

스크립트를 활용한 아바타 콘텐츠 제작 시스템

김지훈¹ · 황대원¹ · 서경진^{2*}

¹(주)인공지능연구원, 기술연구소 캐릭터시팀 엔지니어

^{2*}(주)인공지능연구원 기술연구소 캐릭터시팀 리더

Avatar content creation system using script

Ji-Hoon Kim¹ · Dae-Won Hwang¹ · Kyoung-Chin Seo^{2*}

¹Engineer, Character AI Team, Artificial Intelligence Research Institute, Seongnam 13558, Korea

^{2*}Leader, Character AI Team, Artificial Intelligence Research Institute, Seongnam 13558, Korea

[요약]

COVID-19로 인해 비대면 서비스에 대한 요구가 높아졌고 그에 따라 3D 아바타의 애니메이션을 활용한 콘텐츠에 대한 수요도 높아졌다. 그러나 3D 애니메이션 콘텐츠는 높은 비용이 들어 쉽게 제작할 수 없는 상황이다. 본 논문에서는 3D 아바타를 활용한 콘텐츠를 제작할 수 있는 스크립트 기반 애니메이션 시스템을 제안한다. 텍스트로부터 음성 및 발음 기호 생성, 대사에 맞는 애니메이션 할당, 콘텐츠의 시나리오 정리를 위한 도구들을 개발했다. 스크립트를 통해 다양한 애니메이션을 조합할 수 있도록 신체 부위별로 짧은 동작 단위로 제작한 애니메이션 세트를 구축했다. 정의한 제스처 단계에 따라 대사에 애니메이션을 할당한 스크립트를 제작하여 스크립트의 지시대로 말하며 동작하는 아바타 콘텐츠를 제작할 수 있는 시스템을 개발했다. 최종적으로 사용자와 상호작용 가능한 아바타 콘텐츠 데모를 통해 적은 비용과 노력으로 아바타 콘텐츠를 제작할 수 있음을 증명하였다.

[Abstract]

Due to COVID-19, the demand for non-face-to-face services has increased, and accordingly, the demand for content using animations of 3D avatars has also increased. However, 3D animation content is expensive and cannot be easily produced. In this paper, we propose a script-based animation system that can create contents using 3D avatars. We develop tools for generating voice and lip sync data from text, assigning animation clips to match lines, and organizing content scenarios. In order to combine various animations through scripts, we built an animation set made in short motion units for each body part. We developed a system that can produce avatar content that speaks and operates according to the script's instructions by creating a script that assigns animations to lines according to the defined gesture steps. We show that avatar content can be produced efficiently through our demonstration that 3D avatar can interact with users.

색인어 : 아바타, 스크립트, 애니메이션, 립싱크, 캐릭터

Keyword : Avatar, Script, Animation, Lip Sync, Character

<http://dx.doi.org/10.9728/dcs.2022.23.1.11>



This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Received 26 November 2021; **Revised** 27 December 2021

Accepted 27 December 2021

***Corresponding Author; Kyoung-Chin Seo**

Tel: [REDACTED]

E-mail: kcseo@airi.kr

I. 서론

COVID-19으로 인해 비대면 서비스에 대한 수요가 늘어남에 따라[1] 사람과 대면하지 않으면서도 사람과 상호작용을 하는 느낌을 줄 수 있는 3D 아바타 콘텐츠에 대한 수요도 늘어나고 있다. 그러나 3D 아바타 애니메이션 콘텐츠를 제작하는 것은 많은 시간과 비용이 들어간다.[2]

이러한 문제를 해결하기 위해 본 논문에서는 대사와 동작을 지정하는 스크립트를 제작하여 3D 아바타 콘텐츠를 제작할 수 있는 시스템을 제안한다. 본 시스템에서는 여러 도구들을 활용해 아바타의 대사, 애니메이션을 생성하고 콘텐츠의 시나리오를 편집하여 최종적으로는 사용자와 상호작용할 수 있는 아바타 콘텐츠를 생산한다.

시스템의 입력부에 텍스트를 입력하면 TTS(Text to Speech) 서비스인 Amazon polly[3]로부터 음성 파일과 입모양 정보[4]에 대한 파일이 생성된다. 이 데이터는 캐릭터의 입모양에 대한 음성과 입모양 애니메이션 동작[5]-[7]으로 활용된다. 대사 생성에 활용할 애니메이션 세트는 사전에 신체 부위와 제스처의 강도에 따라 분류하여 DB를 구성한다. 생성된 대사의 입모양 정보를 통해 대사 내에서 음절이 발화되는 타이밍 정보를 활용하여 대사의 타이밍에 어울리는 애니메이션을 할당할 수 있는 도구를 제공한다. 생성된 대사 중 콘텐츠에 활용할 내용과 순서를 정리하는 도구를 통해 콘텐츠의 시나리오를 정한 뒤 3D 아바타 클라이언트를 통해 콘텐츠를 제공할 수 있도록 시스템을 구성하였다.

본 논문에서 제안하는 스크립트를 활용한 아바타 시스템을 통해 빠른 시간과 저비용으로 3D 아바타를 활용한 콘텐츠를 제작할 수 있으며 콘텐츠의 변경 및 추가에 대한 대응도 간편하게 할 수 있을 것으로 기대한다.

