

스크래치 프로그래밍(Scratch Programing)

- 학습목표1. 프로그래밍이란 무엇인지 이해하기
- 학습목표2. 물체 표현과 움직임 표현을 통한 스크래치의 동작원리 이해하기
- 학습목표3. 스크래치를 이용하여 간단한 게임이나 프로그램 만들어 보기

생각 열기

- ① 우리가 문제를 풀 때, 어떤 생각의 과정을 거치는지 알아보자.
예. 갈증이 날 때 갈증을 해소하는 방법

- ② 풍경을 그릴 때 어떤 과정을 거치는지 알아보자.

1. 스크래치 기본 개념

1) 스크래치란?

스크래치(Scratch)는 2007년 미국 MIT대학의 미디어랩과 UCLA의 연구자가 공동으로 개발한 교육용 프로그래밍 도구이다. 컴퓨터프로그래밍 언어를 처음 접하는 학생들이 겪는 어려움을 줄여주고자 개발되었다. 따라서 이런 친숙함과 편리함을 위해 다음과 같은 특징을 가지고 있다.

가. 코드블럭 방식

스크래치로 프로그래밍을 하는 방법은 매우 직관적이다. 명령어를 레고 블록처럼 서로 결합하게 되어 있어 명령에 해당되는 블록을 마우스로 끌어다 놓으면 프로그램이 된다. 이 방식은 서로 결합할 수 있는 블록끼리 모아지기 때문에 오류가 적게 발생한다. 또한 색상으로 명령어가 그룹으로 만들어져 있으며 각각의 블록에는 의미전달이 쉽게 될 수 있도록 설명이 적혀 있다. 처음 프로그래밍을 하는 사람도 쉽게 이해할 수 있도록 되어 있다.



<코드블럭을 이용한 프로그래밍>

나. 미디어 조작

스크래치는 다양한 미디어를 쉽게 다룰 수 있다. 이미지, 음악, 소리 같은 미디어를 혼합하여 이벤트에 맞추어 조작할 수 있어서 재미있는 프로그램 작성을 가능하게 한다. 악기연주, 캐릭터 애니메이션, 게임 등을 만들고 축소, 확대와 같은 그래픽 효과를 통해서 흥미로운 프로그램을 작성할 수 있다.

다. 공유

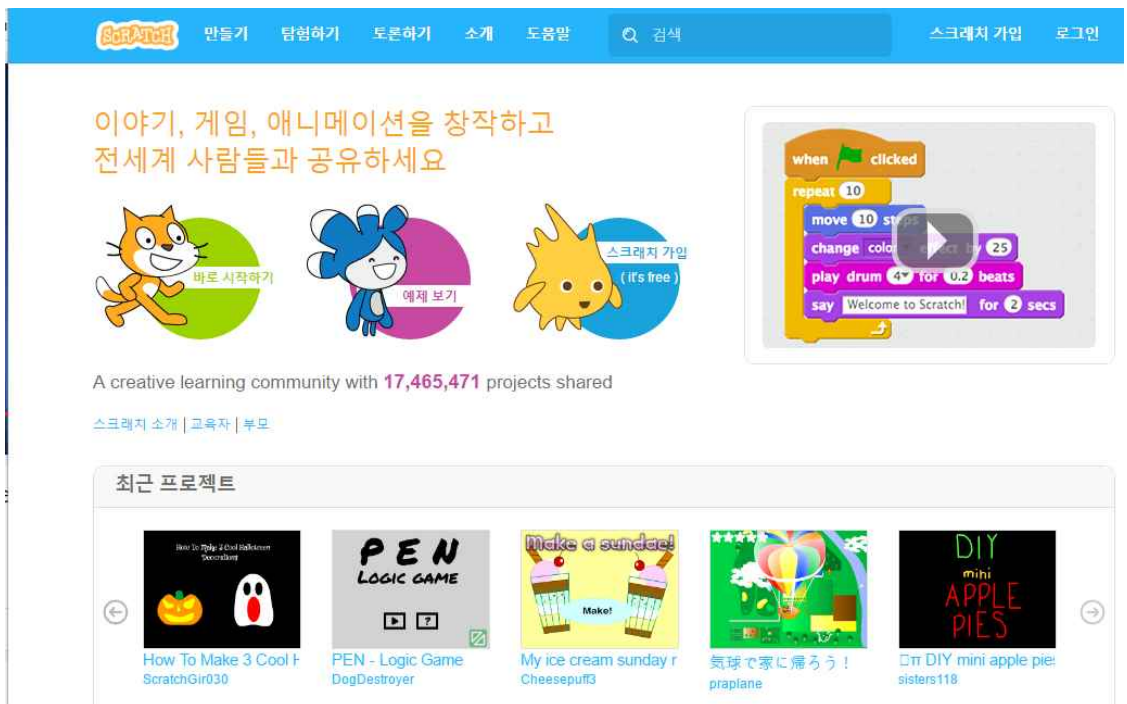
스크래치는 프로그램으로 끝나는 것이 아니라 공유를 할 수 있는 공간을 제공한다. 공식 웹사이트인 <http://scratch.mit.edu/>에서는 사용자들을 위한 장을 제공하고 있다. 다른 사용자가 만든 프로젝트를 내려 받아 동작원리 및 아이디어를 학습할 수도 있으면 자신의 프로그램을 게재하여 전 세계 프로그래머들에게 조언을 구할 수도 있다.

