

아빠와 함께하는

버전2

스크래치 프로그래밍 어드벤처

교사용
지침서



<http://scratch.mit.edu>

이 문서는 학생들의 동아리 활동, 방과 후 수업 등 스크래치 수업에서
《스크래치 프로그래밍 어드벤처》와 함께 활용할 수 있도록 제작한 교사용 지침서입니다.

스크래치 프로그래밍 어드벤처의 리소스 파일 이해하기

우선, 본 교재의 리소스 파일을 <http://www.nostarch.com/scratch>에서 다운로드하십시오. 다운로드한 리소스 파일은 학생들이 쉽게 찾을 수 있는 곳에 컴퓨터마다 복사해 두는 것이 좋습니다.

본 교재용으로 사전에 만들어 제공하는 프로젝트 리소스 파일은 두 버전으로 나뉩니다. 학생들의 연령이나 수준에 따라 선택하여 수업에 활용할 수 있도록 프로그래밍이 빠진 버전과 프로그래밍까지 포함된 버전이 함께 제공됩니다. 두 버전 모두 게임에 필요한 캐릭터와 스프라이트, 사운드 효과 등은 차이 없이 제공됩니다. 프로그래밍까지 제공되는 온전한 프로젝트 파일은 Complete Games 폴더에 들어 있습니다.

학생들에게 직접 프로젝트를 일일이 만들게 할 목적이라면 Complete Games 폴더 자체를 삭제하는 것이 좋습니다.

혹시 여건이 조성되지 않았거나 수업 시간이 부족하다면? 온전한 프로젝트 파일을 로드하여 학생들에게 이것저것 실습해 보도록 유도하는 것이 좋습니다. 시행착오야말로 프로그래밍을 시작하는 최적의 방법입니다. 프로그래머들이 다른 사람의 것을 '표절'하는 것부터 시작하는 경우는 대단히 많습니다.

학생들이 본 교재에 사용된 스프라이트를 자신의 프로젝트에 사용해도 되는 것인지 질문한다면? 전혀 문제없습니다. 원래 프로젝트 파일을 열고, 원하는 스프라이트를 각 학생의 개인 저장소에 끌어다 놓기만 하면 됩니다(개인 저장소는 브라우저 아래쪽에 있습니다. 자세한 내용은 2장을 참고하십시오). 또는 원하는 스프라이트를 스프라이트 목록에서 마우스 오른쪽 버튼으로 클릭한 다음, '내 컴퓨터에 저장하기'를 클릭할 수도 있습니다.

학생들에게 상상의 날개를 달아주자!

본 교재를 통해 한두 게임을 만들어 본 다음에는 학생들에게 자신만의 게임이나 스토리를 스스로 상상하고 만들 수 있는 기회를 제공하십시오. 본 교재에 제공되는 게임을 단순히 복제하는 것이 아니라 스크래치를 즐길 수 있도록 환경을 조성하는 것이 중요합니다. 멋진 게임을 '직접' 만들 수 있는 스크래치가 바로 창의력의 요체입니다.

학생들은 <http://scratch.mit.edu>에서 다른 사람의 작품을 자신의 프로젝트로 가져와 전혀 다른 새로운 모습으로 재창조할 수 있습니다.

지원을 받을 수 있는 곳

MIT에는 교사를 위한 포럼이 <http://scratched.media.mit.edu>에 마련되어 있습니다. 동료 교사의 질문이나 교육 성공담 등을 둘러볼 수 있는 대단히 매력적인 곳입니다.

이 교사용 지침서나 본 교재에 대한 피드백은 tyler@nostarch.com(편집자 주: 해당 메일은 원출판사의 메일입니다. 이 교사용 지침서나 본 도서에 대한 피드백은 jeipub@gmail.com으로 보내주시면 됩니다)으로 보내주십시오. 스크래치에 관한 것이라면 언제든지 어떤 내용이든 모두 환영합니다. <http://scratch.mit.edu/users/nostarch>에도 방문해 보십시오. 학생들이 만든 멋진 프로젝트를 이곳에서 공유할 수도 있습니다. 곧 스크래치 갤러리도 추가할 예정입니다.

스크래치 1.4 버전과 2.0 버전 중 어느 것을 사용해야 하는가?

사용하는 컴퓨터가 ‘다소’ 구형이거나 느리다면 온라인 버전인 스크래치 2.0보다는 스크래치 1.4를 사용하는 것이 좋습니다. 스크래치 1.4 버전은 아래 주소에서 다운로드할 수 있습니다.

http://scratch.mit.edu/scratch_1.4/

오프라인용 스크래치 2.0도 제공됩니다. 아래 주소에서 다운로드할 수 있지만, 아직은 베타 버전입니다.