II. 배경지식

2-1 언리얼 엔진[8]

언리얼 엔진은 에픽 게임즈사에서 제공하는 게임엔진으로 게임 제작 뿐만 아니라 3D 애니메이션 콘텐츠 제작, 건축 시각화, 시뮬레이션, 방송 등 다양한 3D 콘텐츠 제작에 사용된다. C++ 및 '블루 프린트' 라는 비주얼 스크립트 언어를 사용한다. Windows, Mac, Linux 환경에서 개발이 가능하며 엔진의 소스코드를 전체 공개하여 사용자가 원하는대로 엔진의 기능을 변경하여 사용할 수 있는 장점이 있다. 총 수입 \$1,000,000 이상이 될 시 5%의 로열티를 지불해야 하며 수입이 그 이하일 경우 무료로 사용할 수 있다. 멀티 플랫폼 빌드를 지원하기 때문에 Windows, Max, Linux와 같은 PC외에도 Android, ios와 같은 모바일 기기에서 동작하는 콘텐츠를 개발할 수 있다.

2-2 아마존 폴리

아마존 폴리는 아마존에서 제공하는 TTS 서비스로 텍스트를 음성 및 입 모양 파일로 변환한다. 29개의 언어를 제공하며 한국어 음성은 여성 음성 하나를 제공하고 있다. 입 모양 파일은 marks 형식의 파일로 문장을 단어 단위로 잘라 각 단어의 시작 및 끝 시간을 ms 단위의 정보로 제공하며 해당 문장의 발음 기호의 종류와 타이밍 정보를 제공한다.

III. 대사, 동작 기반 스크립트를 활용한 애니메이션 제작 시스템

본 논문에서 제안하는 시스템은 사용자가 대사와 대사에 해당하는 스크립트를 제작해 캐릭터 콘텐츠를 빠르게 제작할 수 있는 시스템이다. 시스템은 그림 1과 같이 4개의 모듈로 구성했다. 먼저 시스템 입력부에 텍스트가 입력되면 대사 생성기를 통해 대사에 해당하는 음성, 발음기호 파일을 생성한다. 스크립트 편집 도구에서 그 정보를 활용해 아바타의 동작을 제어하는 스크립트를 제작한다. 시나리오 편집 도구에서 실제 서비스에서 사용하는 대사와 대사들의 카테고리를 지정한다. 마지막으로 아바타 클라이언트는 앞의 모듈들에서 생성된 결과물들을 활용해 사용자와 상호작용 하는 아바타 콘텐츠를 제공하는 역할을 한다.

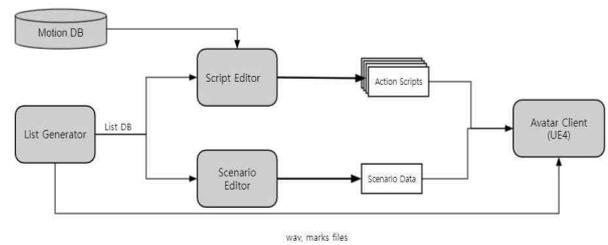


그림 1. 아바타 콘텐츠 제작 시스템의 구조
Fig. 1. Structure of avatar content creation system

3-1 대사 생성기

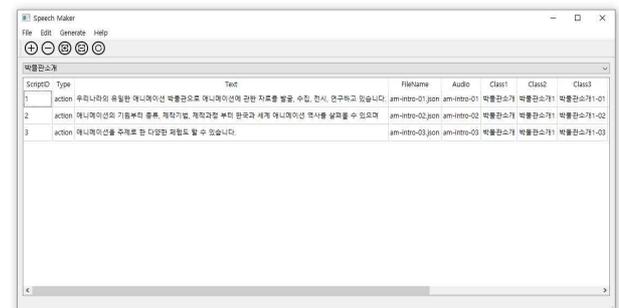


그림 2. 대사 생성기
Fig. 2. Speech data maker

스크립트는 기본적으로 아바타가 전달할 대사와 스피치 음성 정보에 기초한 타임 스탬프 정보가 필요하다. 이를 위해 대사 리스트를 작성하고 TTS를 활용한 스피치 오디오 데이터와 입모양 정보(Viseme) 데이터를 통합적으로 생성 관리하는 도구가 필요하다.

대사 생성기는 필요한 대사를 작성, 편집하고 대사 그룹별 관리 기능을 포함하며 Amazon Polly API를 이용하여 스피치 오디오 파일(wav 포맷)과 함께 타임스탬프 정보와 함께 입모양 정보를 생성하고 다운 받는 기능으로 구성되어 있다.

대사 생성기는 그림 2와 같이 파이썬 Qt기반의 GUI로 구성되어 사용자가 손쉽게 아바타가 전달해야 하는 대사와 립싱크 애니메이션 정보를 관리하는 툴 형태로 제공한다.

3-2 애니메이션 데이터 베이스 설계

1) Motion DB 구조

표 1. Motion DB 속성
Table 1. Motion DB properties

Column	Description
FileName	Animation file name
DisplayName	Animation file name (korean)
Path	File Directory
Class1Label	Large category(korean)
Class2Label	Medium category (korean)
Class3Label	Small category(korean)
Class1	Large category
Class2	Medium category
Class3	Small category
Controller	Controller type
Gesture_Type	Motion/Pose
Blend_Type	Abs/Rel
Periodic	Periodic/Non-Periodic
Body_Parts	Body parts of animation
Duration	duration(ms)

Motion DB란 캐릭터의 신체 부위와 동작의 종류에 따라 세분화된 애니메이션 세트를 구성한 데이터베이스다. 이 데이터베이스에 다양한 제스처 동작과 감정 표현이 포함되도록 일상적인 제스처 뿐만 아니라 연기 이론에서 분류하고 있는 동작들을 참고하였다[9]-[12]. 사용자는 이 DB를 활용하여 애니메이션을 다양한 방식으로 조합하여 콘텐츠를 제작할 수 있다[13]-[19].