2) 스크래치 사용하기

스크래치는 개발팀에 의해 지속적으로 관리되고 있다. 그리고 인터넷이 연결되는 곳에서는 웹브라우저에서 직접 프로그래밍을 할 수 있도록 지원한다. 인터넷이 지원되지 않을 경우에는 다음과 같이 스크래치 에디터를 내려 받아 실행하도록 설치해야 한다.

(1) 스크래치 공식 사이트에 접속한다.

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



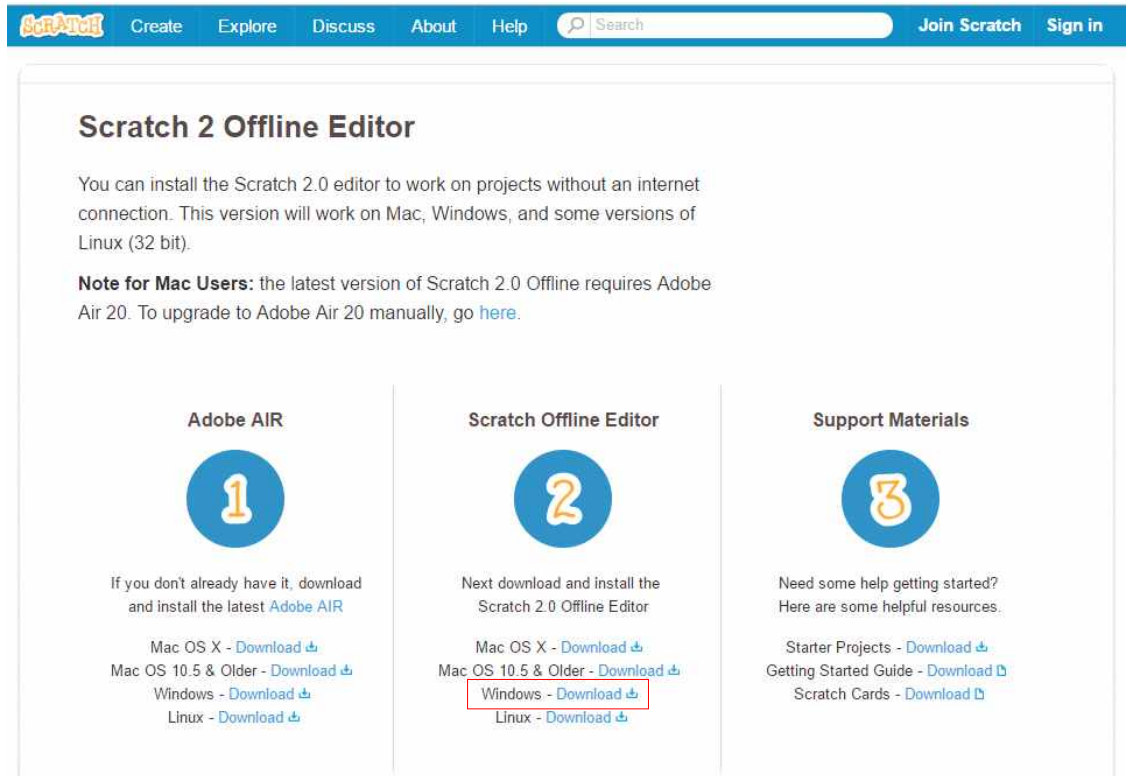
공식홈페이지 아래쪽에 "오프라인에디터"를 선택하면 필요한 프로그램을 내려 받을 수 있는 있다.



<오프라인 에디터 내려 받기>

스크래치 실행을 위해 요구되는 프로그램과 도움이 필요할 때 지원하는 프로그램을 같이 내려 받을 수 있게 되어 있다. 사용하고 있는 컴퓨터 운영체제에 따라 맥, 윈도

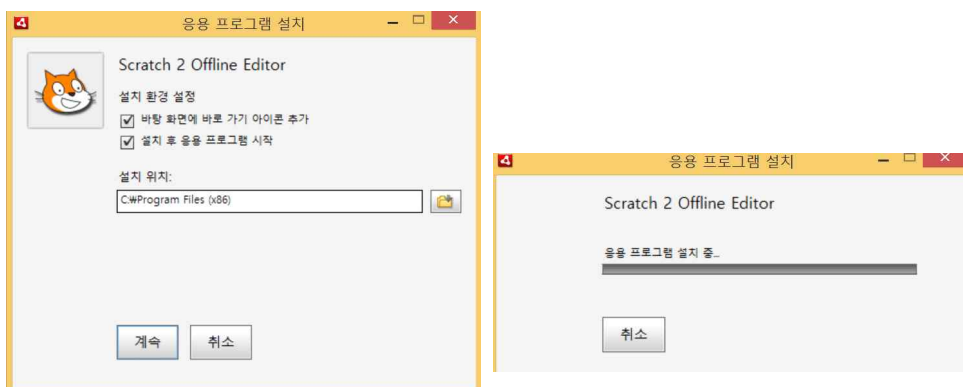
우즈, 리눅스에서 실행되는 프로그램을 지원하고 있다.