<http://scratch.mit.edu/scratch2download/>

학생들의 개인정보 침해가 우려된다면(또는 수업 시 학생들의 인터넷 검색이 우려된다면), 스크래치 1.4 버전이나 오프라인용 스크래치 2.0 버전을 사용하는 것도 좋은 방지책입니다.

스크래치 1.4는 현재 피코보드(Picoboard)를 지원하는 유일한 버전입니다. 피코보드는 스크래치와 함께 사용할 수 있는 하드웨어 인터페이스입니다.

스크래치 1.4와 2.0에서 가장 두드러지는 차이는 각종 인터페이스 요소(무대, 프로그래밍용 팔레트, 스크립트 영역 등)의 화면상 위치입니다. 새로운 팔레트로 이동한 블록도 몇 가지 있고, 개인 저장소와 같은 기능도 새로운 버전에만 제공됩니다.

다음은 본 교재에 제공되는 프로젝트 파일입니다.

<http://nostarch.com/download/Super-Scratch-2-Resources.zip> (2.0 버전)

<http://nostarch.com/download/Super-Scratch-Resources.zip> (1.4 버전)

1.4 버전의 파일을 2.0 버전에는 로드할 수 있지만, 거꾸로는 안 됩니다. 스크래치 2.0은 이전 버전과 호환되지 않습니다. 스크래치 1.4 버전을 사용한다면 각별히 주의해야 합니다.

STAGE 1

★ 필요한 파일

따로 필요한 파일이 없습니다. 단, 컴퓨터와 인터넷 연결을 확인하십시오. <http://scratch.mit.edu>에 방문하여 '만들기'를 클릭하기만 하면 됩니다.

첫 수업에서는 컴퓨터의 이상 유무를 확인하고 스크래치를 즐길 준비를 마무리해야 합니다. 학생들의 온라인 이용을 위한 학부모의 동의가 필요할 수도 있습니다. 학생들에게 미리 <http://scratch.mit.edu>에 등록하도록 조치해 두는 것이 좋습니다.

첫 수업에서는 질문이 쏟아져 나올 수 있습니다. 만반의 준비를 갖춰 놓으십시오.

이 장의 목표는 학생들에게 스크래치의 사용자 인터페이스에 익숙해지도록 하고, 기본적인 프로그래밍 개념을 소개하는 것입니다.

★ 핵심 개념

- 스크래치에서는 짧은 프로그램(스크립트)을 스프라이트에 만들어야 합니다. 프로젝트의 캐릭터를 스프라이트로 부릅니다.
- 팔레트에서 명령 블록을 클릭한 채 끌어다 놓아 명령 스택을 만듭니다. 블록이 모여 프로그램을 이루는데, 프로그래밍 블록을 떼어내거나 삭제하고 복사하는 방법을 익히도록 해주십시오. 프로그램을 만들고 개선할 능력을 길러 주십시오. 이후 장에서는 학생들이 블록을 이동하거나 삭제하고 블록 안에 블록을 두거나 블록을 재배열할 수 있다고 가정하고 있습니다.
- 스프라이트에는 스크립트가 여럿일 수 있습니다.
- 블록을 테스트하려면 블록을 클릭합니다.
- 무대에서 스프라이트의 위치는 좌표로 정해집니다. 이는 수학 시간에 보던 것과 같습니다.
- 컴퓨터는 들은 대로만 움직이는 '바보'입니다. 컴퓨터에 멋진 명령을 내릴 수 있습니까?
- 실수는 일어나게 마련! 실수를 두려워해서는 안 됩니다. 실수를 너무나 많이 하는 프로그래머들은 실수를 해결하는 일에 아예 '디버깅'이라는 특별한 이름까지 붙였습니다.

★ 심화 탐구학습

스크래치의 블록들이 무슨 일을 하는지 궁금하다면? 리소스 파일과 함께 제공되는 MIT의 '시작하기 가이드' PDF 문서를 참고하십시오. 스크래치 자체에 포함된 블록 정보를 확인하려면 브라우저 상단의 도움말을 클릭하십시오.

스크래치의 펜 블록은 전통적인 프로그래밍 학습 도구인 로고(LOGO)처럼(또는 파이썬의 Turtle 모듈처럼) 사용될 수도 있습니다. 펜 블록을 사용하여 다양한 수학적 기능을 익힐 수 있습니다.

