

# Document-View 패턴을 활용한 안드로이드 게임 프레임워크에 관한 연구

최 영 미<sup>\*</sup>

## Study on Android Game Framework Using Document-View Pattern (DVAGF)

Youngmee Choi<sup>\*</sup>

### ABSTRACT

Analyze the Android game framework that can be apply to the Application development of new areas and suggest Document-View pattern of a design pattern, design that this applies the Android framework(DVAGF) for novice Game programming learner. Case of Snake game development through DVAGF is supporting modularity of the code, reusability and expandability that seems that the learners can easily understand the game programming basic structure. In addition, the Android framework (DVAGF) seems to provide a basis that can be easily extended to 2D and 3D games.

**Key words:** Android Game Framework, Design Pattern, Document-View Pattern

### 1. 서 론

초보 게임 프로그래밍 학습자에게 게임 개발은 어렵다. 게임 개발이 어려운 이유는 원하는 게임 코드를 작성하기에 앞서 익혀야할 정보의 양이 방대하기 때문이다. 그 외에도 게임개발 초보자들은 코드를 작성하기에 앞서 게임설계부터 해야 한다는 사실을 간과하여 수없이 많은 게임 개발 프로젝트가 명확한 게임 설계 없이 개발돼 끝내 빛을 못 보고 데모 수준에서 끝나고 만다. 그 원인은 이에 대한 명확한 게임 설계의 표준 방식이 없다는 것이다. 안드로이드는 게임 플랫폼으로써 충분한 장점을 가지고 있다. 그러나 다양한 버전이 공존하는 플랫폼의 파편화 및 표준화된 안드로이드 전용의 게임 개발은 다양하고 신속한 게임 개발에 부족함이 있다[3].

안드로이드 게임 프로그래밍 수업현장에서 대부

분의 학습자들이 게임 개발을 어떻게 시작해야 하는지 막막해 하는 경우가 많다. 그리고 수업이 팀 프로젝트를 진행하는데 협업이 원활하게 이루어지지 않고 특정 학생 몇 명만으로 프로그래밍이 이루어지는 경우가 많다. 이러한 초기 게임개발과 협업의 어려움을 해결하고 개발한 게임을 2D 또는 3D로 계속 확장해나가며 게임의 완성도를 높이기 위한 방법이 필요하다. 이러한 필요성에 부응하기 위하여 본 논문에서는 디자인패턴을 활용한 안드로이드 기반 게임 프레임워크(DVAGF:Android Game Framework Using Document-View Pattern)를 제시한다.

본 논문의 구성은 다음과 같다. 2장에서 Document-View 패턴을 정의했으며 3장에는 먼저 게임 개발에 필요한 공통적 기능인 4가지 컴포넌트와 GUI 환경을 제공하는 안드로이드 게임 프레임워크(AGF)를 설명한 후, AGF에 Document-View 패턴을 적용한

※ Corresponding Author : Youngmee Choi, Address: (14097) Sungkyuldaehak-ro 53, Manan-gu, Anyang, Korea, TEL : +82-31-467-8164, FAX : +82-31-467-8067, E-mail : choiym@sungkyul.ac.kr

Receipt date : Jan. 28, 2016, Revision date : Feb. 15, 2016  
Approval date : Mar. 7, 2016

<sup>\*</sup> Div. of Multimedia Eng., Collage of Engineering, Sungkyul University

DVAGF를 설계하였다. 4장에는 적용 사례인 Snake 게임개발을 통한 DVAGF가 지원하는 코드의 모듈화와 재사용 및 확장성으로 학습자가 효율적이고 효과적인 게임프로그래밍의 기본 구조를 쉽게 이해할 수 있음을 보인다. 마지막으로 5장에는 결론을 기술한다.