DB 에 포함된 애니메이션 세트는 많은 가짓수, 클립들의 짧은 재생 시간이라는 특징을 갖고 있다. 현재 구성된 애니메이션 세트는 애니메이션 클립의 가짓수는 약 800개이며 약 1 초 내외의 재생 시간을 가지고 있다.

애니메이션 세트 내에는 일반적인 애니메이션 클립뿐만 아니라 단일 프레임의 포즈 클립과 동작의 정도에 변화를 주기 위해 Additive 애니메이션으로 활용하기 위한 클립도 존재한다.

사용자가 다양한 애니메이션들을 쉽고 효율적으로 활용할 수 있도록, 또한 시스템 내부에서도 애니메이션 클립들을 통제하는 규칙에 활용하기 위해 'Motion DB'를 SQLite[20]로 구성했다.

2) Motion DB의 속성 정보

Motion DB는 표1과 같이 애니메이션 클립의 파일의 이름 정보(FileName, DisplayName)와 경로(Path), 동작에 따른 분류에 대한 정보(Class1~3 Label, Class1~3)와 각 애니메이션에 대한 속성 정보(Controller, Gesture_Type, Blend_Type, Periodic, Body_Parts) 그리고 시간 정보(Duration)로 구성되어 있다.

이 중에서 동작에 따른 분류 정보(Class1~3)는 3단계로 나누어져 있다. 대분류(Class1)는 애니메이션이 전신(FullBody), 상체(UpperBody), 손(Hand)과 같이 신체의 어떤 부분을 사용하는지에 대한 분류이다. 중분류(Class2)는 동작의 종류를 나타내며 숨쉬기, 상체 젖히기, 머리 끄덕이기, 양손 주먹 쥐기 등이 있다. 소분류(Class3)는 동작의 정도로 나뉜다. 동작의 정도가 없는 애니메이션은 소분류 속성은 가지지 않는다.

애니메이션에 대한 속성 정보 중 컨트롤러(Controller)는 해당 애니메이션을 재생하기 위해 아바타 클라이언트에서 실행되는 애니메이션 컨트롤러 모듈을 지칭한다.

Blend_Type은 애니메이션을 어떻게 활용할 것인가를 정한다. Abs는 애니메이션을 그대로 기존 자세에 블렌딩 하여 사용하고 Rel은 Additive Animation으로 블렌딩 하여 사용한다. Gesture_Type은 애니메이션 클립이 단일 프레임으로 이루어진 클립이라면 Pose, 움직이는 애니메이션이라면 Motion으로 표시한다. Periodic은 애니메이션의 첫 프레임과 마지막 프레임이 같은 자세를 취하고 있다면 Periodic 아니라면 Non-Periodic이다. Body_Parts는 애니메이션의 움직임과 연관된 신체 부위들을 나열한다. 시간정보(Duration)은 애니메이션의 재생 시간 정보이다.

3-3 스크립트 편집 도구

스크립트 편집 도구는 Unreal Engine 4로 제작해 윈도우 PC에서 동작한다. 대사 정보에 맞는 애니메이션의 할당, 조합, 전환 등을 지정할 수 있는 도구이다. 사용자가 대사의 구간 마다 원하는 캐릭터 애니메이션을 할당하면 대사와 애니메이션의 종류, 타이밍 등의 정보가 포함된 json 파일인 '액션 스크립트'를 생성한다.

스크립트 편집 도구의 UI는 그림3과 같다. UI 중앙의 표는 대사에 애니메이션을 배치하는 공간으로 가로축은 대사를 음절 단위로 나뉘었다. 세로축은 캐릭터의 부위별 애니메이션 할당을 위한 컨트롤러들이 있다. 컨트롤러 1개당 두 줄을 사용하며 윗줄은 애니메이션의 단계를 표시하고 아랫줄에는 애니메이션의 이름을 표시한다.

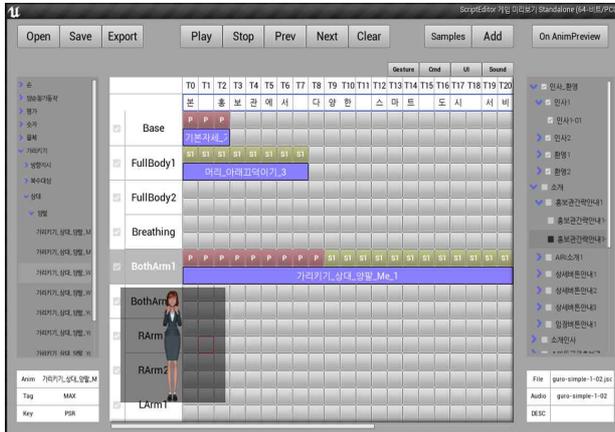


그림 3. 스크립트 편집 도구 UI
Fig. 3. Script editor UI

좌측에는 Motion DB에 포함된 애니메이션의 목록이 애니메이션의 Class 분류에 따라 계층 구조로 표시한다. 오른쪽에는 대사 DB에 포함된 대사 정보가 카테고리 분류에 따라 계층 구조로 표시한다.

사용자는 우측 대사 정보 리스트로부터 대사를 선택 후 중앙 표에서 사용할 컨트롤러를 선택 후 왼쪽의 애니메이션 목록에서 사용할 애니메이션을 선택하여 애니메이션을 재생할길 원하는 부분만큼을 할당하는 방식으로 액션 스크립트를 제작하게 된다.

