확장성을 고려한 UCG 저작도구의 구조 설계

성연식° 조경은* 엄기현*
동국대학교 게임공학과°, 동국대학교 영상미디어대학 게임멀티미디어공학과*
{sung, cke, khum}@dongguk.edu

The Framework Design for Expansion in UCG Authoring Tool

Yun-Sick Sung° Kyung-Eun Cho* Ky-Hyun Um*
Dept. of Game Engineering, Dongguk University°
Dept. of Game and Multimedia Engineering, Dongguk University*

요 약

UCC(User Created Contents) 사이트가 활성화되면서 제작 가능한 콘텐츠 종류가 다양해졌다. 제작 가능한 콘텐츠 중에서 UCG(User Created Game)는 사용자가 직접 제작하고 공유하는 게임이다. UCG의 제작 방법은 사용자의 숙련도에 따라서 두 가지로 분류해 볼 수 있다. 첫 번째, 저작도구를 사용하는 방법은 게임 제작에 익숙하지 않은 초보 사용자에게 적합하지만 사용방법이 쉬워지면서 기능이 제약적이다. 두 번째, 프로그램 언어를 이용한 개발은 게임 제작에 익숙한 고급 사용자에게 적합하지만 개발기간이 길어진다. 이 논문에서는 스크립트를 사용해서 초보 사용자에게 적합한 저작도구의 기능을 확장하는 방법을 제안한다. 그리고 제안한 저작도구는 프로그램 소스를 생성하기 때문에 고급 사용자는 제공하지 않는 기능만 추가 구현해서 개발 기간을 단축한다. 제안한 방법의 UCG 제작 과정을 검증하기 위해서 게임 제작에 필요한 GUI, 스크립트 생성 기능과 프로그램 소스 생성 기능을 포함하는 프레임워크를 설계한다. 그리고 설계한 게임 저작도구를 구현해서 개발된 게임 제작 사례를 보임으로써 본 연구에서 제안하는 방법을 소개한다.

ABSTRACT

For the activation and wide distribution of User Created Contents (UCC) sites have diversified user-created possibilities. Among these contents, User Created Game (UCG) sites are places where users can create and share their game contents with others. The method of UCG development can be classified into two categories according to the users' level of professionalism. First, the method of using the authoring tools is suggested for those unfamiliar with the development or creation of contents. Although the authoring tool is easy to use, there are many functional limitations. Second, development using program languages is suggested for trained advanced users but has the limitation of a prolonged development period. This paper proposes a new method, generating the script which will expand the current functional limitations entailed behind the authoring tools used by first time and less trained users. In order to verify the proposed method in a real UCG development environment, a framework encompassing GUI, script generating function and program source generating function were constructed.

Keyword : UCC, UCG, Game Authoring Tool, Source Generation, Script

접수일자 : 2008년 09월 18일
일차수정 : 2008년 10월 13일
심사완료 : 2008년 10월 20일
※ 교신저자(Corresponding Author)
조경은, 주소 : 서울시 중구 필동 3가 26 동국대학교(100-715), 전화 : 02)2260-3834, E-mail : cke@dongguk.edu

1. 서 론

유튜브, 엠군, 판도라TV, 그리고 아프리카TV 등 UCC(User Created Contents) 사이트는 동영상을 편집하는 인터페이스와 제작한 동영상을 등록하는 인터페이스를 제공한다[1]. 그래서 사용자는 직접 만든 콘텐츠를 다른 사용자와 쉽게 공유한다. 사용자가 직접 동영상 콘텐츠를 제작하는 UCC 사이트가 활성화되면서 UCC 사이트에서 제작이 가능한 콘텐츠 종류가 다양해진다. 그 중에서 UCG(User Created Game) 사이트는 사용자가 제작한 게임을 다른 사용자와 공유하는 사이트이다.

UCG의 제작 방법은 사용자의 숙련도에 따라서 두 가지로 분류해 볼 수 있다. 첫 번째, 저작도구를 사용한 UCG 제작 방법은 게임 제작 과정을 잘 알지 못하는 초보 사용자에게 적합하다[2]. 그리고 사용자가 기획한 게임을 빠르게 제작할 수 있는 장점이 있다. 하지만 초보 사용자를 위한 저작도구는 사용 방법을 쉽게 하기 위해서 사용자가 선택하는 기능이 제약적이다. 예를 들어, 사용자가 선택 가능한 옵션의 개수를 줄이거나 초보 사용자가 이해하기 어려운 기능은 사용자에게 의사를 물어보지 않고 처리한다. 이와 같이 처리한 방법은 게임의 다양성을 떨어뜨리기 때문에 제작 가능한 게임의 범위를 줄인다.

두 번째는 프로그램 소스를 이용해서 게임을 제작하는 방법이다. 이 방법은 게임을 제작한 경험이 있고 게임 엔진과 모듈을 사용해서 게임 제작이 가능한 고급 사용자에게 적합하다. 고급 사용자는 지형 도구, 스프라이트 도구, 그리고 애니메이션 도구와 같이 반복적인 작업을 줄이기 위해서 사용자에게 맞는 저작도구를 직접 제작하고 생성한 데이터를 게임에 사용한다[3]. 그리고 개발 기간과 비용을 줄이기 위해서 다른 사용자가 개발한 엔진과 모듈을 사용한다. 하지만 게임 제작은 기획, 프로토타입 개발, 개발 그리고 테스트 단계를 거치기 때문에 게임 제작 기간이 길어진다. UCG 사이트에서 기획한 게임을 빠르게 제작하고 다른 사용자와 공유하기 위해서는 프로그램 소스를 빠르게 만드는 방법이 필요하다.