## 2. Document-View 패턴

통상적인 애플리케이션 개발 환경에서는 사용자에게 보여지는 화면과 이벤트 핸들링이 밀접하게 얽혀있기 때문에 개발 과정에서 이 둘을 완벽하게 분리하기는 쉽지 않다. Document-View 패턴은 이를 위해 MVC 패턴에서 사용자에게 보여지는 화면과 이벤트 핸들링 영역을 구조적으로 단순화시킨 MVC 패턴의 여러 파생 패턴 중 하나이다[4]. Document-View 패턴은 MVC 패턴에서 파생되었기 때문에 패턴의 영역별 역할이 비슷하다.

Fig. 1은 MVC 패턴과 Document-View 패턴을 비교한 것이다. Document-View 패턴의 Document는 MVC 패턴의 Model 영역과 동일한 내용을 나타내고 Document-View 패턴의 View는 MVC 패턴의 View와 Controller의 역할을 통합해 놓은 것이다. MVC 패턴과 Document-View 패턴의 차이는 패턴의 영역 구분이 세 개의에서 두 개로 축소되었다는 것과 영역의 축소로 인해 MVC 패턴에 있던 영역들 간의 간접관계가 줄었다는 것이다. Document-View 패턴은 MVC 패턴과 마찬가지로 Document와 View가 느슨한 결합이 되어야 한다[4]. Document-View 패턴의 영역별 기능은 다음과 같다.

- Document : 데이터와 데이터 처리 로직을 가지고 있다. MVC 패턴의 Model과 동일한 역할을 수행

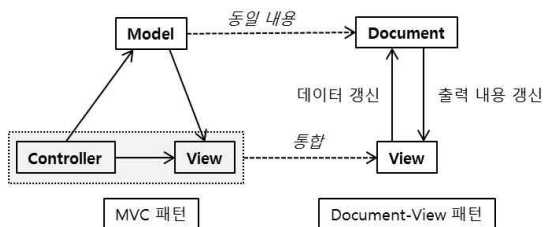


Fig. 1. Comparison of MVC pattern and Document-View pattern.

한다.

- View : 사용자의 입력정보를 받으며 사용자에게 보이는 화면 요소를 그린다. MVC 패턴에서 언급되는 View와 Controller의 역할을 동시에 수행한다.

Document-View 패턴을 사용하는 이유는 Document-View 패턴이 원형인 MVC 패턴과 같은 장점을 갖고 있고 MVC 패턴에 비해 구조가 단순하여 디자인 패턴에 익숙하지 않은 학습자들이 사용하기에 편리하다는 장점이 있다. MVC 패턴에서는 Model, View, Controller라는 세 가지 영역의 기능별로 코드를 구분하여 코딩을 해야 한다. 특히 View와 Controller영역을 명확히 구분하여 프로그래밍하는 것은 디자인 패턴에 익숙하지 않은 학습자들에게 많은 어려움을 준다. 하지만 Document-View 패턴은 Document 영역에서 MVC 패턴의 View와 Controller 영역의 기능을 수행하기 때문에 사용자에게 보여지는 화면과 이벤트 핸들링 영역의 코드를 명확히 구분하지 않아도 된다. 이러한 특징 때문에 디자인 패턴에 익숙하지 않은 학습자들이 사용하기에 Document-View 패턴이 좋다. 단, Document-View 패턴에서는 MVC 패턴의 Controller의 교환 가능성을 희생해야 한다[4].

## 3. DVAGF(Android Game Framework Using Document-View Pattern) 설계

### 3.1 안드로이드 게임 프레임워크(AGF: Android Game Framework)

Fig. 2는 Document-View 패턴으로 안드로이드 용 게임을 제작하기 위해 설계한 안드로이드 게임 프레임워크(AGF)의 클래스 다이어그램이다. AGF는 게임을 제작하기 위한 컴포넌트들인 FileIO, Graphics, Audio, Input을 갖고 있다. 각각의 컴포넌트들이 제공하는 기능을 보면 FileIO는 게임에서 사용될 사용자 정보를 저장하고 불러오는데 사용되는 인터페이스, Graphics는 Canvas API를 사용하는 인터페이스, Audio는 배경음과 효과음을 제어하는 인터페이스, Input은 사용자 입력을 제어하는 인터페이스를 제공한다. FileIO, Graphics, Audio, Input 컴포넌트들은 대부분의 게임에서 공통적으로 사용하는 인터페이스이기 때문에 프레임워크 안에서 구현해