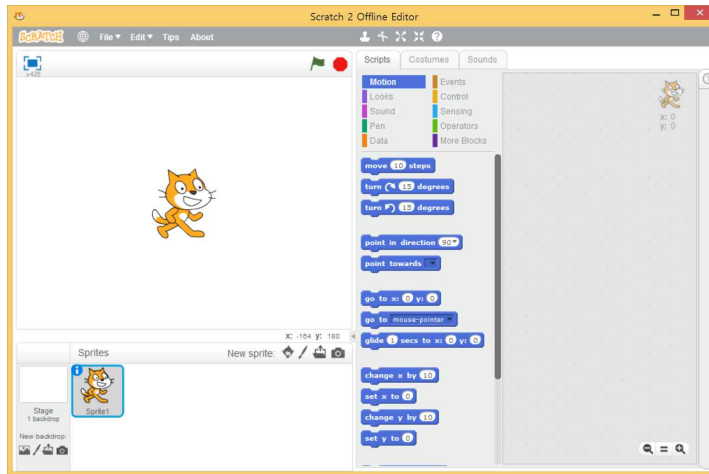
The image shows the Scratch 2 Offline Editor installation guide page. It features a blue header with the Scratch logo and navigation links: Create, Explore, Discuss, About, Help, Search, Join Scratch, and Sign in. The main content is titled "Scratch 2 Offline Editor" and explains that the editor can be installed to work offline on Mac, Windows, and Linux (32 bit). A note for Mac users mentions the requirement for Adobe Air 20. The guide is divided into three numbered steps:

- Adobe AIR**: If you don't already have it, download and install the latest Adobe AIR. Links for Mac OS X, Mac OS 10.5 & Older, Windows, and Linux are provided.
- Scratch Offline Editor**: Next download and install the Scratch 2.0 Offline Editor. Links for Mac OS X, Mac OS 10.5 & Older, Windows (highlighted with a red box), and Linux are provided.
- Support Materials**: Need some help getting started? Here are some helpful resources. Links for Starter Projects, Getting Started Guide, and Scratch Cards are provided.

(2) 내려 받은 파일을 실행하면 다음과 같은 화면이 나온다.



설치가 끝나면 다음과 같이 스크래치 에디터가 실행되어 나타난다.



<스크래치 에디터 실행 화면>

3) 스크래치 사용

(1) 스크래치 공식 사이트에 접속한다.

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

(2) 스크래치 가입

- 메뉴에 있는 스크래치 가입 클릭



- 아래와 같은 내용을 입력하고

스cratch 가입

스cratch 계정은 쉽게 그리고 무료! 로 만들 수 있습니다.

스cratch 사용자 이름 입력

비밀번호 입력

비밀번호 확인

1
2
3
4

다음

스크래치 가입

이 질문들에 대한 응답은 개인 정보로 보호될 것입니다.
이런 정보를 요청하는 이유 ?

생년월일 -월- -년-

성별 ☐ 남자 ☐ 여자 ☐

국가 -국가-



1 2 3 4


- South Korea

스크래치 가입

Enter your email address and we will send you an email to confirm your account.

이메일 주소

이메일 주소 확인



1 2 3 4

- 가입 완료 : 가입한 전자메일에서 확인해야 함

(3) 스크래치 도움말 활용하기

- 스크래치 홈페이지의 아래쪽에 "도움페이지"를 클릭하면 다양한 자료를 접할 수 있다.

스크래치 도움말

스크래치 시작하기



단계별 프로젝트로
스크래치 시작하기



초보자용 프로젝트
탐험하기



스크래치 소개 영상 보
기

스크래치 가이드

스크래치를 배우는데 도움이 될 만한 가이드가 있습니다:



초보자 가이드

스크래치를 처음 접하는 사용자는 이 단계별 가이드(PDF)를 참고하세요.

Download the [English version](#).



스크래치 카드

스크래치 카드에는 사용자가 스크래치에서 해
볼 수 있는 내용이 들어 있습니다.



비디오 튜토리얼

이 비디오 튜토리얼에서는 스크래치에서 그림
을 편집하는 방법, 계급 및 애니메이션을 만들
수 있는 방법 등이 포함되어 있습니다.

스크래치 커뮤니티 탐험하기

탐험하기 스크래처들이 만든 프로젝트와 스튜디오

다음 키워드로 프로젝트를 검색합니다:

스크래치 토론방에서 자기를 소개하세요.

다른 스크래처나 스튜디오의 업데이트 소식을 받고 싶으면 [Join](#) 버튼을 클릭하세요.

스크래치란?



스크래치에 대
해서
부모들을 위한 글
교육자를 위한 글

질문?

다양한 질문에 대한 답변 살펴보
기:

[자주 묻는 질문\(FAQ\)](#)

또는 [토론방 방문하기](#):

- [스크래치에 대한 질문들](#)
- [스크립트 도움말](#)

유용한 자료들

ScratchEd

다양한 교육 이야기, 토론들, 스크
래치 교육과정 가이드와 같은 유용
한 자료를 공유하는 교육자들을 위
한 온라인 커뮤니티

스크래치 위키

스크래치 위키에는 다양한 내용과
수준의 스크래치와 관련된 글과 튜
토리얼 등이 포함되어 있습니다.

Scratch 2 Offline Editor

Download the Scratch 2 offline
editor.

스크래치 1.4 다운로드

이전 버전의 스크래치를 다운로드
받을 수 있는 링크와 정보가 있습
니다.

스크래치 통계

현재까지 스크래치 온라인 커뮤니
티 활동에 대한 통계를 살펴볼 수
있습니다.

스크래치와 하드웨어 세상

Connect your Scratch projects to
the physical world with [MaKey](#)
[MaKey](#), [LEGO WeDo](#), or
[PicoBoard](#).