학생들에게 스크래치로 '호흡기록기'를 만들게 하거나 꽃잎을 여러 번 그려 꽃을 만들게 하는 것도 좋습니다.

★ 추천 읽기 자료(교사용 및 학생용)

스크래치는 1970년대 MIT에서 처음으로 개발된 아이들용 프로그래밍 환경이라는 탄탄한 역사를 지니고 있습니다.

[http://en.wikipedia.org/wiki/Logo_\(programming_language\)](http://en.wikipedia.org/wiki/Logo_(programming_language))

Mindstorms: Children, Computers, and Powerful Ideas,

세이모어 페이퍼트(Seymour Papert) 저, 1980. ISBN 9780465046744

Turtle Geometry: The Computer as a Medium for Exploring Mathematics,

헤럴드 아벨슨(Harold Abelson), 안드레아 디세사(Andrea diSessa) 공저, 1986. ISBN 9780262510370

STAGE 2

★ 필요한 파일

빈 파일인 02 - A Space Odyssey.sb2 또는 온전한 프로젝트 파일을 업로드합니다(파일 ▶ 내 컴퓨터에서 프로젝트 업로드하기). 이번 프로젝트에도 특별한 스프라이트가 필요하지 않습니다. 학생들이 직접 만든 이미지를 게임에 사용해도 무방합니다.

★ 핵심 개념

- 스프라이트에는 모양이 여럿 포함될 수 있습니다. 한 스프라이트에서 모양을 여러 가지로 변경하면 애니메이션 효과를 낼 수 있습니다.
- 무대에는 배경이라는 ‘모양’이 적용됩니다.
- 스크래치에서는 사용자에게서 입력을 받을 수 있습니다. 사용자 입력을 적절히 활용하면 멋진 게임을 편리하게 만들 수 있습니다.
- 명령 블록 안에 다른 명령 블록을 끼워 넣는, 이른바 명령 블록의 중첩을 학생들에게 소개합니다. 블록이 서로 ‘딱 들어맞도록’ 차근차근 진행해야 합니다.
- 게임을 만들기 위해 이미지를 직접 그려야 하는 것은 아닙니다. 스크래치에는 곧바로 사용할 수 있는 이미지가 함께 제공됩니다.

★ 심화 탐구학습

- 이번 게임은 절대로 질 수 없다는 사실을 알아챈 학생이 있습니까? 그런 학생들에게는 본 교재의 저자인 에드먼드(Edmond)가 같은 개념을 적용하여 직접 만든 게임을 소개해 주십시오. 이 게임은 <http://scratch.mit.edu/projects/101909>에서 만날 수 있습니다. 그는 게임을 개선하기 위해 어떻게 생각했을까요? 어느 게임이 더 재밌나요? 학생들에게 자신이 좋아하는 아이폰 게임이나 플래시 게임의 ‘프로토타입’을 직접 만들게 하십시오. 이 프로토타입에서 개선할 점을 찾도록 지도하십시오.
- 미래의 피카소들이 더욱더 성능이 뛰어난 그래픽 에디터를 원하고 있습니까? 무료 그래픽 에디터인 잉크스케이프(Inkscape, <http://inkscape.org>)나 킴프(GIMP, <http://www.gimp.org/>)가 도움이 될 수 있습니다. 스크래치는 다른 프로그램에서 사운드와 그래픽을 가지고 올 수 있습니다.
- 본 교재에 사용된 여러 ‘레트로’ 소리 효과는 <http://www.bfxr.net/>에서 제공하는 무료 온라인 도구로 작성했습니다.

STAGE 3

이번 게임은 직접 플레이한 다음에 학생들에게 만들도록 하는 것이 좋습니다. 게임은 총 2부로 구성되었습니다. 1부는 퀴즈 게임, 2부는 기억력 게임입니다.

이번 게임은 본 교재에서 만들게 될 인터랙티브 게임의 시작입니다. 또한 더욱더 탄탄한 퍼즐 게임을 만들기 위한 토대로서, 게임의 흐름을 어떻게 제어하는지 이해하는 데 큰 도움이 되는 게임입니다. 다만, 성미가 급한 학생들에게는 다소 어울리지 않는 면도 없지 않습니다.

★ 필요한 파일

빈 프로젝트인 03 - Louvre Puzzle.sb2 또는 온전한 프로젝트 파일을 업로드합니다(파일 ▶ 내 컴퓨터에서 프로젝트 업로드하기). 이번 게임에는 다시 만들기가 힘든 복잡한 스프라이트가 여럿 포함되었습니다.