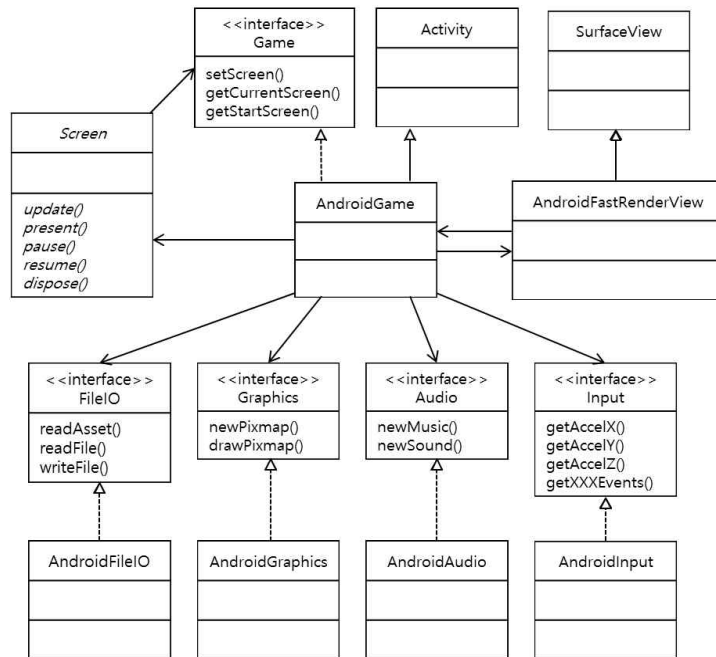


Fig. 2. Class diagram of AGF.

놓고 필요에 사용할 수 있도록 한다.

특히 게임에서 GUI 환경개발은 게임의 완성도를 나타내는 중요한 부분인데 안드로이드 GUI 환경개발에서는 자주 발생하는 공통적인 ANR(Application Not Responding)에러를 해결하기 위해 `SurfaceView`를 사용한다. `SurfaceView`는 별도의 스레드를 자동으로 생성하고 사용하기 때문에 편리하며 GUI 업데이트를 `SurfaceView` 스레드에서 처리해 ANR 에러를 예방할 수 있다. 또한 2D, 3D 그래픽을 사용하기 위해 OpenGL ES에서 제공하는 `GLSurfaceView`는 OpenGL ES을 통한 가속도 지원하므로 차후 AGF를 `GLSurfaceView`로 확장함으로써 2D 또는 3D 그래픽도 표현할 수 있다[3]. 이런 GUI 환경은 게임 개발에 공통적으로 적용 가능한 항목으로 AGF 안에 `AndroidFastRenderView` 클래스로 설계하였다. 그리고 실제 `SurfaceView`에 세팅되어 사용자에게 보이는 `Screen` 클래스는 `Screen` 클래스의 `update()`와 `present()`메서드에 일반적인 게임프레임워크에서 사용되는 Update와 Render 로직으로 각각 구현하면 된다.

따라서 AGF를 적용하여 게임을 구현 할 때는 `FileIO`, `Graphics`, `Audio`, `Input` 컴포넌트와 GUI 환경

을 위한 `AndroidFastRenderView`를 객체참조(Object Reference)하는 `AndroidGame` 클래스와 사용자에게 보여 질 게임화면을 제작하기 위한 `Screen` 클래스를 각각 상속하는 두 개의 클래스만을 구현하면 된다. 이러한 AGF의 `AndroidGame` 클래스와 `Screen` 클래스만을 상속받음으로서 게임 개발에 필요한 기본 항목들을 제공받아 쉽게 개발을 할 수 있다[7].