스크래치 로고

스크래치 로고와 다양한 미디어 파
일을 살펴볼 수 있습니다.

4) 스크래치 인터페이스



스크래치는 스프라이트라는 객체들의 스크립트를 실행하는 방식으로 프로젝트를 수행한다. 따라서 스프라이트에 대한 이해가 필요하다. 스프라이트는 이미지를 기본으로 하고, 모양, 소리, 스크립트의 구성요소를 갖추고 있다. 스프라이트 목록 창에서 추가가 가능하며, 인터페이스 창에서 각각의 스크립트, 모양, 소리를 조작할 수 있다. 모든 스프라이트는 명령어 블록의 조합인 스크립트를 실행함으로써 무대에서 동작한다.

(1) 파일메뉴와 툴바

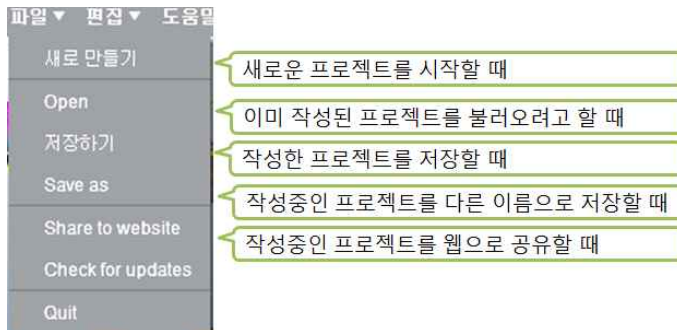
파일메뉴



Scratch 는 스크래치 홈 사이트로 바로 갈 수 있다.

는 제공하는 에디터 프로그램의 언어를 선택할 수 있다. 영어로 되어 있으면 이 버튼을 클릭해서 "한국어"를 선택하면 된다.

파일 ▾ 편집 ▾ 도움말 스크래치란 메뉴에서 파일을 클릭하면 현재 작성 중인 프로젝트에 대해 여러 가지 작업을 할 수 있는 기능을 제공한다. 오프라인에디터에서는 다음과 같이 나타난다.



프로젝트를 저장하게 되면 스크래치2 프로그램에 의해 작성된 것으로 .sb2 라는 파일 확장자 이름을 가진다. 이전 버전은 .sb로 되어 있다.

온라인 에디터는 다음과 같이 나타난다.



툴바(Tool Bar)


스프라이트를 복제하는 버튼이다.



스프라이트를 삭제하는 버튼이다.





스프라이트를 확대/축소하는 버튼이다.

스프라이트 복제와 삭제는 스프라이트 목록에서 오른쪽 버튼을 누르면



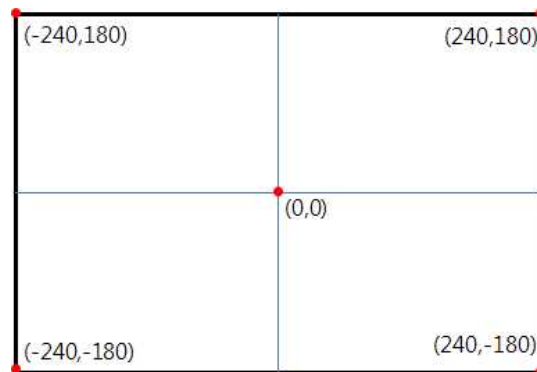
복사, 삭제, 스프라이트를 저장할 수 있는 메뉴가 나타날 때, 원하는 작업을 클릭하면 된다.

시작과 종료


프로젝트를 실행하고 종료하기 위한 버튼이다. 일반적으로 스크래치의 스크립트는 를 클릭해서 시작하도록 프로그래밍 한다. 중간에 스크립트를 중단하고 싶을 때는 를 클릭한다.

무대의 좌표

무대의 마우스가 있는 위치를 **x: 85 y: 176** 와 같이 항상 표시해 준다. 무대의 좌표를 알아 보려면 각 무대 귀퉁이에 마우스를 옮겨보면 알 수 있다.


(2) 스크립트 영역

스크립트 영역은 스크립트 명령을 작성할 수 있는 "스크립트" 탭, 스프라이트의 모양을 편집할 수 있는 "모양" 탭, 스프라이트의 소리를 등록할 수 있는 "소리" 탭이 있다.

스크립트 탭

스크립트는 스프라이트에 명령을 내리는 것이다. 즉 프로그래밍을 하는 곳이다. 여기에는 다음과 같이 10가지의 스크립트 팔레트가 있다.



각 팔레트 안에는 명령블록이 있다.