이 논문에서는 UCG 제작에 관심이 있는 사용자의 요구사항을 만족하는 저작도구 설계 방법을 다음과 같이 제안한다. 첫 번째, 게임 제작에 익숙하지 않은 초보 사용자에게 적합한 저작도구의 기능이 제한되는 문제를 스크립트를 사용해서 해결한다. 복잡한 인터페이스는 초보 사용자의 저작도구 사용을 어렵게 한다. 그래서 자주 사용하는 기능만 GUI로 제공하고 자주 사용하지는 않지만 제작하는 게임의 기능을 확장하거나 세밀하게 제어하는 기능은 스크립트로 처리해서 저작도구의 확장성을 높인다. 두 번째, 고급 사용자가 프로그램 소스로 게임을 제작할 때 제작 기간이 길어지는 문제는 저작도구로 프로그램 소스를 생성해서 제작 기간을 단축한다. 저작도구에서 제공하는 기능을 사용해서 게임을 제작하고 제공하지 않는 기능만 프로그램 소스를 사용해서 개발함으로써 제작 기간을 단축할 수 있다.

제안한 방법을 사용한 UCG 제작 과정을 검증하기 위해서 게임 제작에 필요한 GUI, 스크립트 생성 기능 그리고 프로그램 소스 생성 기능을 포함하는 프레임워크를 설계한다. 그리고 설계한 게임 저작도구를 구현해서 게임 제작 사례를 소개함으로써 본 연구에서 제안하는 방법을 실험한다.

이 논문의 구성은 다음과 같다. 2장에서 게임 제작 방법론을 소개한다. 3장에서는 제안하는 저작도구의 구성을 설명한다. 4장에서는 구현한 저작도구의 인터페이스를 소개하고 생성한 결과물을 기술한다. 5장에서는 결론을 내리고 향후 연구 과제를 제시한다.

2. 게임제작 방법론

이 장에서는 게임을 제작하는 방법을 소개한다. 게임 제작 방법을 구분하고 각각의 방법으로 개발할 때 장단점을 알아본다.

게임을 개발하는 방법은 [표 1]과 같이 저작도구

를 이용한 방법, 규칙 언어를 이용한 방법 그리고 프로그램 언어를 사용하는 방법이 있다. 첫 번째, 저작도구를 이용한 방법은 규칙 언어와 프로그램 언어를 이용한 방법에 비해서 제공하는 기능을 쉽게 알 수 있다. 저작도구는 사용자에 따라서 구성을 다르게 한다. 초보 사용자를 위한 저작도구는 다양한 기능보다는 사용자가 쉽게 배우고 사용할 수 있는 인터페이스를 구성한다. 예를 들어, 초보 사용자를 위한 GamearraY는 웹 기반의 인터페이스를 제공해서 사용자가 콘텐츠를 제작하고 관리하기 때문에 게임 디자이너나 다양한 콘텐츠를 가진 사용자가 프로그래머의 도움 없이 자신의 콘텐츠를 이용해서 게임을 제작하고 배포할 수 있다[4]. GamearraY에서는 초보 사용자를 위해서 횡 스크롤 게임, 종 스크롤 게임 등 수정이 가능한 5가지 종류의 게임을 제공한다. 그래서 사용자는 제공하는 게임을 기반으로 원하는 게임을 제작한다. GamearraY는 사용자가 쉽게 게임을 제작할 수 있는 인터페이스를 제공하지만 제공하는 기능이 다양하지 못한 단점이 있다. 고급 사용자를 위한 저작도구는 사용방법보다는 많은 기능을 제공하도록 구성한다. 고급 사용자를 위한 저작 도구 중에서 SimBionic은 에이전트의 행동과 알고리즘을 GUI로 정의한다[5]. 에이전트의 움직임은 SimBionic 비주얼 AI 코드 생성기 플랫폼을 이용해서 C++ 프로그램 소스를 생성한다. SimBionic은 저작도구에서 많은 기능을 제공하기 때문에 처음 사용하는 사용자가 어려움을 느끼고 익숙해지는 데 많은 시간이 걸린다.

두 번째는 규칙 언어를 이용해서 게임을 제작하는 방법이다. 이 방법은 어려운 프로그램 소스를 감추고 복잡한 기능을 쉽게 표현한다. 하지만 사용자가 모든 규칙을 이해하고 사용해야하는 어려움이 발생한다. 규칙 기반 언어를 사용하는 방법 중에 RC++는 소니 컴퓨터 엔터테인먼트가 플레이스테이션2에서 인공지능을 표현하기 위해 정의한 규칙 기반 언어이다[6]. RC++는 인공지능을 표현하는 많은 조건과 상태를 쉽게 처리한다. RC++는 RC++ 컴파일러로 컴파일해서 C 혹은 C++ 프로그램 소스로 만들기 때문에 프로그래머가 기능을 추가 삭제가 가능하다.