1)액션 스크립트

액션 스크립트는 문장 단위의 대사의 음성 파일 경로 할당된 애니메이션과 애니메이션 재생에 사용될 타이밍 정보를 담고 있다. 이를 활용해 캐릭터 클라이언트는 대사를 하며 립싱크와 애니메이션을 할 수 있게 된다.

그림 4와 같이 액션 스크립트는 대사 정보, 제스처 스케줄 (gesture schedule), 오디오 스케줄(audio schedule), 시간 정보(duration), 작성 날짜(date) 항목으로 구성되어 있다. 이 중 제스처 스케줄은 사용자가 지정한 애니메이션과 연관된 컨트롤러 종류, 애니메이션 속성, 시간 정보를 포함한다. 이 정보들을 토대로 아바타 클라이언트에서는 애니메이션 클립 선택과 애니메이션 재생 속도, 그리고 각 컨트롤러의 모션 합성 가중치에 대한 매개변수인 Blend weight[21]-[23]를 조절한다.

액션 스크립트의 총 소요 시간은 오디오 파일의 길이 보다 짧을 수 없다. 사용자가 액션 스크립트를 제작할 때 음성이 끝난 이후에도 뒤에 임의로 칸을 추가하여 애니메이션을 음성보다 더 길게 재생하도록 한 경우에는 총 소요 시간이 오디오 시간보다 길어질 수 있다.

gesture_schedule에 적힌 애니메이션의 duration에 맞게 애니메이션이 재생될 수 있도록 지정된 애니메이션의 속도와 컨트롤러의 Blend weight를 조절한다.



그림 4. 액션 스크립트 예시
Fig. 4. Example of action script

2) 애니메이션 구동 단계

본 논문의 시스템에서는 애니메이션을 제스처 단위를 구성하는 단계에 따라 준비(Preparation), 동작(Stroke), 복귀(Relax)의 3단계로 구분하여 각 애니메이션의 끝맺음과 메인 동작이 자연스럽게 연결되도록 설계하였다[24]-[26]. Preparation은 Stroke를 하기 위해 준비하는 단계로 애니메이션을 재생시킬 준비를 한다. Preparation 단계에 있는 Animation Controller는 현재 포즈에서 애니메이션의 첫 번째 프레임으로 애니메이션을 블렌딩 한다. Stroke는 제스처가 표현되는 단계이며 애니메이션을 재생한다. Relax는 적용된 애니메이션을 기본 자세로 되돌려 놓는 단계로 Animation Controller의 Blend weight를 0으로 만들어 해당 컨트롤러를 최종 포즈에 적용하지 않는 작업을 한다. 애니메이션의 Blend_Type 속성에 따라 Absolute와 Relative로 나뉘는데 Absolute는 애니메이션을 그대로 재생하여 제스처를 표현하는 역할을 하고 Relative는 Absolute 애니메이션에 강도를 추가하거나 흔들림 등을 추가하는 애니메이션이다. Absolute속성의 애니메이션은 Stroke를 하기 위해서 Preparation 단계가 반드시 필요하지만 Relative 애니메이션은 Preparation 단계 없이 Stroke를 통해 애니메이션을 적용하고 Relax하여 멈출 수 있다.

3) 대사-애니메이션 타이밍 정보 동기화

대사와 애니메이션 타이밍 정보를 동기화하기 위한 과정을 두 단계로 구성되어있다. 먼저 대사에서 문자 단위로 제스처의 단계들을 논리적 시간(Logical Time)에 할당하여 동기를 맞춘다. 그 다음 논리적 시간을 미리 구한 입모양 데이터를 활용하여 물리적 시간(Physical Time)으로 변환 한다.

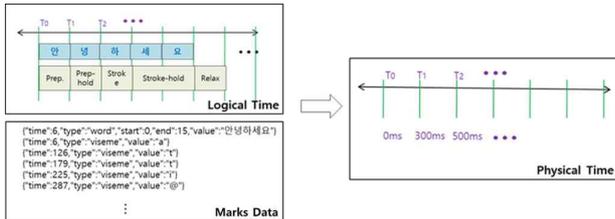


그림 5. 입모양 데이터를 이용하여 논리적 시간에서 물리적 시간으로 변환

Fig. 5. Converting logical time to physical time using rip sync data

3-4 시나리오 편집 도구

시나리오 편집 도구는 콘텐츠의 추가나 변경 시 쉽게 대응할 수 있게 한다. 실제 사용할 카테고리 와 액션 스크립트를 선별하고 콘텐츠의 제공 순서를 변경할 수 있는 기능이 있다. 대사 생성기에서 생성한 대사 DB를 통해 액션 스크립트와 카테고리의 목록을 불러올 수 있으며 UI 테마 선택과 콘텐츠가 버튼 형식의 UI를 통해 사용자의 입력에 반응하는 모드로 사용될지, 정해진 대사를 시나리오의 순서대로 반복 재생하는 모드로 사용될지 정할 수 있다.

