# GM:S의 풍부한 API와 낮은 진입 장벽의 교육적 활용 방법

김진관† · 이희준† †
백양고등학교† · 동안고등학교† †

# Brand New SW Education Solution with GM:S

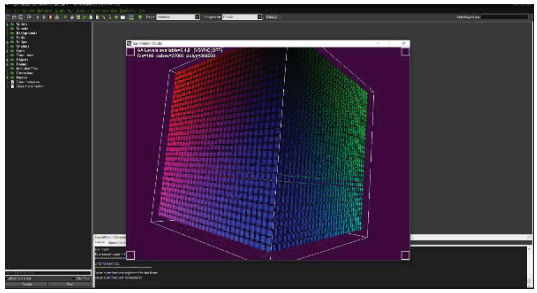
Jinkwan Kim† · Heejun Lee† †
Paik-yang High School† · Dong-an High School† †

## 요 약

기존의 프로그래밍 교육용 프로그램(스크래치, 엔트리)들은 프로그래밍의 진입장벽을 낮춰 교육의 효율을 높일 수 있었다[1][2]. 하지만 그러한 접근 방식은 실용적인 프로그래밍에 대한 지표가 되기에는 좋지 않다. 우리는 교육용 프로그램으로 게임메이커 스튜디오(이하 GM:S)를 사용하는 것을 제안 하고, 기존 지표의 단점을 풍부한 스크립팅 API와 네이티브 라이브러리 링크 등으로 극복한 GM:S를 교육적 목적으로 사용하는 방법에 대해 알아보고자 한다.

**주제어** : 게임 메이커 스튜디오, 소프트웨어 개발, 소프트웨어 교육

## 1. 서 론



[그림 1] 게임메이커 스튜디오

Mark Overmars 교수의 그래픽스 툴 Animo를 기반으로 한 게임메이커는 초창기에 델파이(Delphi)라는 언어로 만들어진 게임 개발 교육 목적에 특화된 프로그램이었으며, 이것이 성장하여 훗날 요요게임즈(YoYo Games)의 GM:S가 되었다. 훗날의 게임메이커 스튜디오는 이전 버전의 게임메이커부터 포함되어 온 편리한 GML을 이용하여 코드를 작성하고, 이를 네이티브 저급언어로 게임을 컴파일 할 수 있는 경지에 이르렀고, 이는 최근 결국 게임뿐만 아니라 다양한 유틸리티 또한 하이엔드급 성능으로 빌드 할 수 있는 효율적인 IDE로 발전했다. 하지만 GM:S에 대한 보편적인 사람들의 인식은 아직 게임 제작에만 특화된 프로그램쯤에서 머무르고 있다.

하지만 우리는 GM:S가 기존의 SW 교육 방식을 완전히 새롭게 바꿀 수 있다고 생각하여, 기존의 스크래치 등의 프로그램으로 지도하기 힘들었던 고급 프로그래밍 기술을 GM:S를 기반으로 교육하는 편리한 방법에 대해 연구했다.

## 2. 이론적 배경

### 2.1 게임메이커 스튜디오(GM:S)에 대한 이해

#### 2.1.1 게임메이커 및 GM:S의 특징

ⓐ. 게임메이커 스튜디오는 프로그램 개발용 IDE로, GML (Game Maker Language) 이라는 독자적인 쉬운 스크립트 언어와, GML에 포함된 함수나 구문을 액션 단위로 쌓는 간편한 에디터를 포함하고 있어 프로그래밍을 처음 접하는 학생들도 쉽게 이해할 수 있다.

ⓑ. GM:S는 단순 게임 개발 용도의 API뿐만 아니라 그래픽스, 오디오와 수학 등을 위한 깊은 내장 API를 지니고 있으며 외부 언어로의 확장 자유도가 높다. 따라서 이름처럼 게임을 제작하는 용도뿐만 아니라 유틸리티나 일반적인 응용 소프트웨어 또한 쉽게 개발하는 것이 가능하다.

ⓒ. 풍부한 API를 보유하고 있으며 이와 더불어 기획, 디자인, 프로그래밍이 잘 어우러질 수 있는 통합적인 IDE를 제공한다.