★ 핵심 개념

- 스프라이트가 말을 하려면 ‘말하기’ 블록이 필요합니다.
- ‘방송’을 사용하여 스크립트의 흐름을 제어합니다. 큰 프로그램을 여러 개의 작은 조각으로 나누기 위한 방송 사용 방법을 익힙니다.

★ 심화 탐구학습

더욱더 흥미진진한 퍼즐을 만들도록 하십시오. 학생들이 가장 좋아하는 보드 게임을 스크래치 버전으로 탈바꿈시킬 수 있을까요? 카드의 짝을 맞추는 기억력 게임은 어떨까요?

STAGE 4

★ 필요한 파일

빈 프로젝트인 04 - Hack Attack!.sb2 또는 온전한 프로젝트 파일을 업로드합니다(파일 ▶ 내 컴퓨터에서 프로젝트 업로드하기). 이번 프로젝트에도 특별한 스프라이트가 필요하지 않습니다. 학생들이 직접 만든 이미지를 게임에 사용해도 무방합니다.

★ 핵심 개념

- 스크래치 블록 안에 다른 블록을 끼워 넣을 수 있습니다. 조건 블록(만약 _라면 블록)은 프로그래머에게 대단히 강력한 도구입니다.
- 더욱더 복잡한 게임에 익숙해지며, 플레이어의 마우스 클릭으로 제어하는 게임을 만들 수 있습니다.

★ 심화 탐구학습

게임이 더 어려워지도록 두 번째 바이러스를 추가하십시오. 프로그래밍을 약간 다듬어야 할 것입니다.

진정한 도전 과제가 필요하다면 고전 아케이드 게임인 팡(Pong)을 다시 만들어 보십시오. 팡에 관한 정보는 <https://www.youtube.com/watch?v=XNRx5hc4gYc>에서 찾아볼 수 있습니다. 팡은 전문 프로그래머 몇 명이 몇 달이나 걸려 만든 게임입니다. 학생들이 팡을 스크래치로 다시 만드는 데 얼마나 걸릴까요? 지난 40여 년간 프로그래머들이 즐겨 사용하는 도구가 얼마나 발전을 거듭했는지 생각해 보는 것도 의미 있을 것입니다. 요즘은 많이 편해진 것일까요?

STAGE 5

★ 필요한 파일

빈 프로젝트인 05 - Rio Shootout.sb2 또는 온전한 프로젝트 파일을 업로드합니다(파일 ▶ 내 컴퓨터에서 프로젝트 업로드하기).

★ 핵심 개념

- 변수를 사용하여 게임에서 점수를 기록하는 등 숫자를 담을 수 있습니다.
- 이번 게임의 로직은 대부분 Ball 스프라이트에 담겼으므로 프로그램의 전반적인 구성도 이 스프라이트로 알 수 있습니다(특별한 의도로 Ball 스프라이트를 선택한 것은 아닙니다).

STAGE 6

★ 필요한 파일

빈 프로젝트인 06 - Desert Rally.sb2 또는 온전한 프로젝트 파일을 업로드합니다(파일 ▶ 내 컴퓨터에서 프로젝트 업로드하기). 이번 게임에는 다시 만들기가 힘든 복잡한 스프라이트가 여럿 포함되었습니다.

★ 핵심 개념

- 무대의 배경을 변경하여 멋진 애니메이션을 만들 수 있습니다.
- 변수를 사용하여 소품을 화면 이곳저곳으로 이동할 수 있습니다. 이번 게임에서는 두 개의 Road 스프라이트로 이 효과를 구현했습니다.

★ 심화 탐구학습

학생들에게 더욱더 빠른 게임으로 만들도록 하십시오. 스파이 헌터(https://en.wikipedia.org/wiki/Spy_Hunter) 스타일의 레이싱 게임처럼 종스크롤 게임으로도 변형하도록 하십시오.

STAGE 7

★ 필요한 파일

빈 프로젝트인 07 - The Maze.sb2 또는 온전한 프로젝트 파일을 업로드합니다(파일 ▶ 내 컴퓨터에서 프로젝트 업로드하기). 반드시 이번 게임에 포함된 이미지를 그대로 사용해야 프로그래밍이 올바르게 동작합니다. 게임 로직이 스프라이트의 색상에 맞춰 구성되었기 때문입니다.

★ 핵심 개념

- 가장자리 판단 조건이나 게임 종료 조건 등을 ‘_색에 닿았는가?’ 블록으로 만들 수 있습니다.