3-5 아바타 클라이언트

1) 애니메이션 컨트롤러 구조

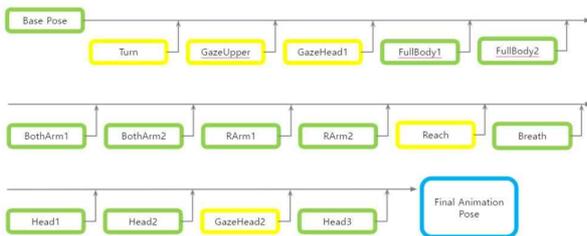


그림 6. 애니메이션 컨트롤러의 순서
Fig. 6. Animation controller order

아바타 클라이언트는 그림 6처럼 부위별로 동작을 생성하기 위한 애니메이션 컨트롤러들로 구성되어 있다. 본 논문에서는[13]-[16]에서 제안한 컨트롤러 연결방식을 기반으로 부위별로 컨트롤러를 뒀으로써 다양한 동작 생성을 가능하게 한다. 컨트롤러는 매 프레임 마다 바로 앞의 컨트롤러에서 만

들어진 포즈에 현재 컨트롤러에서 재생하는 애니메이션의 포즈를 블렌딩 하여 다음 컨트롤러에 넘겨준다. 이로 인해 최종 포즈는 모든 컨트롤러의 애니메이션들이 더해진 결과가 되며 더 나중에 연결되는 컨트롤러의 애니메이션이 앞의 컨트롤러들에 의해 만들어진 포즈와 동일한 부위를 움직이는 경우 이전 컨트롤러의 움직임을 덮어쓰는 성격을 가진다. 또한 컨트롤러들은 Preparation과 Relax단계에서 애니메이션을 점차 적용 시키거나 점차 적용 시키지 않는 작업을 한다. 이를 활용하여 이전 컨트롤러의 동작이 점차 드러나거나 덮어지면서 자연스럽게 제스처의 전환이 이뤄진다.

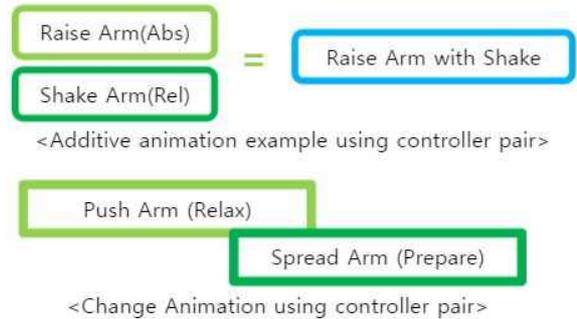


그림 7. 컨트롤러쌍 활용 예시
Fig. 7. Using controller pair example

각 컨트롤러들은 동시에 하나의 애니메이션만을 재생하며 같은 부위의 컨트롤러들은 2개 1쌍으로 구성했다. 이는 컨트롤러가 동시에 1개의 애니메이션만을 재생하는 규칙을 지키면서 자연스러운 애니메이션 전환 및 Additive 애니메이션을 추가할 수 있게 한다. 그림 7을 보면 오른팔 컨트롤러 1에서 팔을 드는 애니메이션을 재생하고 동시에 오른팔 컨트롤러 2에서 팔을 흔드는 Additive 애니메이션을 재생하면 팔을 흔들면서 드는 애니메이션을 생성할 수 있다. 컨트롤러가 2개이기 때문에 그림 5처럼 팔을 내밀던 동작에서 팔을 벌리는 동작으로 자연스럽게 전환할 수 있다.

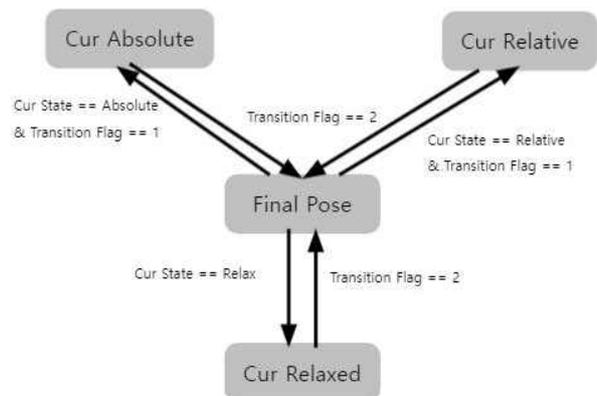


그림 8. 컨트롤러의 내부 상태 기계 구조
Fig. 8. State machine structure of controller

컨트롤러들은 State machine 구조로 동작한다. 기본적으로는 Final Pose State 상태이며 Preparation, Stroke의 동작을 수행 할 때 이 전에 수행했던 동작의 종류에 따라 다른 State로 넘어가 동작을 수행하고 다시 Final Pose State로 돌아온다. 각 State 내부는 이전 컨트롤러의 Pose를 받아와 현재 컨트롤러에서 재생할 애니메이션을 적용하는 구조로 되어있으며 Final Pose는 각 State의 동작이 끝났을 때의 Pose를 재생한다.

2) 클라이언트 UI

클라이언트는 시나리오 에디터에서 설정한 속성 정보에 따라 interactive mode와 Video mode 중 하나로 실행된다. Interactive mode는 시나리오 에디터에서 설정한 카테고리들이 화면에 버튼 형식으로 표시되고 사용자가 버튼을 누르면 카테고리에 해당하는 액션 스크립트들을 재생한다. Video mode는 버튼 UI가 표시되지 않고 시나리오 에디터에서 설정한 카테고리들이 순서대로 반복해서 계속 재생되는 방식이다.



그림 9. 아바타 클라이언트 UI의 3가지 테마
Fig. 9. Three themes of Avatar client UI

Interactive mode의 UI는 그림 9처럼 3가지 테마 중 선택할 수 있다. 공통적으로 시나리오 에디터에서 설정한 카테고리가 버튼 형식으로 화면에 일부 표시되며 화면 하단 배너에 전체 카테고리 목록이 배치된다. 화면 상단에는 현재 시간 및 시나리오의 이름을 표시한다.