[표 1] 게임 제작 방법

종류	상용제품 대상	특징, 장점, 단점
저작 도구	GamearraY 초보	웹 기반 인터페이스 배우기 쉬움, 기능 확장에 제약
	SimBionic 고급	에이전트의 행동과 알고리즘 정의 다양한 UI 제공, 초기 사용에 어려움
규칙 언어	RC++ 초보,고급	인공지능 표현, 조건과 상태를 쉽게 정의 규칙을 모두 이해해야 제작이 가능
프로 그램 언어	DirectX 고급	게임용 SDK 게임 개발에 필요한 모듈 제공 개발기간이 길어짐, 제작이 어려움
	XNA GSE 고급	통합 게임 개발 도구 다양한 모듈과 저작도구를 제공 개발기간이 길어짐, 제작이 어려움

마지막으로 프로그램 언어를 이용한 방법이다. 프로그램 언어는 개발 방법이 어렵기 때문에 고급 사용자에게 적합하다. SDK를 이용한 개발 방법 중에서 가장 대표적인 DirectX는 게임 제작에 필요한 기능을 제어하는 API를 제공한다[7]. 사용자는 통합 개발 도구에서 프로그램 소스와 DirectX를 연결하고 제공하는 API를 호출해서 게임을 제작한다. 그리고 마이크로소프트에서 크로스 플랫폼을 위해서 개발한 XNA 게임 스튜디오 익스프레스(XNA GSE)는 UCG 사이트에서 제공한다[8]. XNA GSE는 게임 제작에 필요한 모듈과 그래픽과 오디오를 편집하는 저작도구를 제공한다. 프로그램 언어를 이용한 게임 제작 방법은 프로그램 언어와 SDK를 모두 이해해야 게임을 제작할 수 있기 때문에 제작에 대한 난이도가 높다. 복잡한 제작 방법은 UCG 사이트의 취지와는 달리 사용자가 쉽게 게임을 제작하고 공유하기 어렵다.

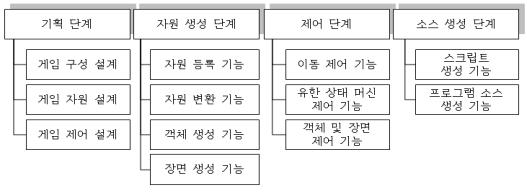
게임을 개발하는 3가지 방법 중에 저작도구를 이용한 제작방법이 게임 제작에 경험이 없는 초보 사용자에게 가장 적합하지만 확장성이 높은 인터페

이스가 필요하다. 프로그램 언어와 규칙 언어를 사용하는 방법은 고급 사용자에게는 편리한 인터페이스를 제공하지만 게임 제작 기간이 길어진다.

이 논문에서는 초보 사용자가 저작도구로 쉽게 게임을 제작하고 스크립트로 기능을 확장하는 구조를 소개한다. 그리고 고급 사용자는 저작도구와 스크립트를 사용해서 게임을 제작하고 생성한 소스로 확장하는 UCG 저작도구의 구조를 제안한다.

3. UCG 제작에 적합한 저작도구

UCG를 제작하는 저작도구는 초보 사용자도 쉽게 게임을 제작하면서 확장성이 높은 인터페이스를 제공해야한다. 그리고 제작한 게임은 스크립트와 프로그램 소스를 생성해서 고급 사용자가 다양한 엔진과 모듈을 사용하도록 한다. 게임 엔진에는 렌더링 엔진, 애니메이션 엔진, 서버엔진 그리고 사운드 엔진 등이 있다[9]. 이 장에서는 게임 제작 과정을 [그림 1]과 같이 기획 단계, 자원 생성 단계, 제어 단계 그리고 소스 생성 단계로 구분하고 각 단계에서 필요한 기능을 기술한다.



[그림 1] UCG 제작에 적합한 저작도구의 구성도

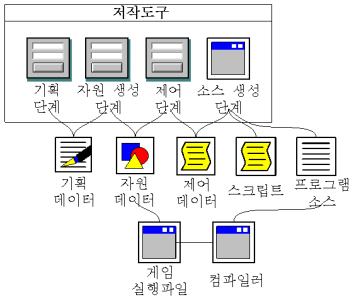
기획 단계는 사용자가 기획자의 입장에서 제작하려는 게임을 기획하는 단계이다. 초보 사용자는 기획하는 내용을 세부적으로 알지 못하기 때문에 게임을 기획할 때 어려움을 느낀다. 그래서 기획을 위한 저작도구는 기획에 필요한 항목을 보기 쉽게 나열하고 각각의 항목을 직관적으로 입력할 수 있는 인터페이스를 제공해야한다. 기획 단계는 내용에 따라서 3가지로 나눈다. 게임 구성 설계는 게임의 틀을 잡고 게임의 기본 방향을 정한다. 예를 들어, 게임 개념, 게임 장르, 게임 시점 그리고 스토리를 사용자가 기술할 수 있는 인터페이스를 제공한다. 게임 자원 설계는 게임에 나타나는 캐릭터, 아이템, 그리고 장면 등의 항목을 정의하고 관리한다. 마지막으로, 게임 제어 설계는 게임 자원 설계에서 정의한 항목이 동작하는 방법을 정의한다.

자원 생성 단계는 게임에 필요한 이미지와 사운드를 등록하고 변환하는 인터페이스를 제공한다. 자원 편집기는 기능에 따라서 4가지로 나눈다. 자원 등록 기능에서는 저작도구에서 사용할 이미지 파일과 사운드 파일을 등록하는 인터페이스를 제공한다. 자원 변환 기능에서는 등록한 이미지와 사운드 파일을 자원 편집기와 컨트롤 편집기에서 사용하는 형태로 변환한다. 객체 생성 기능은 게임에 등장하는 캐릭터와 아이템의 움직임을 등록한 이미지와 사운드로 정의한다. 장면 생성 기능은 배경 이미지를 지정하고 등장하는 객체를 정한다.