★ 심화 탐구학습

자동차 레이싱 게임을 만들 때 ‘_색에 닿았는가?’ 블록을 어떻게 활용할 수 있을까요? 가령, 경주 도로의 가장자리에 자동차가 부딪히면 타이어가 핑크나도록 할 수 있습니까?

STAGE 8

★ 필요한 파일

빈 프로젝트인 08 - Wizard's Race.sb2 또는 온전한 프로젝트 파일을 업로드합니다(파일 ▶ 내 컴퓨터에서 프로젝트 업로드하기).

★ 핵심 개념

- 단순한 게임도 경우에 따라서는 꽤 복잡하게 프로그래밍할 수밖에 없습니다.
- 치트 키(cheat key)를 방지하기 위한 로직을 게임에 적용했습니다.

★ 심화 탐구학습

어떻게 하면 본 교재에 제공되는 다른 게임에도 플레이어의 ‘치트 키’를 방지할 수 있을까요?
게임 디자이너는 어떻게 치트 키를 게임에 적용할까요?

STAGE 9

이번 게임은 본 교재에서 제공하는 게임 가운데 가장 크고 복잡합니다! 학생들에게 프로젝트를 완수할 수 있도록 충분한 시간을 제공하는 것이 좋습니다. 프로젝트를 완료하기까지 몇 시간이 걸릴 수도 있고, 경우에 따라서는 여러 날이 필요할 수도 있습니다. 학생들의 연령이나 욕구, 의지 등에 따라 달라집니다. 프로그램의 전체 크기가 상당하므로 프로그래밍 도중 그동안 진행한 결과물을 테스트하는 것이 좋습니다. 테스트하기에 적당한 곳은 본 교재에 표시되어 있습니다.

★ 필요한 파일

빈 프로젝트인 09 - Final Fight.sb2 또는 온전한 프로젝트 파일을 업로드합니다(파일 ▶ 내 컴퓨터에서 프로젝트 업로드하기). 이번 게임에는 복잡한 스프라이트가 사용되었습니다. 게임 내 캐릭터에 필요한 여러 스프라이트와 애니메이션 등이 그 예입니다.

★ 핵심 개념

- 학생들은 스크래치의 면면을 활용하는 과정에서 Cat 스프라이트의 스프라이트 처리 방식을 어려워할 수도 있습니다.

★ 심화 탐구학습

고양이 스치의 독특한 결투 동작이나 어둠의 마법사의 새로운 공격 기술을 만들 수 있습니다.

더욱더 고차원적인 게임을 만들 수 있을까요? 새로운 캐릭터를 만들어 플레이어에게 선택하도록 하는 화면을 추가하는 것은 어떨까요? 미치의 니키는 상당히 치명적이라는 소문도 있습니다.

앞으로 할 일

스크래치로 더 많은 프로그램을 만들 준비가 되어 있습니까? info@nostarch.com으로 이메일을 보내면 《Learn to Program with Scratch》의 일부 내용을 무료로 받을 수 있습니다(편집자 주: 아래 주소에서 'Learn to Program with Scratch'의 일부 내용을 다운로드할 수 있습니다). 이 책은 학생들에게 친숙한 인터페이스로 컴퓨터 과학의 여러 개념을 파헤쳐 줍니다.

<http://nostarch.com/learnscratch>

텍스트 기반 프로그래밍 언어를 배울 준비가 끝났다면 파이썬을 권장합니다. 제이슨 브릭스(Jason Briggs)의 《Python for Kids》를 한번 살펴보십시오.

<http://nostarch.com/pythonforkids>

프로그래밍을 공부하려는 십대라면 가벼운 마음으로 리스프(Lisp)를 접하는 것도 좋습니다. 삽화로 진행되는 《Land of Lisp》는 고전적인 프로그래밍 언어를 살펴보는 데 적합합니다. 또는 초보자에게 친숙한 DrRacket 개발 환경이 적용된 《Realm of Racket》도 좋습니다.

<http://nostarch.com/lisp.htm>

<http://nostarch.com/realmofracket>

ScratchEd

잊지 말고 꼭 ScratchEd를 둘러보십시오. ScratchEd는 스크래치를 사용하는 선생님들의 정보 공유 사이트입니다. 이곳에서 성공담이나 스크래치 리소스 등을 공유하고 질문하십시오.

<http://scratched.media.mit.edu/>

본 교재와 관련된 것이라면 어떤 피드백도 환영합니다. 본 교재를 선택해 주셔서 감사드립니다. 스크래치와 늘 함께하십시오!