가. 동작(Motion): 스프라이트의 이동과 방향 전환 관련 명령 블록

나. 형태(Looks): 스프라이트 치장물이나 색상 변경, 말풍선 등 화면에 보이는 모습 제어

다. 소리(Sound): 스프라이트에 등록된 사운드의 재생과 정지, 볼륨, 빠르기 등 제어

라. 펜(Pen): 펜의 색상, 굵기 등 제어

마. 데이터(Data)

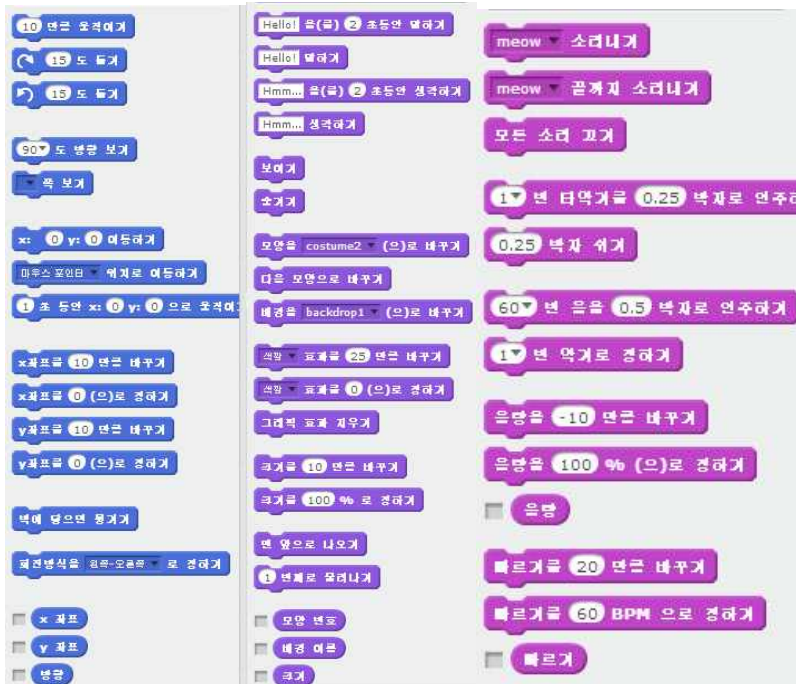
바. 이벤트(Events): 시작과 종료, 이벤트 발생 제어

사. 제어(Control): 스크립트 제어하기 위한 명령

아. 관찰(Sensing): 타이머, 마우스 위치 등 센서에 관한 명령

자. 연산(Operators): 연산관계 명령

차. 추가블록(More Blocks)



```

when green flag clicked
  say [글쎄졌을 때] for 2 sec
  say [스페이스 키를 눌렀을 때] for 2 sec
  say [여 스프라이트를 글쎄졌을 때] for 2 sec
  say [배경에 backdrop1 (으)로 바뀌었을 때] for 2 sec
  say [종말 > 10 일 때] for 2 sec
  say [message1 음(글) 변했을 때] for 2 sec
  say [message1 방송하기] for 2 sec
  say [message1 방송하고 기다리기] for 2 sec
  
```

새로운 변수

변수 이름:

☒ 모든 스프라이트에서 사용 ☐ 이 스프라이트에서만 사용

☐ Cloud variable (stored on server)

확인

취소

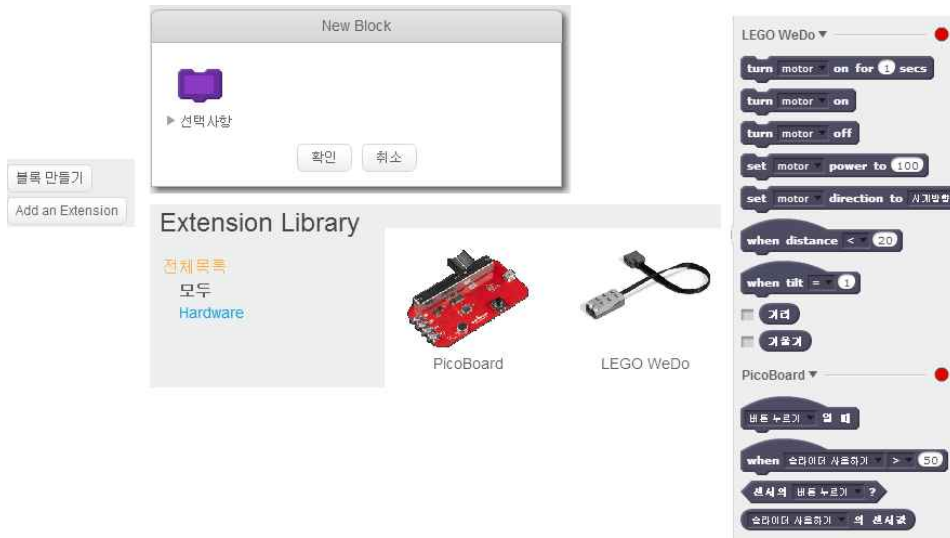
새로운 리스트

리스트 이름:

☒ 모든 스프라이트에서 사용 ☐ 이 스프라이트에서만 사용

확인

취소



2. 도움말을 이용한 스크래치 사용해보기

목표

- 스크래치 기능을 빨리 익힐 수 있다.
- 스크래치 사용법을 빨리 익힐 수 있다.
- 다양한 미디어의 활용을 빠르게 익힐 수 있다.

1) 도움말을 활용한 물체의 표현과 애니메이션 만들기

스크래치는 여러분만의 재미있는 이야기, 애니메이션, 게임, 음악, 작품을 만들 수 있게 도와주는 프로그래밍 언어입니다.



- 무대를 만든다.
- 캐릭터를 움직이게 한다.
- 소리를 넣어 본다.
- 여러 동작을 하는 캐릭터를 활용하여 애니메이션을 보여준다.
- 여러 가지 이미지 및 미디어를 활용해 본다.
- 반복 동작을 만들어 본다.

- 만들어진 프로그램을 어떻게 저장하는 알아본다.

이 책은 스크래치에서 어떻게 작품(프로젝트)을 만드는지 배우게 됩니다.