제어 단계는 자원 단계에서 구성한 장면과 객체의 이벤트를 처리한다. 예를 들어, 사용자의 입력을 받아 처리하고 게임 상태에 따라서 이벤트를 발행시킨다. 컨트롤 편집기는 3가지 기능을 제공한다. 이동제어 기능은 자원 편집기에서 등록한 객체의 동작을 제어한다. 상태머신 표현 기능은 객체의 내부 상태를 상태머신으로 표현한다. 마지막으로 제어 표현 기능은 이동제어 기능과 상태머신 표현 기능을 연결해서 게임 진행을 구성한다.

마지막으로 소스 생성 단계는 스크립트를 생성하는 기능과 프로그램 소스를 생성하는 기능으로 구분한다. 스크립트를 생성하는 기능에서는 제어 단계에서 저장한 파일을 읽고 사용자가 수정 가능한 스크립트를 생성한다. 저작도구에서 입력한 값을 프로그램 소스로 변환하지 않고 스크립트를 생성하면 다음과 같은 장점이 있다. 프로그램 소스는 초보 사용자가 이해하기 어렵지만 스크립트는 프로그램 소스보다 쉽고 가독성이 높기 때문에 사용자가 결과물을 확인하기 쉽고 수정이 편리하다. 그리고 스크립트는 사용자가 프로그램 언어를 몰라도 저작도구에서 제공하지 않는 기능 설정을 스크립트 문법에 맞추어 원하는 기능으로 수정할 수 있다. 프로그램 소스를 생성하는 기능은 스크립트를 읽어서 프로그램 소스를 생성한다.

자원 생성 단계에서 생성한 자원 데이터와 소스 생성 단계에서 생성한 프로그램 소스는 제안한 저작도구의 최종 결과물이다. 프로그램 소스는 사용자가 통합 개발 도구를 이용해서 실행파일로 만든다. 실행파일은 자원 데이터를 사용해서 게임을 수행한다.



[그림 2] 저작도구의 생성 파일

[그림 2]는 각 단계에서 생성한 데이터와 입력 데이터를 표현한다. 기획 단계에서 입력한 내용은 기획 데이터로 저장한다. 그리고 자원 생성 단계에서 입력한 값은 자원 데이터로 저장한다. 생성한 자원 데이터는 제어 단계와 게임 실행 파일에서 사용한다. 제어 단계는 자원 생성 단계에서 입력한 이미지와 사운드를 사용해서 게임을 제작하고 제어 데이터를 생성한다. 소스 생성 단계에서는 제어 데이터와 스크립트를 입력 받아 스크립트와 프로그램 소스를 생성한다. 마지막으로 컴파일러로 게임 실행 파일을 생성한다.

4. RISE 게임 저작도구 구현

이 장에서는 제안한 방법으로 게임 저작도구를 구현한 RISE(Realtime Interactive Simulation Engineering) 게임 저작도구를 소개한다. RISE 게임 저작도구는 게임 제작에 필요한 4단계를 하나의 저작도구에서 모두 처리한다.

4.1 기획 편집기

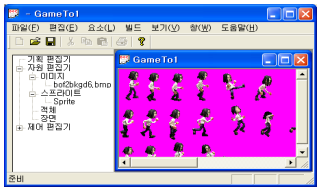
기획 편집기에서는 게임 기획에 필요한 인터페이스를 제공한다. 게임 구성 설계에서는 게임 개념과 게임 구성으로 구분해서 입력하고 게임 방향을 설정한다. 게임 자원 설계는 캐릭터와 아이템을 기획하고 각각의 속성을 입력하는 인터페이스를 제공한다. 게임 제어 설계는 제어, 효과음 그리고 화면 구성으로 나누어 기획한다.

[그림 3] 기획 편집기 화면

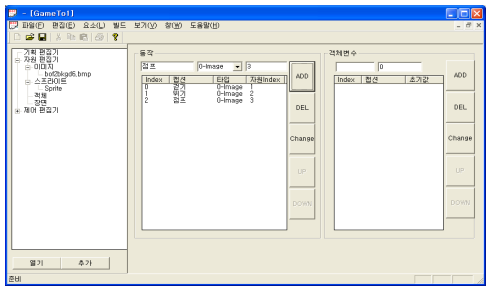
예를 들어, 캐릭터 정보는 [그림 3]과 같이 이름, 종류, 그리고 개념 이미지를 설정한다. 그리고 게임 제작에 사용하는 캐릭터의 속성을 사용자가 지정하고 입력하는 인터페이스를 제공한다.

4.2 자원 편집기

자원 편집기에서는 게임 제작에 필요한 이미지와 사운드를 등록한다. 등록한 이미지와 사운드는 제어 단계에서 사용하는 스프라이트, 객체 그리고 장면을 생성할 때 사용한다.



