만 따로 잘라내서 배열에 저장한다. 따라서남는 문자열은 Clair_de_Lune이 된다. SCG의 경우는 조금 특수한 경우이기에, 두 번째 공백을 기준으로 자르며, 첫 번째 공백은 언더바로 대체된다. 이후 이를 콘솔창에 한하여출력하는 것이 현재까지의 기능이다.

- 적용된 배운 내용 (예: 반복문, 조건문, 클래스, 함수, 포인터 등)

조건문, 반복문, 함수, 벡터 배열 등이 사용된다. 배우지 않은 내용이지만 set라이브러리가 중복 방지용으로 사용된다.

- 코드 스크린샷

```
◎ | 智 - 🛎 💾 😰 |
  main.cpp ◆ ×
                                                                                                                                                                                   cout << "1. BGM 재정의" << endl;
cout << "2. SE 재정의" << endl;
cout << "3. SCG 재정의" << endl;
cout << "4. BG 재정의" << endl;
                               subchoice_wrong_input = SubChoice(subchoice);
if (subchoice_wrong_input == true) {
                          i break;
} else if (choice == 3) {
cout << "사용된 리소스 검출 작업을 시작합니다." << endl;
                                                                                                                                                                                   ◆ 출루션 탐색기 Glt 변경 내용
                                                                                                                                                                                   } else {
cout << "잘못된 입력입니다. 다시 입력해 주세요." << endl;
             7 😑 bool SubChoice(int subchoice) {
② 문제가 검색되지 않음 -
                                                                                                                                                                ↑ 소스 제어에 추가 • ♦ 리포지토리 선택 • 🚨
고 준비
                                                                                                                                                                               로그인 첫 — G X
Live Share R
.
● 마일(F) 편집(E) 보기(V) Gh(G) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N) 도구(T) 확장(X) 창(W) 도움말(H) 검색(CbH+Q)
main.cpp + X
                    ্ৰেপ্ৰস্থল

void ProcessLine(const string& line, const string& search_string, set<string>& result_set);

void PrintSet(const string& set_type, const set<string>& result_set);

void ProcessLineForSCG(const string& line, const string& search_string, set<string>& result_set);
                   ⊟int main() {
                     | Int main() {
| int choice = 0;
| int subchoice = 0;
| cout << "렌파이 전용 리소스 관리 모듈입니다." << endl;
| cout << "해야 할 작업의 번호를 입력해 주세요." << endl;
                   | while (true) {
| : cout << "1. 전체 재정의" << endl;
| : cout << "2. 부분 재정의" << endl;
| : cout << "3. 사용된 리소스 검출" << endl;
                          if (choice == 1) {
cout << "전체 재정의 작업을 수행합니다." << endl;
                                                                                                                                                                                   break;
} else if (choice == 2) {
cout << "부분 재정의 작업을 수행합니다." << endl;
cout << "원하는 작업의 번호를 입력해 주세요." << endl;
               ◎ 문제가 검색되지 않음
                                                                                                                                                                       스 제어에 추가 🔺 💠 리포지토리 선택 🔺 🚨
```

```
%p - ≅ 🖺 📵
 main.cpp ◆ ×
                                                                                                                                            - (전역병위)
if (subchoice == 1) {
    cout << "BGM 재정의를 시작합니다." << endl;
                    BGMDefine();
                  return false;
} else if (subchoice == 2) {
cout << "SE 재정의를 시작합니다." << endl;
                  return false;
} else if (subchoice == 3) {
cout << "SCG 재정의를 시작합니다." << endl;
SCGDefine();
                  } else if (subchoice == 4) {
cout << "BG 재정의를 시작합니다." << endl;
                                                                                                                                            ◆ 출루션 탐색기 Git 변경 내용
                     BGDefine();
                  } else {
| cout << "잘못된 입력입니다." << endl;
                                                                                                                                            BGMDefine();
           SCGDefine()

● 문제가 검색되지 않음
                                                                                                                             ↑ 소스 제어에 추가 • ♦ 리포지토리 선택 • 🚨
000 파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 벨드(B) 디버그(D) 테스트(S) 분석(N) 도구(T) 확장(X) 창(W) 도움말(H) 검색((Ctrl+Q)

    ⊕ • ○ | %p • ≅ № № | %

                                                                                                                                       main.cpp ⊕ X
              BGDefine();

  □void BGMDefine() {

                 vector<string> original_bgm_file_paths;
vector<string> original_bgm_file_names;
                  string bgm_output_filename = "bgm.rpy
                  ofstream out_file(bgm_output_filename);
                  fs::path bgm_dir = "./renpy_example/game/audio/BGM/"; // 문서 경로 지정 if (fs::exists(bgm_dir) &&