### 3.2 DVAGF(Android Game Framework Using Document-View Pattern)

DVAGF는 초기 게임개발과 협업의 어려움을 해결하고 개발한 게임을 2D 또는 3D로 계속 확장해 나가며 게임의 완성도를 높이기 위한 방법으로 고안한 디자인패턴을 활용한 안드로이드 기반 게임 프레임워크(DVAGF:Android Game Framework Using Document-View Pattern)이다. 수업현장에서 학습자들이 DVAGF에 기반하여 게임을 개발 하면 학습자들의 게임 접근성이 높고 Document-View 패턴의 장점인 확장성과 유지보수성을 가질 수 있다. 또한 안드로이드 게임 프레임워크를 통해 학생들의 게임 개발 접근성, 개발시간 단축, 협업능률을 높여줄 수 있다.

Fig. 3은 앞서 설계한 안드로이드 게임 프레임워크(AGF)를 사용하고 Document-View 패턴을 적용하여 일반적인 게임을 구현할 경우의 주요 클래스 다이어그램이다. GeneratedGame 클래스는 상위 클래스의 생성자를 호출함으로써 안드로이드 프로그램의 기본인 Activity와 GUI 환경을 위한 Surface-View를 생성한다. GeneratedGame 클래스에서 생성한 Activity의 생명주기를 바탕으로 GeneratedGame 클래스는 getStartScreen() 메서드로 GeneratedGameScreen 클래스를 객체참조 한다. 사용자에게 보여 질 게임화면과 컨트롤을 담당하는 GeneratedGameScreen 클래스는 Screen 클래스를 상속받아 구현되고 Document-View 패턴의 View 영역을 담당한다. GeneratedGame 클래스로부터 생성된 SurfaceView 클래스의 SurfaceView 스프레드는 매 프레임마다 GeneratedGameScreen 클래스의 update()와 present()메서드를 호출한다. update에서는 getXXXEvents() 메서드를 사용하여 사용자 입력(컨트롤)을

처리하고 present()에서는 Document 영역의 데이터 상태 정보를 받아 사용자에게 보이는 게임화면을 구현한다.

Document 영역은 게임의 모든 데이터 상태와 데이터의 로직을 처리하고 View영역에서 제공하는 인터페이스를 바탕으로 데이터 상태를 갱신한다. 기본적인 게임에서는 주인공 캐릭터와 적 캐릭터 그리고 아이템 데이터 클래스를 설계하고 이들 데이터의 로직과 객체들을 이용한 전체 맵을 구현하는 클래스로 Document영역을 구현한다.

개발자가 프레임워크 외에 추가하는 개발 코딩에서 Document-View 패턴을 적용하면 디자인 패턴을 적용하지 않을 때보다 게임 설계와 구현이 쉽다. Document-View 패턴은 데이터(Document)를 게임 화면-컨트롤(View)과 분리하면 똑같은 데이터를 사용하는 전혀 다른 디자인 형태를 빠르게 추가할 수 있어 확장성을 높일 수 있다. 즉 참조하는 데이터의 상태 변화에 따라 다른 결과를 화면에 출력할 수 있