Video Mode는 Interactive mode에서 버튼과 하단 배너가 빠진 상태로 Interactive mode와 마찬가지로 3가지 테마로 실행할 수 있다.

IV. 프로그램 시연

4-1 개발 환경

Motion DB는 SQLite를 활용했다. 아바타 클라이언트와 스크립트 에디터는 i9-9900KF 프로세서, 32GB의 메모리,

GeForce RTX 2080Ti의 그래픽 카드를 사용한 하드웨어 환경에서 개발하였다. 개발에 사용한 소프트웨어로는 Epic Games사의 Unreal Engine 4의 4.22 버전을 활용하였다. 대사 생성기는 Python3.7 환경에서 PyQt5 라이브러리를 활용해 개발했다.

4-2 콘텐츠 제작 프로세스 소개

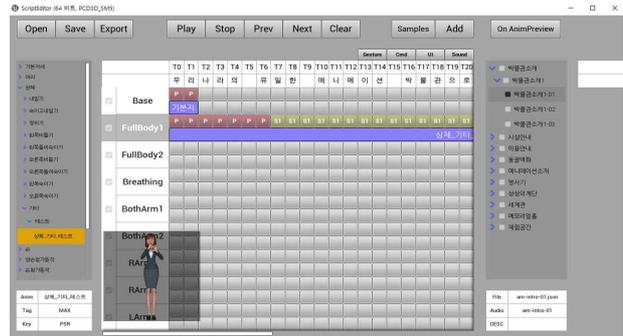


그림 10. 대사에 맞는 애니메이션 할당
Fig. 10. Assign animations to lines

콘텐츠를 제작하기 위해선 우선 아바타가 해야 할 대사를 정해야 한다. 대사 생성기에 원하는 대사와 대사의 파일 이름, 대사를 분류 할 카테고리를 입력하고 생성 버튼을 누른다. 지정된 파일 이름으로 대사의 음성, 입 모양 파일이 생성되며 대사, 파일 이름, 카테고리 정보가 담긴 Scriptlist.db 파일이 생성된다.

대사 생성기를 통해 생성한 파일들과 ScriptList.db 파일들을 스크립트 편집 도구의 'data'폴더 안으로 옮긴 뒤 스크립트 편집 도구를 실행해 대사에 알맞은 애니메이션을 할당 한다.

대사에 애니메이션을 할당하는 작업이 끝나면 Export 버튼을 통해 액션스크립트들을 생성한다. 이 때 사용할 애니메이션만 로드하여 아바타 클라이언트 시작 시 소요시간을 낮추기 위해 대사에 사용된 애니메이션 목록 정보가 담긴 UsedAnimation.json 파일도 함께 생성된다.

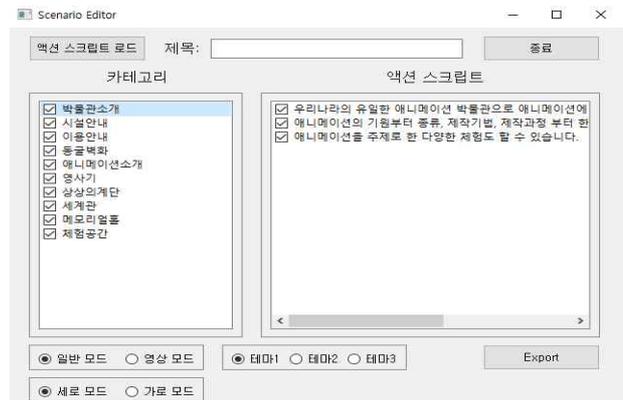


그림 11. 시나리오 편집도구 실행 예시
Fig. 11. Use of scenario editor example

시나리오 편집도구에서는 대사 생성기에서 생성된 ScriptList.db를 활용한다. 시나리오 에디터에서 왼쪽 상단의 액션 스크립트 로드 버튼을 통해 ScriptList.db 파일을 불러오면 ScriptList.db의 테이블 정보가 왼쪽의 카테고리 탭으로, 해당 테이블에 포함된 대사정보가 오른쪽의 액션 스크립트 탭으로 표시된다.

시나리오 에디터에서 실제 아바타 콘텐츠에 사용할 카테고리 and 액션 스크립트를 선택한다. 콘텐츠의 재생 방식과 배경 테마 3가지 중 하나를 선택해 Export하여 Category.json 파일을 생성한다.

UsedAnimation.json, Category.json, 액션 스크립트, wav, marks 파일들을 아바타 클라이언트의 Data 폴더에 위치시킨 뒤 클라이언트를 실행하면 콘텐츠 개발 프로세스가 완료 된다.