ⓓ. 오랜 역사와 함께 거대한 커뮤니티와 사용자층을 보유하고 있다.

ⓔ. GM:S는 네이티브 컴파일러를 통하여 소프트웨어를 빌드 할 수 있어 높은 성능을 낼 수 있으며, 또한 섬세한 빌드관련 설정을 할 수 있다.

### 2.1.2 게임메이커 및 GM:S 의 상세 기능

게임메이커는 게임을 만들 때 필요한 기능들이 주로 탑재되어 있다. 크게 이미지나 오디오등의 리소스를 관리하는 시스템, 실제 프로그램 화면을 구성하는 룸 시스템, 프로그램 안의 오브젝트를 제어하는 스크립팅 시스템이 있다.

리소스로써는 스프라이트 이미지, 클립 오디오, 네이티브 라이브러리를 위한 dll따위를 포함한 외부 파일, 미리 작성된 스크립트 등이 대부분이다. 내부적으로 스프라이트 에디터를 지원하고 간단한 이미지 효과나 도트 드로잉의 기능을 포함하고 있다. 오디오는 내부 에디터는 존재하지 않고 빌드 시 사용할 코덱등을 지정할 수 있다. 미리 작성된 스크립트들은 각 스크립트 파일이 함수와 같이 호출 할 수 있어, gui 툴킷이나 라이팅 엔진 등 사용자 라이브러리를 작성하는데 유용하게 사용된다.

룸은 게임의 화면을 지정 하는 것으로써, 작성된 게임 오브젝트를 인스턴스로 룸에 생성하여 실행할 수 있다. 룸은 룸에디터를 통해 작성될 수 있으며, 생성할 오브젝트의 종류, 화면상의 위치, 화면의 크기 등을 지정할 수 있다.

오브젝트를 제어하는 스크립트 작성은 액션 단위의 블럭형 에디터와 텍스트형 에디터를 둘 다 지원한다. 텍스트형 에디터를 통한 스크립트는 GML 및 HLSL, GLSL 등으로 작성할 수 있다. 스크립트는 이벤트 드리븐 형식으로 빌드 되어 실행되고, 작성된 스크립트들은 연결된 이벤트 발생 시에 실행된다. 여기서 미리 작성된 스크립트를 호출하여 타인이 작성한 라이브러리를 사용할 수 도 있다. 스크립트는 간단한 조건문이나 반복문부터 외부 라이브러리 연동, 내부 프레임워크 API 등에 이르기까지 확장성이 높아 게임뿐만 아니라 여러 일반적인 프로그램에도 사용할 수 있게 해준다. 엔진 API에는 세이더를 포함한 GPU 기반 2D/3D 그래픽스, 오디오 프로세싱, 수학, 시스템 API 등 필요한 많은 기능들이 기본적으로 구현되어 있어 사용 가능하다.

### 2.1.3 게임메이커 및 GM:S 에서의 프로그램 제작 과정

제일 먼저 사용될 게임 오브젝트를 생성하고, 오브젝트가 화면에서 보일 스프라이트를 지정하여 오브젝트를 룸에 배치 시 화면에 출력할 수 있다. 그리고 오브젝트에 이벤트에 연결된 액션(스크립트)을 추가하여 오브젝트를 프로그래밍 할 수 있다. 스크립팅을 통하여 오브젝트 프로그래밍이 완료 되면, 이는 하나의 인스턴스로서 역할을 수행한다. 룸은 오브젝트 스크립트에 의해 다른 룸으로 이동될 수 있으며, 룸 에디터를 통해 뷰포트의 상태 등을 지정할 수 있다.

오브젝트의 제작과 룸 배치를 통해 룸 제작을 완료했으면, 프로젝트를 가능한 플랫폼으로 빌드하여 디버거와 함께 디버깅을 할 수 있다. 디버거는 기본적인 FPS 미터와 중단점을 지원하고 또한 스택트레이스, 메모리 힙 분석, 코드 실행 시간 분석등 전문적인 기능까지 지원한다. 이러한 디버깅 과정을 통해 프로그램을 최적화하고, 버그를 최소한으로 만들 수 있다.