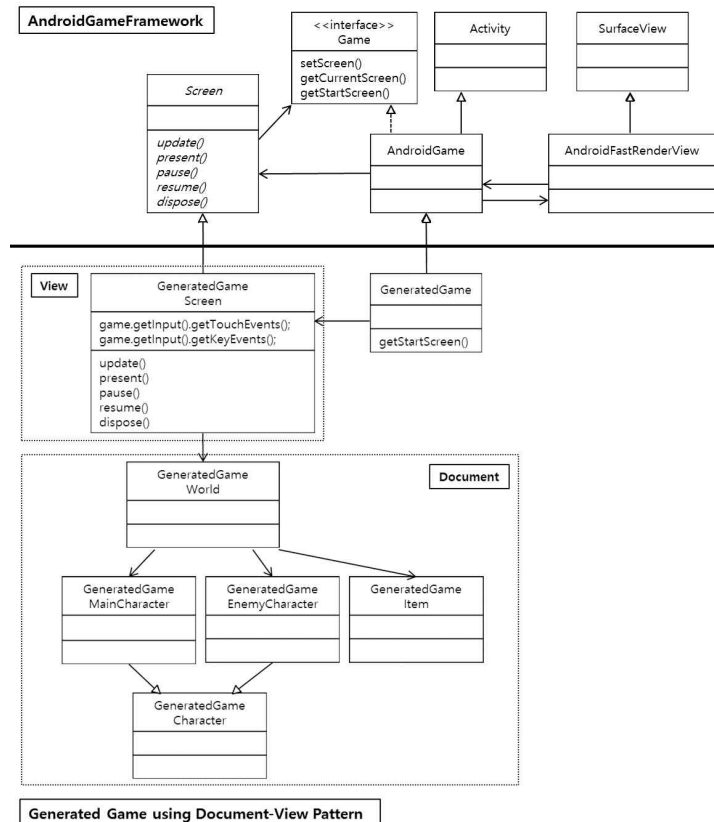


Fig. 3. Class diagram of DVAGF for game development case.

고, 서로 느슨한 결합을 통해 외부로부터 요구되는 구조 변경에도 융통성 있게 대응할 수 있어 유지보수성도 높일 수 있다.

#### 4. DVAGF 적용사례

DVAGF를 통해 효과적으로 게임프로그래밍의 기본 구조를 이해하고 2D, 3D 게임으로 확장도 쉽게 할 수 있는 기본 틀을 제공함을 Snake게임 개발을 적용사례로 하여 다음과 같이 단계적으로 보인다. Snake 게임은 주인공인 뱀이 적을 피해 먹이를 먹으며 꼬리를 키워나가 높은 점수를 획득하는 게임이다.

##### 4.1 데이터 설계(Document 영역)

Snake 게임에서 Document-View 패턴의 Document 영역인 게임데이터를 설계해보자. 전체 게임 맵, 주인공 캐릭터, 적 캐릭터, 먹이로 게임 데이터를 구분할 수 있다. 여기서 Document-View 패턴의 게임데이터(Document)와 게임화면-컨트롤러(View)를 명확히 구분하여 설계하면 캐릭터 혹은 아이템의 추가, 게임 맵 추가, GUI 확장 등 전체적인 게임의 확장성, 재사용성, 유지보수성을 얻을 수 있다. 특히 데이터설계(Document)는 기본 틀을 설계하는 중요한 부분으로 View와의 관계를 생각하며 명확히 설계하는 것이 중요하다.

##### 4.2 게임화면과 컨트롤러 설계(View 영역)

데이터 설계를 마치면 해당 데이터를 가지고 어떻게 화면에 출력해 줄지 결정하면 된다. 비트맵(Canvas API), 2D, 3D그래픽(OpenGL ES) 중 원하는 방법으로 화면설계를 한다. Snake의 초기 구현은 비트맵(Canvas)을 사용하기 때문에 AGF의 Graphics 컴포넌트를 사용할 것이다. 2D, 3D 사용은 차후에 추가 확장할 GLGraphics 컴포넌트를 사용하면 된다.

화면설계와 이벤트 핸들링은 밀접하게 얽혀있기 때문에 화면설계와 같이 설계하는 것이 컨트롤러 설계이다. Snake 게임에서는 AGF가 제공하는 Input 컴포넌트를 통해 AccelerationEvent, TouchEvent, KeyEvent 중 터치이벤트만을 사용한다. 구현하고자 하는 게임의 UI에 따라 Input 컴포넌트가 제공하는 인터페이스를 사용하면 된다.