(a) 이미지 관리



(b) 객체 관리

[그림 4] 자원 편집기 실행 화면

자원 편집기는 이미지와 사운드를 등록하는 자원 등록 기능과 등록한 이미지를 스프라이트로 변환하는 자원 변환 기능을 제공한다. 스프라이트는 [그림 4]의 (a)처럼 등록한 이미지에서 드래그로 역영을 선택해서 스프라이트를 등록한다. 등록한 스프라이트는 객체의 동작과 장면의 배경에서 사용한다. 그리고 객체 생성 기능은 [그림 4]의 (b)와 같이 객체의 동작과 속성을 정의한다. 객체를 제어할 때 필요한 정보를 묶어서 관리를 함으로써 데이터 관리가 용이하다.

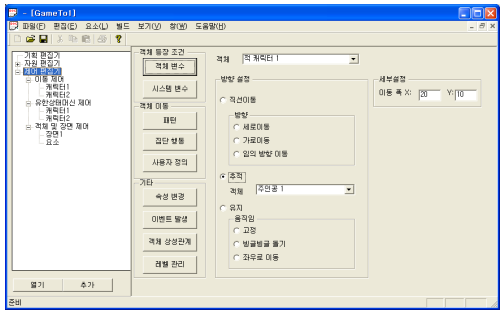
4.3 제어 편집기

제어 편집기는 제어 단계에 필요한 인터페이스를 제공한다. 기획 편집기와 자원 편집기에서 생성한 데이터를 사용해서 기획한 게임을 제작한다. 이 절에서는 RISE 게임 저작도구에서 제공하는 이동 제어, 유한 상태머신 제어 그리고 객체 및 장면 제어 방법을 소개한다. 이동 제어와 유한 상태머신 제어는 객체 관리에서 정의한 속성에 따라서 행동을 수행하고 이벤트를 발생한다. 객체 및 장면 제어는 객체 간의 관계와 장면의 이벤트를 정의한다.

4.3.1 이동 제어

이동 제어에서는 자원 편집기에서 정의한 객체의 이동 경로를 다음과 같이 정의한다. 첫 번째, 객체가 등장하고 이동하는 조건을 설정한다. 조건은 자원 편집기에서 정의한 등장 객체의 속성과 다른 객체의 속성을 사용해서 만든다. 두 번째, 객체의 이동 경로를 정의한다. 이동 경로는 패턴 이동, 집단행동 이동, 그리고 사용자 정의 이동으로 구분한다. 패턴 이동은 반복적인 객체의 움직임을 설정하고 집단행동 이동은 객체가 이동할 때 동일한 객체끼리 뭉쳐서 이동할 때 설정한다. 사용자 정의 이동은 패턴 이동과 집단행동 이동으로 정의가 불가능한 이동을 기술한다. 세 번째, 이동 중인 객체의 속성 값을 비교해서 이벤트를 발생시키고 객체의 속성을 변경한다.

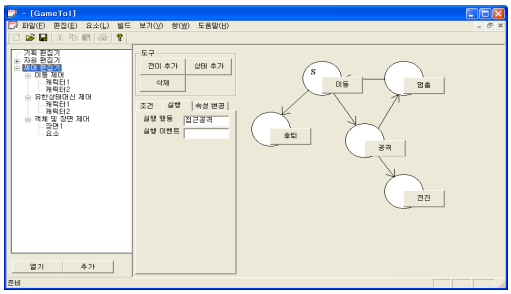
예를 들어, 이동 경로 중에서 패턴 이동은 [그림 5]와 같이 사용자가 한 방향으로 이동하는 직선이동, 설정한 객체를 쫓아가는 추적, 그리고 일정한 범위에 머물러 있는 유지 중에 하나를 선택하고 이동 폭을 입력한다. 패턴 이동을 정의한 객체는 객체 등장 조건에서 정의한 조건에 부합할 때 객체가 등장하고 등장한 객체는 입력한 패턴에 맞추어 움직인다.



[그림 5] 이동 제어 화면

4.3.2 유한 상태머신 제어

유한 상태머신 제어는 객체의 내부 상태를 유한 상태머신으로 정의한다. 객체의 상태는 전이 조건, 실행 그리고 속성 변경 정보를 가진다. 전이 조건은 다른 상태에서 전이가 가능한 조건을 표현한다. 전이가 가능한 상태는 방향이 있는 선으로 연결하고 상태에서 정의한 조건이 부합할 때 전이가 발생한다. 실행은 전이가 발생할 때 수행할 행동과 이벤트를 정의한다. 마지막으로 속성 변경은 상태 전이가 이루어질 때 변경할 객체 속성을 나열하고 변경할 값을 설정한다.

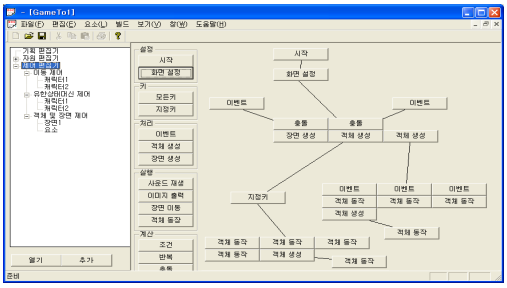


[그림 6] 유한 상태머신 제어 화면

예를 들어, [그림 6]은 이동하고 공격하는 객체의 내부 상태를 표현한다. 객체는 시작 표시가 된 이동 상태에서 상태 전이를 시작한다. 이동 상태에서는 객체의 상태에 따라서 후퇴와 공격 상태로 전이한다. 상태의 전이가 이루어지면 변경한 상태에서 정의한 행동과 이벤트를 실행하고 객체의 속성 값을 변경한다.