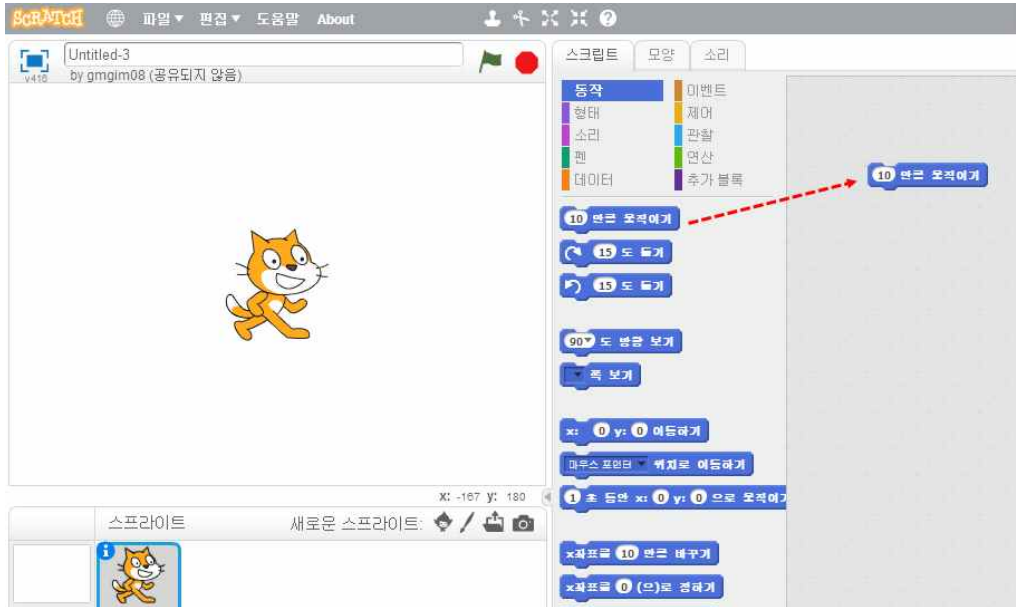


스크래치 에서 만들기 메뉴를 눌러 새로운 프로젝트를 만들어 보세요.



스크래치에 회원가입이 되어 있으면 내가 만든 프로젝트가 저장될 수 있게 로그인 하세요.

(1) 동작하게 만들기



스크립트 지역으로 동작 블록을 끌고 오세요.



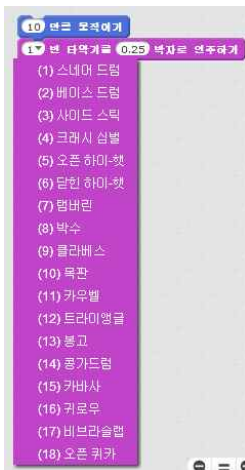
블록을 클릭하여 고양이를 움직여 보세요.

(2) 소리나게 만들기



소리팔레트에서 **타악기 연주하기** 블록을 끌고 와서 **움직이기** 블록에 붙이세요.

10 만들 움직이기
1 번 타악기를 **0.25** 박자로 연주하기 블록을 클릭해서 들어보세요. 소리가 들리지 않는다면 컴퓨터소리가 켜져 있는지 확인하세요.



1번 숫자 옆에 작은 화살표를 누르면 다른 드럼을 선택할 수 있어요.

(3) 춤추게 만들기



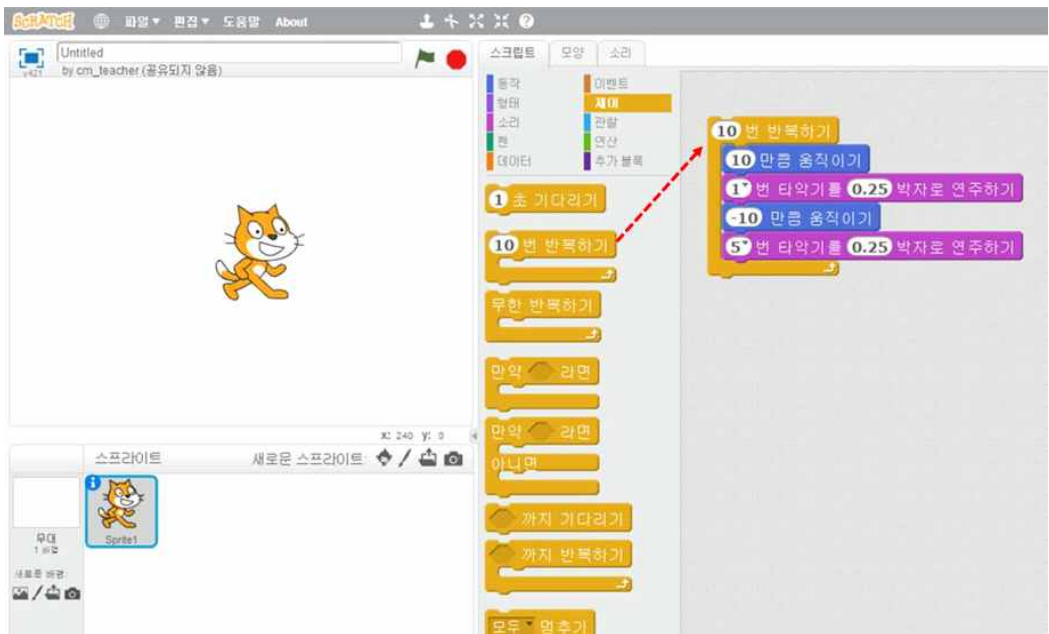
다른 **움직이기** 블록을 더하세요. 움직이기 블록의 숫자를 클릭한 후에 키보드로 빼기(-)기호를 넣어보세요.