##### 4.3 사운드, 파일 입출력

게임의 필요에 따라 Audio, FileIO 컴포넌트를 이용하여 배경음과 효과음, 사용자정보 데이터를 입출력하는 기능을 구현하면 된다. Snake 게임에서는 기본적인 배경음과 효과음 몇 가지를 사용하기 위해 Audio 컴포넌트를 사용하고 사용자의 이전 기록들을 저장하고 불러오기 위해 FileIO 컴포넌트를 사용한다. 그 외에 게임 맵을 하나의 파일로 관리하고 이를 추가하고 불러오는데 FileIO 컴포넌트를 사용할 수 있다.

Fig. 4는 DVAGF를 적용하여 구현한 Snake 게임을 Document-View 영역별로 구분한 클래스 다이어그램이다. 게임 기획을 바탕으로 데이터 설계를 실시하여 Document 영역 구성한다. Snake 게임에서는 Character 클래스를 상속받는 Snake 클래스와 Enemy 클래스, Food 클래스를 구현하고 게임속의 객체들인 Snake, Enemy, Food들의 데이터 상태를 게임 맵 데이터와 포괄적으로 다루는 GameWorld 클래스를 구현하여 Document 영역을 구성한다. 데이터설계 다음으로 설계한 데이터의 상태로 게임화면을 출력하고 사용자의 입력을 받는 컨트롤러를 구현한다. 게임화면과 컨트롤러는 Screen 클래스를 상속받은 GameScreen 클래스에 구현한다. GameScreen

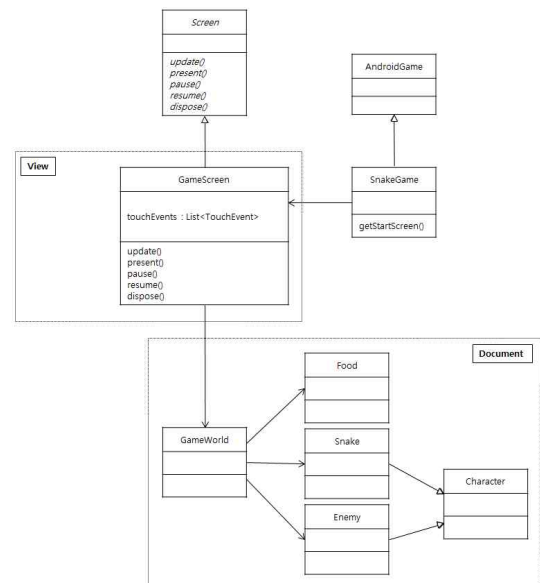


Fig. 4. Division about Document-View pattern of snake game class diagram.

클래스의 `present()` 메서드에는 게임 데이터 상태를 바탕으로 비트맵(Canvas)을 화면에 출력해 준다. 2D, 3D 그래픽을 구현하고 싶다면 비트맵(Canvas)이 아닌 OpenGL ES를 사용하여 구현하면 된다. `GameScreen` 클래스의 `update()` 메서드에서는 `getTouchEvent`를 사용하여 사용자가 방향키를 터치하는지 확인하여 방향전환 로직을 수행한다. 안드로이드의 기본인 `Activity` 생성과 GUI 환경을 구축할 `SurfaceView` 생성은 `AndroidGame` 클래스를 상속받는 `SnakeGame` 클래스가 상위클래스의 생성자를 호출함으로써 자동 생성된다.