4.3.3 객체 및 장면 제어

객체 및 장면 제어는 제어 단계에서 객체간의 관계를 시간의 흐름이나 이벤트에 맞추어 제어하는 인터페이스를 제공한다. 그래서 객체의 움직임은 이동 제어와 유한 상태머신 제어에서 입력하고 입력한 객체 간의 관계는 객체 및 장면 제어에서 설정한다.



[그림 7] 객체 및 장면 제어 화면

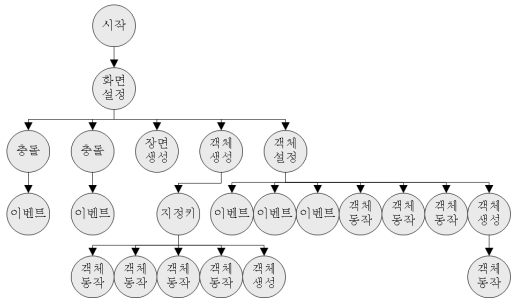
[그림 7]은 RISE 게임 저작 도구의 객체 및 장면 제어에서 게임을 초기화하고 사용자의 입력을 받는 객체와 유한 상태머신으로 움직이는 객체를 정의한 화면이다. [그림 6]에서 설정한 제어는 다음과 같이 3가지 작업으로 구분한다. 첫 번째, 게임을 초기화하는 작업이다. 시작 명령을 이용해서 게임 시작 부분을 정의하고 화면설정 명령을 사용해서 해상도를 정한다. 화면 설정이 끝나면 장면과 등장할 객체를 생성한다. 그리고 충돌 명령을 사용해서 객체끼리 충돌이 일어날 때 발생하는 이벤트를 정의한다. 두 번째, 사용자의 객체 움직임을 설정한다. 생성한 객체 중에서 사용자의 입력을 받는 객체는 지정한 키 입력으로 움직이기 때문에 지정 키 명령을 사용해서 4방향으로 캐릭터를 이동하고 무기를 사용하면 총알 객체를 생성한다. 생성한 객체는 객체 동작 명령을 사용해서 이동 제어와 유한 상태머신 제어에서 설정한 행동으로 이동한다. 세 번째, 유한 상태머신 제어를 사용해서 움직임을 설정한다. 유한 상태머신 제어에서 상태의 전이 조건과 실행 정보를 입력하고 제어에서는 수행한 행동과 발생한 이벤트를 처리한다.

4.4 소스 생성기

이 절에서는 기획 편집기, 자원 편집기 그리고 제어 편집기를 이용해서 입력한 게임 데이터를 가지고 스크립트와 프로그램 소스를 만드는 과정을 소개한다.

4.4.1 스크립트 생성

RISE 게임 저작도구는 프로그램 소스를 생성하기 전에 스크립트를 생성한다. 스크립트는 저작도구에서 제공하지 않는 명령과 옵션을 사용자가 정의해서 사용할 수 있다. 스크립트는 자원 편집기와 제어 편집기에서 입력한 데이터를 읽고 다음과 같이 생성한다. 첫 번째, 자원 편집기에서 입력한 이미지 목록, 스프라이트 목록, 객체 목록, 그리고 장면 목록을 읽어서 스크립트를 생성한다. 1개의 자원 데이터는 1개의 스크립트로 표현한다.



[그림 8] 객체 및 장면 제어의 자료 구조

두 번째, 제어 편집기에서 입력한 이동 목록과 유한 상태머신 목록도 읽어서 객체의 제어 데이터를 1개의 스크립트로 표현한다. 그리고 객체 및 장면 제어는 자원 편집기와 제어 편집기에서 생성한 스크립트와 연동해서 게임의 흐름을 16개의 스크립트로 표현한다.

예를 들어, [그림 7]과 같이 입력한 제어 데이터는 [그림 8]과 같이 순환이 가능한 방향 그래프로 저장하고 시작 명령을 시점으로 너비 우선 탐색을 시작한다. 예를 들어, [그림 8]을 스크립트로 생성할 때 가장 먼저 시작 명령을 읽어서 스크립트를 생성하고 시작 명령에 연결한 화면 설정 명령의 스크립트를 생성한다. 화면 설정 명령에 연결한 다른 명령도 너비 우선 탐색 순서대로 스크립트를 생성한다. 그래프가 순환을 가지기 때문에 탐색하다가 스크립트를 생성한 명령은 더 이상 탐색하지 않는다.

```
Start
SetScreen       640         480     16
Collision       1           2       1
Collision       2           3       2
SetScene        1           1       1
CreateObject    캐릭터1     50      400
CreateObject    캐릭터2     590     400
```

[그림 9] 생성한 스크립트

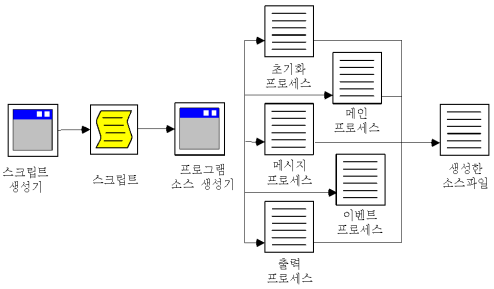
[그림 9]는 시작 명령에서 2번째 객체 설정 명령까지 생성한 결과이다. 생성한 스크립트는 사용자가 수정이 가능하고 프로그램 소스를 생성할 때 사용한다.