                      fs∷is_directory(bgm_dir)) // 해당 디렉토리 존재 여부 확인
                   original_bgm_file_names.push_back(
    entry.path().filename().stem().string());
                          .
// 파일 경로 출력
          ❷ 문제가 검색되지 않음
                                                                                                                                      付에 추가 ▲ 🔷 리포지토리 선택 ▲
```

```
160 - ≦ 🖺 📵
 main.cpp ⊕ ×
                                                                                                                                         cout << entry.path() << endl;
                      out << "파일명을 렌파이 인식 가능 버전으로 전환중..." << endl;
for (const auto& str : original_bgm_file_names) {
                        string modified = str;
                                                                                                                                              ▷ ++ main.cpp
등 해더 파일
                        for (char& c : modified) {
   if (c == ' ')
                                                                                                                                          ◆ 출루션 탐색기 Git 변경 내용
                                                                                                                                         ◎ 문제가 검색되지 않음
                                                                                                                           ↑ 소스 제어에 추가 • ♦ 리포지토리 선택 • 🚨
                      구 : Debug • x64 • ▶ 오컵 Windows 디버기 • ▷ Ø • 하 등 등 등 등 등 등 및 1 등 개 및 의 자자 및 다
⊕ - ○ | %p - ≅ ₽ ₽ |
                                                                                                                                     main.cpp ⊕ ×
                        ্ৰেপ্ৰপ)
out_file << "\"" << original_bgm_file_paths[i] << "\"" << endl;
                     out_file.close(); // 파일 스트림 닫기
cout << "파일 저장 완료: " << bgm_output_filename << endl;
} else {
                   } catch (const fs::filesystem_error& e) {
| cerr << e.what() << endl;
                   cout << "디렉토리가 존재하지 않습니다." << endl;
cout << "메인으로 돌아갑니다." << endl;
             □void SEDefine() {
                                                                                                                                         vector<string> original_se_file_paths;
                 vector<string> modified_se_file_names;
string se_output_filename = "se.rpy";
                fs::path se_dir = "./renpy_example/game/audio/SE/"; // 문서 경로 지정
if (fs::exists(se_dir) &&
          □: fs::is_directory(se_dir)) // 해당 디렉토리 존재 여부 확인
● 및제가 함께되지않함 (
                                                                                                                                    에 추가 ▲ ♦ 리포지토리 선택 ▲
```

```
%p - ≅ 🖺 📵
  main.cpp + X
                                                                                                                                                                            -0-0-0
                                                                                                                                                                                 ▷ ++ main.cpp
등 해더 파일
                                original_se_file_names.push_back(
entry.path().filename().stem().string());
                            기 최근 정도 출력
cout << entry.path() << endl;
                                                                                                                                                                            ◆ 출루션 탐색기 Git 변경 내용
                           r
cout << "파일명을 렌파이 인식 가능 버전으로 전환중..." << endl;
for (const auto& str : original_se_file_names) {
string modified = str;

    문제가 검색되지 않음

                                                                                                                                                          ↑ 소스 제어에 추가 • ♦ 리포지토리 선택 • 🚨
000 파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 벨드(B) 디버그(D) 테스트(S) 분석(N) 도구(T) 확장(X) 창(W) 도움말(H) 검색((Ctrl+Q)
                            구 : Debug • x64 • ▶ 오컵 Windows 디버기 • ▷ Ø • 하 등 등 등 등 등 등 및 1 등 개 및 의 자자 및 다
⊕ - ○ | %p - ≅ ₽ ₽ |
                                                                                                                                                                      main.cpp ⊕ ×
                            · (전역병위)
: // 변경된 문자열을 새 벡터에 추가
                              modified_se_file_names.push_back(modified);
                           cout << "파일 저장중..." << endl;
if (out_file.is_open()) {
                             if (out_file.is_open()) {
  out_file << "init:" << endl;
  // 벡터 배열의 내용을 토대로 정의하기
  for (size_t i = 0; i < modified_se_file_names.size(); ++i) {
    out_file << " define audio." << modified_se_file_names[i] << " = ";
    out_file << "\\"" << original_se_file_paths[i] << "\\"" << endl;
                           out_file.close(); // 파일 스트림 닫기
; cout << "파일 저장 완료: " << se_output_filename << endl;