모든 개발 작업이 완료되면, 릴리즈 모드로 빌드하여 내보낼 수 있다. 내보낸 실행 파일은 배포되어 실행 될 수 있으며, 모바일이나 콘솔로 빌드하게 되면 패키징하여 앱 스토어 등에 등록이 가능하다.

### 2.1.4 비용

| | Trial | Desktop | Web | UWP | Mobile | PS4 | Xbox One | Ultimate |
|---|---|---|---|---|---|---|---|---|
| Unlimited Resources | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Expert Features | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Target Platform(s) | Windows TEST only | Windows, Mac, Ubuntu | HTML5 | Microsoft UWP | Android, iOS | PS4 | Xbox One | All Platforms |
| GameMaker: Studio 1.4 Professional Access | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Marketplace | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Support | GMC | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| License Type | Permanent | Permanent | Permanent | Permanent | Permanent | 12 Month | 12 Month | 12 Month |
| | DOWNLOAD | $99.99 | $149.99 | $399.99 | $399.99 | $799.99 | $799.99 | $1500.00 |

[그림 2] 게임메이커 스튜디오 에디션별 비교표

GM:S는 Windows 응용 프로그램용 IDE를 공짜로 이용할 수 있는 세션을 열어두고 있으며, 크로스 플랫폼용 응용프로그램 개발 툴 및 네이티브 컴파일러를 유료로 배포하는 합리적인 마케팅을 하고 있다. 이와 더불어, 연중행사 (계절별 할인 행사 및 블랙 프라이데이 등) 때마다 적극적인 할인을 제공하고 있어 타 개발 툴에 비해 저렴하게 개발을 시작할 수 있다.

## 3. 본론

### 3.1 게임메이커와 기존 프로그램과의 비교

기존의 교육용 프로그램 (스크래치, 엔트리, 플레이봇) 들은 기본적으로 블록형 프로그래밍 방식을 채택하고 있다. 또한 샌드박싱된 언어 기능과 프레임워크 API를 통해 프로그래밍의 진입장벽과 난이도를 낮추는데 주력하고 있다. 하지만 이러한 프로그램들은 너무 진입장벽과 난이도를 낮추는 것에 초점을 맞추고 있다고 생각한다. 만일 스크래치를 통해 프로그래밍을 교육 받았다 하더라도 텍스트 기반 프로그래밍 언어를 사용하려면 곤란할 점이 많다. 예를 들어 C의 포인터 개념이나 네이티브 라이브러리 바인딩, 프레임워크나 API와 관련된 버그의 디버깅에 대한 이해 등 기존의

샌드박싱된 교육용 프로그램으로는 배울 수 없는 것들이 많다. 또한 메이저급 프로그램을 개발하게 된다면 하드웨어적 이해와 관련 플랫폼의 이해또한 필요하기 때문에 이러한 점도 빠른 프로그래밍 교육의 발목을 잡는다[1][2][3].

하지만 게임메이커는 기존의 교육용 프로그램들이 지니는 근본적 문제를 해결하고 더 나아가 디자인, 디버깅 등을 포함한 실무 방향으로 확장된 교육이 가능한 도구이다. 게임메이커는 스크래치와 같이 블록형 에디터를 지원하지만 또한 스크립트(GML) 및 셰이더 언어(HLSL, GLSL 등)기반의 텍스트형 에디터도 지원한다. 이 점은 사용자가 게임메이커의 블록형 코딩을 습득한다면 텍스트형 코딩 또한 큰 문제없이 이해할 수 있음을 의미한다[4]. 또한 GML은 C의 문법을 간략화한 스크립트 언어이기에 간단히 배울 수 있고, 차후 포인터와 같은 로우레벨 기능들만 이해한다면 하이레벨 뿐만 아니라 로우레벨의 언어를 사용하는 데에도 문제가 없을 것이다.