스크립트를 실행하기 위해 블록 중 한 곳을 **클릭**해 보세요.

다른 **타악기 연주** 블록을 더한 수에 다른 타악기로 바꿔 보세요.
클릭하여 실행시켜 보세요.

(4) 다시 또 하기



반복하기 블록을 끌고 와서 스크립트 위쪽에 가져다 놓으세요. **반복하기** 블록이 입을 벌려 다른 블록들을 감싸게 됩니다.

블록들을 동시에 이동시키려면 **젤** 위의 블록을 잡아야 합니다.

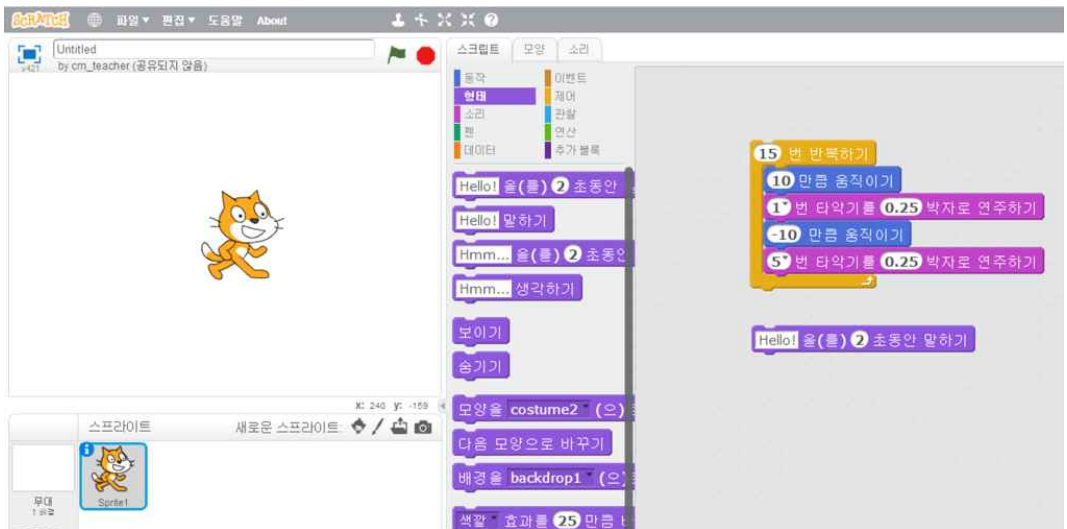


얼마나 많이 반복할지 선택할 수 있습니다.

클릭해서 실행해 보세요.

실행시키려면 아무 블록이나 클릭하세요.

(5) 말하기

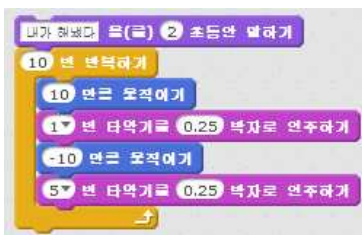


메뉴 탭의 형태팔레트에서 **말하기** 블록을 가져오세요. **말하기** 블록 안쪽 글자부분

을 클릭해서 말을 바꿔 보세요.
보세요.




클릭해서 실행해





말하기 블록을 제일 위에 붙여 보세요.

(6) 녹색 깃발

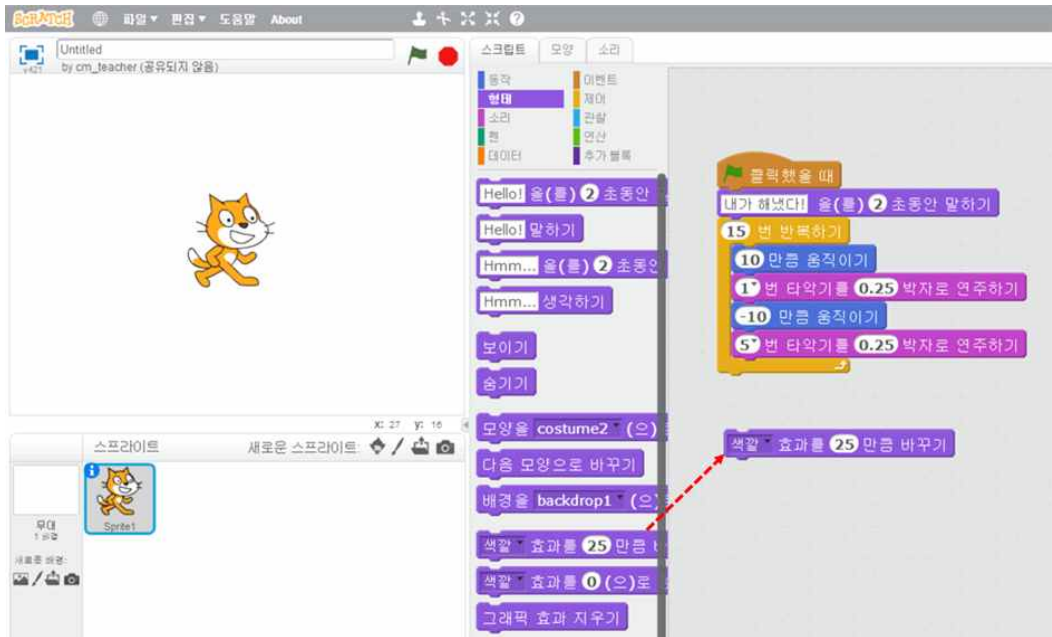


 블록을 꺼내서 스크립트 제일 위에 붙이세요.