4-3 시스템 데모 시연

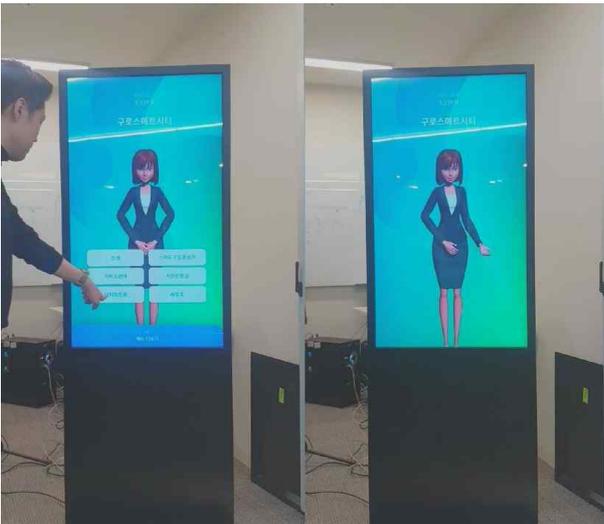


그림 12. 데모 시연 장면
Fig. 12. Demonstration scene

아바타 콘텐츠 제작 시스템을 활용하여 신도림역 지하 1층에 위치한 ‘스마트 구로 홍보관’을 소개하는 데모 콘텐츠를 제작하였다. 총 116개의 액션 스크립트를 제작하여 구로 스마트 홍보관의 전시물들에 대한 소개 콘텐츠를 제작하였으며 약 6분 내외의 분량으로 제작됐다. i5-6700k cpu, 16GB 메모리, Geforce GTX 1060 그래픽의 Windows PC 하드웨어 환경에서 시연하였으며 터치 인터페이스가 탑재된 FHD 해상도의 디스플레이를 사용했다.

대사 생성부터 액션 스크립트 및 시나리오 편집까지 콘텐츠를 개발하는데 애니메이션 제작 작업을 해본 적 없는 개발자 3명이 작업하여 이들이 소요됐다. 스크립트 제작은 한 문장당 약 5분 내외의 시간이 걸렸다. 전문 애니메이션 업체에 의뢰하여 후보작 작업까지 거친 애니메이션을 제작하는 경우 2

달 이상이 소요될 분량이었으며 5분 분량의 3D 애니메이션 콘텐츠 제작에 1000만원 이상의 비용이 필요한 것을 고려하면 비록 애니메이션의 자연스러움은 부족하지만 스크립트를 제작하는 방식으로 전문가의 도움 없이 콘텐츠를 제작할 수 있다는 점에서 제작 시 필요한 비용과 시간을 상당히 줄일 수 있는 결과로 보인다.

시연 결과 애니메이션 블렌딩 및 대사에 따른 입 모양과 동작의 속도는 자연스러웠지만 제스처 단계의 전환 시 약간 경직되는 느낌을 받았다[27]. 액션 스크립트 제작 오류로 동작이 Relax되지 않고 남아있는 문제가 3건 발견되었고 해당 액션스크립트만 수정하여 문제를 해결하는데 약 10분이 소요됐다. 전문 애니메이터의 도움 없이 116문장으로 이루어진 콘텐츠를 제작할 수 있었던 것은 고무적인 성과로 보인다.

V. 결론 및 향후연구

본 논문에서는 대사와 동작 기반의 스크립트를 활용한 애니메이션 제작 시스템에 대해 다뤘다. 애니메이션 세트를 1초 내외로 짧게 나누어 아바타의 신체 부위마다 애니메이션을 할당하여 전신 애니메이션을 만드는 방식으로 인해 더 다양한 애니메이션을 만들 수 있었다. 대사를 음절 단위의 칸으로 나누어 제스처의 동작 단계에 따라 애니메이션을 할당하는 도구를 제공해 편리하게 애니메이션을 제작할 수 있도록 했다. 시나리오 편집 도구를 통해 콘텐츠의 변동이나 추가가 있을 때 대응하기 쉬웠다. 전문 애니메이터의 작업을 요구하지 않고 시스템에서 제공하는 도구들을 활용해 상호작용 가능한 3D 아바타 애니메이션 콘텐츠를 제작할 수 있다는 의의가 있다.

모든 애니메이션이 1초 내외로 나뉘어 있기 때문에 애니메이션들이 약간은 딱딱하게 보이며 액션 스크립트 제작에 소요되는 다소 어렵다는 문제점이 있었다. 따라서 기본 동작들의 애니메이션을 조금 더 길게 만들어 애니메이션 전환의 회수를 줄일 수 있다면 제스처가 부드러워지고 액션 스크립트를 제작할 때도 조금 더 쉬워질 수 있을 것을 기대 한다.

추후 스크린 위에 웹캠을 부착하고 인체 영역을 추출하는 알고리즘을 탑재한 모듈을 추가해 사용자를 향해 고개를 돌리거나 눈을 맞추는 등의 상호작용을 추가하면 사용자가 아바타와 상호작용할 때 더 사실감을 느낄 수 있을 것을 기대 한다.

참고문헌

- [1] Nam-seok Baek, and Su-jung Lee, For the digital transformation of a company, Trend of non-face-to-face software, National IT Industry Promotion Agency(NIPA), Daejeon, Issue report 2020-13, 2020.
- [2] Ministry of Culture, Sports and Tourism, Animation Industry Promotion Basic Plan, 2021.