4.4.2 프로그램 소스 생성

프로그램 소스는 액션 게임, RPG 게임 등 다양한 장르의 게임에 적용 가능한 범용적인 구조를 가진다. 장르마다 생성한 프로그램 소스의 구조가 달라지면 사용자가 서로 다른 장르의 게임을 제작할 때마다 프로그램 소스를 다시 분석하고 이해해야하는 단점이 발생한다.

프로그램 소스 생성은 구조 소스 생성 단계와 스크립트 소스 생성 단계로 구분한다. 구조 소스 생성은 프로그램을 수행하기 위한 함수를 생성한다. 함수는 [그림 10]과 같이 초기화 프로세스, 메인 프로세스, 메시지 프로세스, 이벤트 프로세스 그리고 출력 프로세스로 기능 단위로 묶어서 처리한다.

초기화 프로세스에서는 컴파일에 필요한 헤더파일을 포함하고 자원 편집기와 제어 편집기에서 정의한 데이터를 저장하기 위한 변수를 정의하고 프로그램 시작할 때 파일에서 데이터를 불러온다. 메시지 프로세스는 사용자의 입력을 받아 객체를 이동하는 작업과 이벤트를 발생시키는 작업을 수행한다. 이벤트 프로세스는 주기적으로 이벤트가 발생하는 지 모니터링하고 조건에 부합할 때 해당 함수를 호출한다. 그리고 객체간의 충돌을 검사해서 충돌이 발생하면 충돌을 처리하는 함수를 호출한다. 출력 프로세스는 자원 편집기에서 등록한 객체의 이미지를 함수 paint안에서 출력한다. 마지막으로 메인 함수는 프로그램을 시작할 때 초기화 프로세스를 호출하고 시스템 메시지가 발생하면 메시지 프로세스에 전달하는 작업을 처리한다. 그래서 주기적으로 출력 프로세스와 이벤트 프로세스를 호출해서 이미지를 출력하고 이벤트를 처리한다.



[그림 10] 프로그램 소스 구조

스크립트 소스 생성 단계는 생성한 스크립트를 하나씩 읽어서 구조 소스 생성 단계에서 생성한 프로그램 소스에 기능을 추가한다. 스크립트는 기능에 따라서 설정 스크립트, 키보드 입력 스크립트, 처리 스크립트, 실행 스크립트 그리고 계산 스크립트로 구분한다. 설정 스크립트는 게임의 실행 환경을 설정한다. 키보드 입력 스크립트는 사용자의 입력을 처리하는 스크립트로 메시지 프로세스에 소스를 생성한다. 처리 스크립트, 실행 스크립트 그리고 계산 스크립트에서 기능 처리는 이벤트 프로세스에 프로그램 소스를 생성하지만 이미지 출력이 필요한 스크립트는 출력 프로세스에도 프로그램 소스를 생성한다. 예를 들어, 스크립트 중에서 이벤트를 처리하는 이벤트 스크립트는 다음 같이 프로그램 소스를 생성한다. 첫 번째, 이벤트 함수를 생성한다. [그림 11]의 (a)와 같이 사용자가 정의한 이벤트는 개수만큼 함수를 정의한다. 두 번째, 발생한 이벤트를 처리한다. 마지막으로 생성한 이벤트 함수를 호출한다. [그림 11]의 (c)는 객체의 충돌이 발생할 때 이벤트를 호출하는 프로그램 소스이다.

```
void Event1() {
}
```

(a) 이벤트 함수 생성

```
void Event1() {
    if(_object[0].Energy>0) _object[0].Energy--;
}
```

(b) 이벤트 구현

```
void Event1() {
    if(_object[0].Energy>0) _object[0].Energy--;
}
void Collision1() {
    if(_object[0].X1 < _object[1].X2
        && _object[0].X2 > _object[1].X1
        && _object[0].Y1 < _object[1].Y2
        && _object[0].Y2 > _object[1].Y1)
        Event1();
}
```

(c) 이벤트 호출

[그림 11] 프로그램 소스 생성

4.5 생성한 게임 소개

[그림 12]는 RISE 게임 저작도구를 이용하여 생성한 프로그램 소스 파일을 컴파일하고 실행한 액션 게임의 화면이다.

객체와 배경을 출력하기 위해서 [그림 13]의 (a)처럼 다양한 동작을 포함하는 이미지와 [그림 13]의 (b)와 같이 장면에 사용할 이미지를 등록한다.

[그림 12] 실행화면

제어 편집기에서는 상태 캐릭터가 임의의 위치에 등장해서 패턴 이동을 하도록 설정한다. 사용자의 캐릭터는 키보드 입력을 받아 동작한다.

이 절에서는 액션 게임을 제작한 사례로 소개했지만 RISE 게임 저작도구를 사용한다면 시뮬레이션, RPG 그리고 슈팅 게임 등 다양한 장르의 게임도 개발이 가능하다.