} else {
                                                                                                                                                                            } catch (const fs::filesystem_error& e) { cerr << e.what() << endl;
                      } else {
cout << "디렉토리가 존재하지 않습니다." << endl;
cout << "메인으로 돌아갑니다." << endl;

    문제가 검색되지 않음

                                                                                                                                                                     에 추가 🔺 🔷 리포지토리 선택 🔺
```

```
ு - ≅ 🖺 📵
               면역 발에 

면에 발에 

EM SCG는 리소스를 저장할 때 반드시 파일명에 언더바가 한번씩 들어가야만 한다. 

// 이는 렌파이 기능의 원활한 수행을 위해 어쩔 수 없다. 

// 언더바가 없는 파일은 배열 포함에 제외되도록 처리되었다. 

EVOId SOGDefine() {
 main.cpp + X
                                                                                                                                                              vector<string> original_scg_file_paths; vector<string> original_scg_file_names;
                                                                                                                                                                    P ++ main.cpp

타 헤더 파일
                    vector<string> modified_scg_file_names;
vector<string> modified_first_file_names;
                     vector<string> modified_second_file_names;
                    ofstream out_file(scg_output_filename);
                    fs::path scg_dir = "./renpy_example/game/images/SCG/"; // 문서 경로 지정 if (fs::exists(scg_dir) &&
                                                                                                                                                               ◆ 출루션 탐색기 Git 변경 내용
                          fs∷is_directory(scg_dir)) // 해당 디렉토리 존재 여부 확인
                      original_scg_file_names.push_back(
entry.path().filename().stem().string());
            ◎ 문제가 검색되지 않음
                                                                                                                                              ↑ 소스 제어에 추가 • ♦ 리포지토리 선택 • 🚨
000 파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 벨드(B) 디버그(D) 테스트(S) 분석(N) 도구(T) 확장(X) 창(W) 도움말(H) 검색((Ctrl+Q)
                          '-| Debug • x64 - ▶요컵 Windows 다비가 • ▷ ② • 하 등 등 등 등 에 함께 및 의 취 기 및
⊕ - ○ | %p - ≅ ₽ ₽ |
                                                                                                                                                          main.cpp ⊕ ×
                         cont 《 파물공을 텐파이 단역 가능 버진으로 진환중...* 《 endl;
for (const auto& str : original_scg_file_names) {
: // 언더바가 없는 파일은 건너뜀
: if (str.find('_') == std::string::npos) {
: cout < str << " - 언더바가 없어 처리되지 않는 파일입니다." << endl;
                            string modified = str;
                                                                                                                                                              ◆ 출루션 탐색기 Git 변경 내용
                                                                                                                                                               // 변경된 파일 제목 문자열을 첫 번째 언더바를 기준으로 절단
for (const auto& str : modified_scq_file_names) {
                                                                                                                                                        에 추가 🔺 🔷 리포지토리 선택 🔺
```

```
160 - ≦ 🖺 📵
 main.cpp ⊕ X
                                                                                                                                            • (전역병위)
size_t underbar_cut = str.find("_"); // 첫 번째 언더바의 위치를 찾음
if (underbar_cut != std::string::npos) {
| modified_first_file_names.push_back(
                           str.substr(0, underbar_cut)); // 첫 번째 부분
modified_second_file_names.push_back(
str.substr(underbar_cut + 1)); // 두 번째 부분
                                                                                                                                                 for (const auto& str : modified_scg_file_names) cout << str << endl;
                      ◆ 출루션 탐색기 Git 변경 내용
                                                                                                                                            i out_file.close(); // 파일 스트림 닫기
cout << "파일 저장 완료: " << scg_output_filename << endl;
} else {
           ↑ 소스 제어에 추가 • ♦ 리포지토리 선택 • 🚨
000 파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 벨드(B) 디버그(D) 테스트(S) 분석(N) 도구(T) 확장(X) 창(W) 도움말(H) 검색((Ctrl+Q)
                       ⊕ - ⊙ | %p - ≅ № № | ¹
                                                                                                                                       main.cpp ⊕ ×
                        . (면역병에)
cerr << "파일을 열 수 없습니다: " << scg_output_filename << endl;
                    } catch (const fs::filesystem_error& e) {
  cerr << e.what() << endl;
}</pre>
                    refse {