게임메이커는 직관적인 디버거 또한 포함한다. 디버거에는 그래프로 CPU/RAM사용량, FPS등을 실시간으로 모니터링 가능하고, 간단히 소스코드의 원하는 라인에 중단점을 찍어 프로그램을 일시 정지시키고 디버깅을 가능하게 한다. 이러한 디버깅 과정은 산업용 IDE에서도 사용되는 것으로 학습자가 바로 실제 컴퓨팅 문제를 해결할 수 있게 만들어 준다.

스크래치나 엔트리와 같은 개발 환경은 GM:S에 비해 다소 많이 샌드박싱되어 프로그램의 구성, 최적화, 디자인, 기획 및 로드맵과 같은 사항을 고려하며 프로그래밍 하는 것을 배우기에는 적합하지 않다고 생각한다. 스크래치나 엔트리로써는 프로그램 개발 프로젝트의 범위가 작게 제한이 되기 쉽다. 그것들은 블록형 프로그래밍을 사용하기 때문에 긴 코드를 작성할 시 코드의 흐름을 파악하기 힘들고 이는 곧 프로그램의 구현 및 구상을 하기 힘들게 만들어 메이저급 프로그램을 구현하는데 많은 어려움이 따르게 한다[5]. 또한 스크래치는 데이터를 관리하는데 있어 구조체에 대한 기능 구현이 미흡한 탓에 데이터를 관리하고 사용하는 것을 배우고 구현하기 적합하지 않다. 반면 게임메이커는 배열과 리스트, 딕셔너리등 산업용 언어에서도 지원하는 많은 구조체 타입을 지원하기 때문에 사용자가 스크립트 내부에서 사용 가능하다.

### 3.2 게임메이커 활용 방안


[그림 3] 저자가 직접 GM:S로 개발한 프로그램

GM:S는 실제로 배포해도 전혀 부끄럽지 않은 프로그램을 제작할 수 있는 세밀한 도구이다.

다르게 말하면, GM:S로 제작한 프로그램을 통한 수익 창출 및 관련 사업 진행도 전혀 어색하지 않다는 이야기이다. 2.1.4로 인해 초기 비용이 비싸보일 수 있으나, 다양한 마켓(Google Play, App Store, Windows Store 등지)에 GM:S로 개발한 프로그램을 배포한 후 GM:S 제작사(YoYo Games)와의 수익 나눔없이 100% 개발자가 수익을 창출할 수 있다는 점을 고려하면 더욱이 2.1.4는 합리적이라고 할 수 있다. 따라서 GM:S를 이용하면 개발자는 아이디어와 적은 초기 비용만으로 무궁무진한 개발을 할 수 있다.

작게는 엔터테인먼트를 위한 소프트웨어부터, 웹 소프트웨어, 유틸리티 소프트웨어, 교육용 소프트웨어 등에 이르기까지 말이다. 소프트웨어는 사실상 대부분 GM:S를 주축으로 개발될 수 있다.

## 4. 결론 및 요지

본 연구는 기존의 프로그래밍 교육용 소프트웨어의 활용성 한계를 GM:S의 다양한 API, 네이티브 접근성, 쉬운 스크립트 언어(GML)를 통해 산업적 현실과 동떨어진 소프트웨어 개발 교육 실태를 극복할 수 있는 게임메이커 스튜디오에 대해 알아보았다. GM:S를 통한 교육은 합리적이고, 개발자가 고급 프로그래밍 능력을 기를 수 있는 좋은 초석이 되어 준다.

## 참 고 문 헌

[1] Scratch. 2017년 07월 28일 검색 https://scratch.mit.edu/
[2] Entry. 2017년 07월 28일 검색 https://playentry.org/
[3] PlayBot. 2017년 07월 28일 검색 http://playbot.spaceii.com/
[4] 김수환 (2015). Computational Thinking 개념 평가를 위한 스크래치 코드 분석 시스템 개발. 한국컴퓨터교육학회 논문지. 18(6), 13-22.
[5] 소미현, 김자미 (2016). 블록형 프로그래밍 학습에서 텍스트형 프로그래밍 학습으로의 전이. 한국컴퓨터교육학회 논문지. 19(6), 55-68