- [3] Amazon. Amazon Polly[Internet], Available: <https://aws.amazon.com/ko/polly/>.
- [4] Amazon. Amazon Polly Viseme Tables[Internet]. Available: <https://docs.aws.amazon.com/polly/latest/dg/ph-table-english-us.html>.
- [5] Xu, Yuyu, et al. "A practical and configurable lip sync method for games," *Proceedings of Motion on Games*, Dublin. pp. 131-140, November, 2013. <https://doi.org/10.1145/2522628.2522904>
- [6] Cao, Yong, et al. "Real-time speech motion synthesis from recorded motions," *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, Goslar. pp. 345-353, 2004. <https://doi.org/10.1145/1028523.1028570>
- [7] Deng, Zhigang, et al. "Animating blendshape faces by cross-mapping motion capture data," *Proceedings of the 2006 symposium on Interactive 3D graphics and games*, California, pp. 43-38, March, 2006. <https://doi.org/10.1145/1111411.1111419>
- [8] Epic Games. Unreal Engine 4. Available: <https://www.unrealengine.com/>.
- [9] Adam Kendon, "Gesture", *Pagijong Press*, pp. 126, 2013.
- [10] Lee Nam-Kook, "Animation acting theory : aniactinory" *Taehakwon*, 2009
- [11] Thiebaut, Marcus, Brent Lance, and Stacy Marsella. "Real-time expressive gaze animation for virtual humans," *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, Budapest, pp. 321-328, May, 2009. <http://dx.doi.org/10.1145/1558013.1558057>
- [12] Zordan, Victor B., et al. "Breathe easy: Model and control of human respiration for computer animation." *Graphical models*, Vol. 68, No. 2, March, 2006. pp. 113-132, <https://doi.org/10.1016/j.gmod.2005.03.005>
- [13] SmartBody. Available: <http://smartbody.ict.usc.edu/>.
- [14] M. Kallmann, S. Marsella, "Hierarchical Motion Controllers for Real-Time Autonomous Virtual Humans," *International Workshop on Intelligent Virtual Agents*, Kos, pp. 253-265, September, 2005. https://doi.org/10.1007/11550617_22
- [15] A. W. Feng, Y. Xu, A. Shapiro, "An Example-Based Motion Synthesis Technique for Locomotion and Object Manipulation," *Symposium on Interactive 3D Graphics and Games*, California, pp. 95-102, March, 2012. <https://doi.org/10.1145/2159616.2159632>
- [16] Feng, Andrew, et al. "Automating the transfer of a generic set of behaviors onto a virtual character," *International Conference on Motion in games*, Berlin, Heidelberg, pp. 134-145, November, 2012. https://doi.org/10.1007/978-3-642-34710-8_13
- [17] Shapiro, Ari. "Building a character animation system," *International conference on motion in games*, Berlin, Heidelberg, pp 98-109, November, 2011. https://doi.org/10.1007/978-3-642-25090-3_9
- [18] Lance, Brent, and Stacy Marsella. "Glances, glares, and glowering: how should a virtual human express emotion through gaze?," *Autonomous Agents and Multi-Agent Systems*, Vol. 20, No. 1, pp. 50-69, May, 2010. <https://doi.org/10.1007/s10458-009-9097-6>
- [19] Sqlite, Available: <https://www.sqlite.org/index.html>.
- [20] Rose III, Charles F., Peter-Pike J. Sloan, and Michael F. Cohen. "Artist-directed inverse-kinematics using radial basis function interpolation," *Computer Graphics Forum*, Vol. 20, No. 3, pp 239-250, September, 2001. <https://doi.org/10.1111/1467-8659.00516>
- [21] Kovar, Lucas, and Michael Gleicher. "Automated extraction and parameterization of motions in large data sets," *ACM Transactions on Graphics*, Vol. 23, No. 3, pp 559-568, August, 2004. <https://doi.org/10.1145/1015706.1015760>
- [22] Huang, Yazhou, and Marcelo Kallmann. "Motion parameterization with inverse blending," *International Conference on Motion in Games*, Berlin, Heidelberg, pp 242 – 253, November, 2010. https://doi.org/10.1007/978-3-642-16958-8_23
- [23] Hyun Kang Kim, Jae Young Song, and Yu Ri Shin, "A Methodological Study of Discourse Material Analysis and Description," *Korean Literature and Linguistics Association*, Vol. 53, pp. 85–106, December, 2010. <http://doi.org/10.22784/eomun.2010.53.85>
- [24] Marsella, Stacy, et al. "Virtual character performance from speech," *Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. California, pp. 25-35, July, 2013. <https://doi.org/10.1145/2485895.2485900>
- [25] Neff, Michael, et al. "Gesture modeling and animation based on a probabilistic re-creation of speaker style," *ACM Transactions on Graphics*, Vol. 27, No. 1, pp. 1-24, March, 2008. <https://doi.org/10.1145/1330511.1330516>
- [26] Demo Video, Available: <https://youtu.be/mUhbDXfTtI>.



김지훈(Ji-Hoon Kim)

2018년 : 아주대학교 대학원 (미디어학석사)

2018년~현 재: (주)인공지능연구원, 기술연구소 캐릭터AI팀 엔지니어

※ 관심분야 : 게임 디자인(Game Design), 디지털 휴먼(Digital Humane), 애니메이션 리타겟팅(Animation Retargeting) 등



황대원(Dae-Won Hwang)

2018년 : 한양대학교 대학원 (공학석사)

2018년~현 재: (주)인공지능연구원, 기술연구소 캐릭터AI팀 엔지니어

※ 관심분야 : 영상처리(Computer Vision), 인체 자세 추정(Human Pose estimation), 제스처 생성(Gesture generation) 등



서경진(Kyoung-Chin Seo)

2001년 : 서강대학교 대학원 (공학석사)

2011년 : 서강대학교 영상대학원 (미디어공학박사)

2011년~2012년: (주)애니프레임 미디어아트팀 팀장

2012년~2018년: 네이버 커넥트재단(구. NHN NEXT) 교수

2018년~현 재: (주)인공지능연구원 기술연구소 캐릭터AI팀 리더

※ 관심분야 : 컴퓨터 비전(Computer Vision), 디지털 휴먼(Digital Humane), 인간컴퓨터상호작용(Human Computer Interaction), 인체 자세 추정(Body Pose Estimation)