    cout << "디렉토리가 존재하지 않습니다." << endl;
    cout << "메인으로 돌아갑니다." << endl;
                  vector<string> original_bg_file_names;
vector<string> modified_bg_file_names;
                  string bg_output_filename = "bg.rpy";
ofstream out_file(bg_output_filename);
                                                                                                                                           fs::path bg_dir = "./renpy_example/game/images/BG/"; // 문서 경로 지정
if (fs::exists(bg_dir) &&
: fs::is_directory(bg_dir)) // 해당 디렉토리 존재 여부 확인
           에 추가 ▲ ♦ 리포지토리 선택 ▲
```

```
७ - ≅ 🖺 🛭
   main.cpp ⊕ ×
                                                                                                                                                                                                                                                                                                                                                                                                                      달부선 8세기

○ ☆ ♬ * 0 - 45 급**

요주선 84개 2 점(cul+3) 요구-
급 송추선 80M (1 프로젝트의

▲ IROM

> oo 청조

> IROM

> po 청조

- IROM

> po box po 
                                                                        if (fs∷is_regular_file(entry)) {
                                                                            original_bg_file_paths.push_back(entry.path().string());
original_bg_file_names.push_back(
                                                                                        entry.path().filename().stem().string());
                                                                                                                                                                                                                                                                                                                                                                                                                                    ▷ ++ main.cpp
등 해더 파일
                                                                 ,
cout << "파일명을 렌파이 인식 가능 버전으로 전환중..." << endl;
                                                                                                                                                                                                                                                                                                                                                                                                                       ◆ 출루션 탐색기 Glt 변경 내용
                                                                        if (c == '
                                                                                                                                                                                                                                                                                                                                                                                                                       for (const auto& str : modified_bg_file_names) cout << str << endl;
                                                                                                                                                                                                                                                                                                                                                                            ↑ 소스 제어에 추가 • ♦ 리포지토리 선택 • 🚨
                                                                                                                                                                                                                                                                                                                                                                                                               로그인 첫 — CF X
                                                                   '- Debug - x64 - ▶ 요설 Windows 디네거 - ▷ 선 - 即 司 - 12-16 1 1 1 1 1 1 日 日 月 月 月 月 月 日
⊕ - ○ | %p - ≅ ₽ ₽ |
                                                                                                                                                                                                                                                                                                                                                                                                          main.cpp ◆ ×
                                                                 if (out_file.is_open()) {
                                                                    )
out_file.close(); // 파일 스트림 닫기
cout << "파일 저장 완료: " << bg_output_filename << endl;
} else {
cerr << "파일을 열 수 없습니다: " << bg_output_filename << endl;
                                                          } catch (const fs::filesystem_error& e) { cerr << e.what() << endl;