DVAGF를 적용한 Snake 게임을 보면 AGF를 제외한 실제 개발자가 추가하는 로직이 전체 소스코드의 약 20%이다. 이는 AGF로 단순 반복되는 로직을 프레임워크로 대체하고 개발자가 추가되는 로직에 집중해서 효율을 높일 수 있게 하고 반복적인 코드 작성을 줄이고 재활용성을 높여서 개발 효율을 높일 수 있다. 즉 빠른 속도로 게임을 개발할 수 있다. 또한 AGF를 사용하여 Document-View 패턴을 적용하는 DVAGF로 데이터(Data)와 게임화면-컨트롤(View)을 분리 코딩함으로써 언제든지 데이터를 쉽게 추가 수정할 수 있어 유지보수성이 높고 사용자에게 보이는 게임화면도 비트맵(Canvas)에서 2D, 3D 그래픽(OpenGL ES)으로 쉽게 변경할 수 있어 확장성도 높다.

#### 4.4 2D, 3D 그래픽으로의 확장

Fig. 5는 OpenGL ES를 사용하여 2D, 3D 그래픽을 구현하기 위해 기존 AGF에 `GLGame`, `GLSurfaceView`, `GLGraphics` 클래스를 추가 확장하고 이를 이해하기 쉽도록 표현한 클래스 다이어그램이다. Fig. 3과 같이 기존 클래스 자리에 확장한 클래스를 배치했을 뿐 실제로는 기존 클래스들을 삭제한 것이 아니며 모든 클래스가 AGF안에 공존하고 있다. 확장한 AGF의 사용은 기존 방식과 같으며 픽셀 이미지의 GUI 환경으로 개발되는 게임은 앞선 방법과 같이 `AndroidGame` 클래스를 상속받아 사용하면 되고 2D, 3D 그래픽의 GUI환경으로 개발되는 게임은 추가 확장한 `GLGame` 클래스를 상속받아 사용하면 된다.

Fig. 6은 앞서 DVAGF 적용사례에서 제작한 Snake 게임을 2D, 3D로 확장하기 위해 확장된 AGF

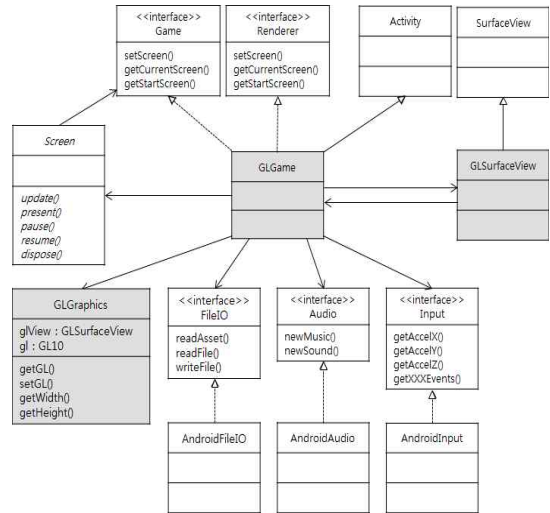


Fig. 5. Extended class diagram of AGF.

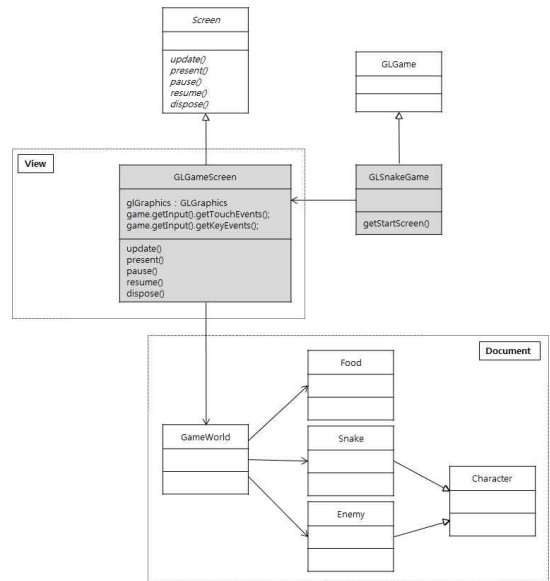


Fig. 6. Snake game class diagram for expansion in 2D and 3D.