(a)캐릭터 이미지 (b)장면 이미지

[그림 13] 입력 이미지

5. 결 론

이 논문에서는 초보 사용자부터 고급 사용자까지 모두 사용할 수 있는 게임 저작도구를 제안했다. 저작도구만 사용해서 게임을 제작하고 저작도구에 입력한 값은 스크립트와 프로그램 소스를 생성했다. 초급 사용자를 위한 저작도구에서 제작 가능한 게임의 범위가 제한되는 문제는 스크립트를 사용해서 보완하는 방법을 제안했다. 스크립트를 사용함으로써 인터페이스의 간결성은 유지하면서 GUI에서 제공하지 않는 기능도 제어할 수 있었다. 고급 사용자가 프로그램 소스로 게임을 제작할 때 개발기간이 길어지는 문제는 저작도구로 프로그램 소스를 생성해서 개발 기간을 단축하는 방법을 제안했다. 저작도구로 기획한 게임의 프로토타입을 빠르게 제작하고 다른 사용자와 공유해서 게임이 가지고 있는 문제점을 미리 진단할 수 있었다. 프로토타입을 완성하면 게임 저작도구로 생성한 스크립트와 프로그램 소스를 이용해서 개발 기간을 단축하고 다양한 엔진과 모듈을 연동해서 게임을 완성도를 높일 수 있었다.

게임 제작 방법은 기획 단계, 자원 생성 단계, 제어 단계 그리고 소스 생성 단계로 나누고 각 단계에서 필요한 항목을 기술했고 기획 단계에서는 개요부터 게임에 등장하는 캐릭터, 아이템 등 기획에 필요한 항목을 설명했다. 자원 생성 단계에서는 게임 제작에 필요한 사운드와 이미지를 등록하고 제어 단계에서 필요한 형태로 변환하는 과정을 소개했다. 제어 단계는 기획한 게임을 제작하는 기능을 소개하고 마지막으로 소스 생성 과정을 기술했다.

구현에서는 RISE 게임 도구의 사용자 인터페이스를 각 단계에 맞추어 소개했다. 기획 편집기에서 기획을 위한 인터페이스를 설명했다. 자원 편집기에서 이미지를 등록하고 객체와 장면을 생성하는 인터페이스를 소개했다. 제어 편집기에서는 기능을 이동 제어, 유한 상태머신 제어 그리고 객체 및 장면 제어로 나누고 각각의 게임 제작 과정을 설명했다. 그리고 소스 생성기에서는 생성한 스크립트와 프로그램 소스를 설명하고 프로그램 소스 구조를 기술했다.

향후에는 효율적인 스크립트 설계와 사용자 관점에서 가독성을 높이고 수정이 용이한 스크립트 제작 방법에 관한 연구가 이루어져야 한다.

참고문헌

[1] 안성혜, 송수미, "UCC 서비스의 현황 및 발전전망", 한국콘텐츠학회 2007 추계 종합학술대회 논문집, 제5권, 제2호(하), pp. 691-697, 2007.

[2] Mark Green, "Towards Virtual Environment Authoring Tools for Content Developers", Proceedings of the ACM symposium on Virtual reality software and technology, pp. 117-123, 2003.

[3] 김종찬, "윈도우 게임 프로그래머를 위한 X파일", 사이버출판사, 2000 .

[4] http://www.gamearray.com

[5] Robert Richards, Rihard Stottler, Ben Ball, and Coskun Tasoluk, "Intelligent Track Analysis on Navy Platforms Using Soft Computing", Lecture Notes in Computer Science, Vol. 3849, pp. 195-204, 2006.

[6] Ian Wright and James Marshall, "RC++ : A Rule-Based Language For Game AI", Proceedings GameOn 2000: 1st International Conference on Intelligent Games & Simulation, 2000.

[7] Frank D. Luna, "Introduction 3D Game programming with DirectX 9.0", WORDWARE publishing, 2003.

[8] Benjamin Nitschke, "Professional XNA Game Programming: For Xbox 360 and Windows", Wrox publishing, 2007.

[9] 이헌주, 박태준, 김현빈, "온라인 3D 게임엔진 개발에 관한 연구", 한국게임학회 논문지, 제3권, 제2호, pp. 42-55, 2003.

성연식(Yun-Sick Sung)

2004년 2월 부산대학교 컴퓨터공학 학사
2006년 2월 동국대학교 일반대학원 컴퓨터공학 석사
관심분야 : 다중 에이전트 시스템, 학습, 전문가시스템

조경은(Kyung-Eun Cho)

1993년 2월 동국대학교, 전자계산학과(공학사)
1995년 2월 동국대학교, 컴퓨터공학과 대학원(공학석사)
2001년 8월 동국대학교, 컴퓨터공학과 대학원(공학박사)
2003년 9월～2005년 8월 동국대학교 정보산업대학 컴퓨터멀티미디어공학과 전임강사
2005년 9월～현재 동국대학교 영상미디어대학 게임멀티미디어공학과 조교수
관심분야 : 컴퓨터 게임 알고리즘, 게임 인공지능

엄기현(Ky-Hyun Um)

1975년 2월 서울대학교 공과대학 응용수학과 공학사
1977년 2월 한국과학기술원 전산학과 이학석사
1994년 2월 서울대학교 대학원 컴퓨터공학과 공학박사
1978년 3월～2007년 6월 동국대학교 컴퓨터멀티미디어공학 교수
2007년 7월～현재 동국대학교 영상미디어대학 게임멀티미디어공학과 교수
1995년 3월～1999년 2월 동국대학교 정보관리처장 역임
2001년 3월～2003년 2월 동국대학교 정보산업대학 학장 역임
2005년 3월～현재 한국 게임학회 자문위원
1998년 12월～2001년 12월 한국 멀티미디어학회 부회장, 자문위원, 수석부회장 역임
2007년 1월～2007년 12월 한국멀티미디어학회 회장
관심분야 : 게임시스템 및 구조 설계, 멀티미디어응용시스템, 멀티미디어데이터베이스