                                                                                                                                                                                                                                                                                                                                                                                                                       cout << "디렉토리가 존재하지 않습니다." << endl;
cout << "메인으로 돌아갑니다." << endl;
                                        ● 문제가 검색되지 않음
                                                                                                                                                                                                                                                                                                                                                                                                       에 추가 ▲ ♦ 리포지토리 선택 ▲
```

```
160 - ≦ 🖺 📵
   main.cpp ⊕ X
                                                                                                                                                                                                                                    [// 있습니다.

曰void ScriptChecker() {

| string script_path = "./renpy_example/game/script.rpy"; // 읽을 파일의 경로

| string bgm_string = "play music"; // 찾고자 하는 문자열 위치

| string sound_string = "play sound";

| string bg_string = "scene";

| string scg_string = "show";

| setSstring = used bom_string: // 충복을 받지하는 set 라이브러리 사용
                                                                                                                                                                                                                                           ▷ ++ main.cpp
등 해더 파일
                              set<string> used_bgm_string; // 중복을 방지하는 set 라이브러리 사용
set<string> used_sound_string;
                              set<string> used sca string;
                              ifstream file(script_path);
                                                                                                                                                                                                                                    ◆ 출루션 탐색기 Glt 변경 내용
                        │
│ if (!file.is_open()) {
│ i cerr ≪ "파일을 열 수 없습니다: " ≪ script_path ≪ endl;
                                                                                                                                                                                                                                    while (getline(file, line)) {
   ProcessLine(line, bgm_string, used_bgm_string);
   ProcessLine(line, sound_string, used_sound_string)
                  ProcessLine(line, bg_string, used_bg_string);
ProcessLineForSCG(line, scg_string, used_scg_string);
ভূষ্ণা ধ্ৰথম প্ৰভ
                                                                                                                                                                                                             ↑ 소스 제어에 추가 • ♦ 리포지토리 선택 • 🚨
.
● 마일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 벨드(B) 디버그(D) 테스트(S) 분석(N) 도구(T) 확장(X) 창(W) 도움말(H) 검색(Ctrl+Q)

    ⊕ • ○ | %p • ≅ № № | %

                                                                                                                                                                                                                            main.cpp ⊕ X
                            // 결과 돌덕
PrintSet("BGM", used_bgm_string);
PrintSet("SE", used_sound_string);
PrintSet("BG", used_bg_string);
PrintSet("SCO", used_scg_string);
                       // 외부 함수 처리용. 첫 번째 공백 기준으로 오른쪽은 잘라다 버림.
void ProcessLine(const string& line, const string& search_string,
set<string>& result_set) {
| size_t pos = line.find(search_string);
| if (pos != string::npos) {
                              if (pos != string::npos) {
    string next_string = line.substr(pos + search_string.length());
    stringstream ss(next_string);
                                                                                                                                                                                                                                    ss >> first_word;
if (!first_word.empty()) {
                                 result_set.insert(first_word);
}
                                                                                                                                                                                                                     △ 제어에 추가 🔺 💠 리포지토리 선택 🔺
```

2) 테스트 결과

(1) 사용되지 않은 리소스 식별 기능

- 설명

현재는 사용된 리소스들이 식별된다. 나머지 부분은 전술한 그대로이다.

- 테스트 결과 스크린샷

4. 계획 대비 변경 사항

없음.

5. 프로젝트 일정

리소스 정의 부분은 SCG부분에서 언더바가 없는 파일이 존재할 시 에러를 뱉던 부분을 처리되지 않도록 개선 완료함으로써 완성되었다.

업무	11/3	11/26	12/4	12/23
제안서 작성	완료			
기능1		완료		
기능2			진행 중	
디버그				>