를 사용한 클래스다이어그램이다. Fig. 5와 비교해서 보면 Document 영역은 변경되지 않으며 사용자에게 보이는 GUI 환경과 관련된 View영역과 이를 지원하는 `GLSnakeGame` 클래스만 변경되었다. `GLSnakeGame` 클래스는 OpenGL ES를 사용하기 위해 `GLSurfaceView`를 지원하는 AGF의 `GLGame` 클래스를 상속받는다. View 영역에 속하는 `GLGameScreen` 클래스의 코드는 이벤트 핸들링 로직은 거의

같고 사용자에게 보이는 화면을 구성하는 present() 메서드의 로직만 비트맵(Canvas) 처리 코드에서 OpenGL ES 코드로 구현하는 방법만 바뀌게 된다. 이와 관련되어 2D, 3D GUI 환경을 쉽게 구현하는 부수적인 클래스는 개발자의 취향에 따라 적절히 사용하면 된다. 여기서 나타내고자 하는 바는 게임의 핵심 로직은 같고 게임의 일부만 확장 혹은 유지보수할 때 DVAGF를 활용한다면 확장성 유지보수성을 높일 수 있다는 것이다. 특히 게임 프로그래밍의 연례된 프로그래밍 기법 강의와 학습자들이 쉽게 게임을 개발할 수 있도록 해준다.

## 5. 결 론

본 논문에서는 안드로이드 게임프로그래밍 수업 현장에서 학습자들이 보다 쉬운 게임 개발을 지원하기 위하여 게임개발에 적절한 디자인 패턴을 찾고 필요한 API를 제공하는 디자인패턴을 활용한 안드로이드 기반 게임 프레임워크(DVAGF:Android Game Framework Using Document-View Pattern)을 제시하였다.

본 논문에서 고안한 DVAGF에 기반하여 게임을 개발 하면 학습자들의 게임 접근성이 높고 Document-View 패턴의 장점인 확장성과 유지보수성을 가질 수 있다. 또한 안드로이드 게임 프레임워크를 통해 학습자들의 게임개발 접근성, 개발시간 단축, 협업능력을 높여줄 수 있다. 종합적으로 게임 프로그래밍 기법을 능률적으로 학습할 수 있었다. 즉, DVAGF를 통해 효과적으로 게임프로그래밍의 기본 구조를 이해하고 2D, 3D 게임으로 확장도 쉽게 할 수 있는 기본 틀을 제공함을 보였다.

## REFERENCE

[1] J. Choi, H. Kim, and J. Cha, "The Design and Implementation of a Media Framework Inter-

face Based on the Android Platform," *Proceeding of the Summer Conference of the Korean Society of Broadcast Engineers*, pp. 390-392, 2011.

- [2] E. Gamma, R. Helm, R. Johnson, J. Vlissides, *Design Patterns*, Pearson EducationKorea, Seoul, 2015.
- [3] M. Zechner, *Beginning Android Games*, Wiki-Books, Paju, 2011.
- [4] F. Buschmann, *Pattern-Oriented Software Architecture, A System of Patterns*, Wiley, New Jersey, 2013.
- [5] M. Seo, "Design and Implementation for Android Game Framework Using the Linked Observer Pattern," *Journal of the Korea Society of IT Services*, Vol. 12, No. 3, pp. 421-432, 2013.
- [6] T. Lee, *Game Programming Using C*, Now Communication, Ulsan, 2011.
- [7] Y. Park and H. Byun, "Efficient Development and Quality Evaluation of Online Library System using Framework," *Journal of Korea Multimedia Society* Vol. 13, No.5, pp. 627-640, 2010.



최 영 미

1979년 이화여자대학교 수학과 (이학사)  
 1981년 이화여자대학교 전산학 전공 (이학석사)  
 1993년 아주대학교 컴퓨터공학과 (공학박사)

1994년~현재 성결대학교 멀티미디어공학부 교수  
 관심분야 : 지능형교수시스템, 게임프로그래밍