녹색  깃발을 클릭할 때 마다 스크립트가 시작됩니다. 동작을 멈추려면 **붉은색**  원모양 버튼을 누르세요.

(7) 색깔 바꾸기

이제 뭔가 다른 것을 시도해 봅시다.



효과 바꾸기 블록을 꺼내세요.

색깔 효과를 25 만큼 바꾸기 어떤 변화가 생길지 클릭해 보세요.

(8) 키보드 누르기




스페이스 키를 눌렀을 때 를 효과 바꾸기 블록 위에 붙여 보세요.

키보드의 스페이스 키를 눌러 보세요.

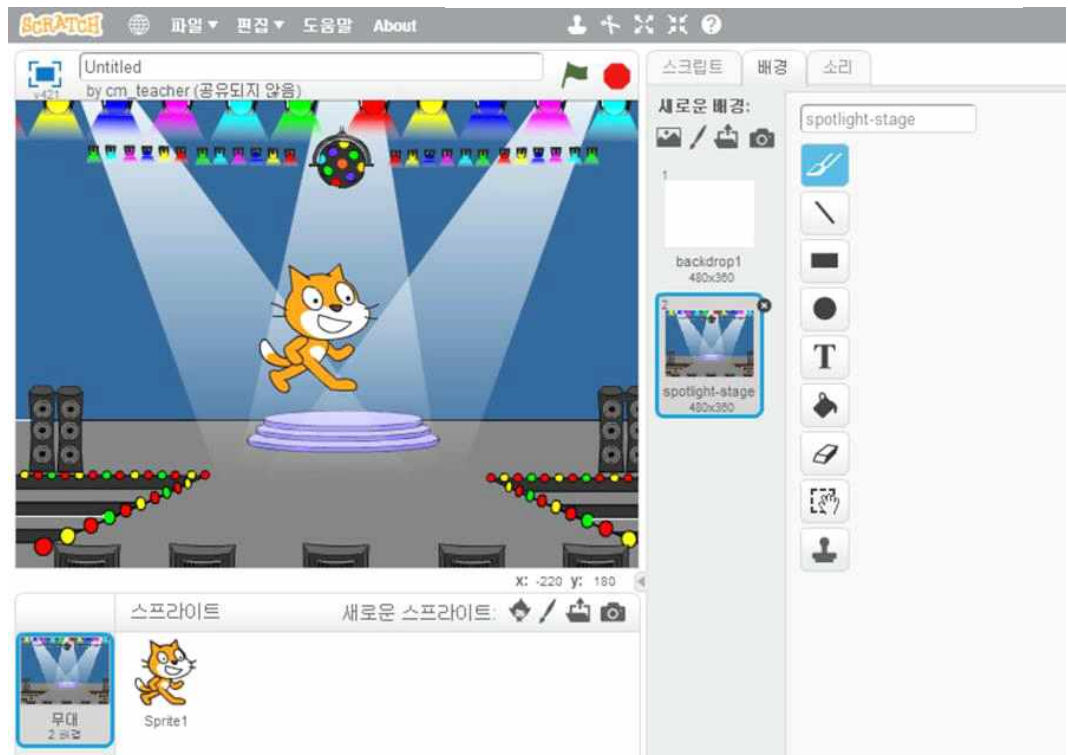


블록 안에 작은 화살표를 누르면 다른 키를 선택할 수 있습니다.

(9) 배경 바꾸기

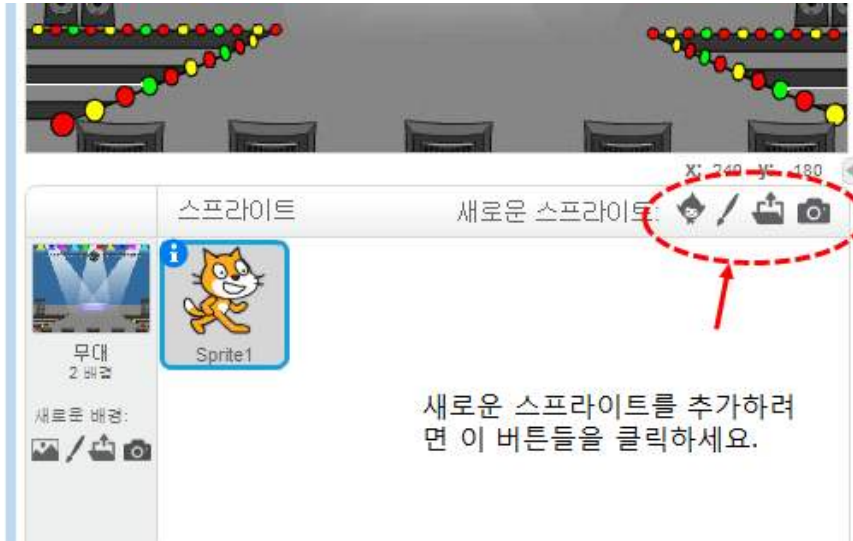
무대(스테이지)에 배경을 추가할 수 있습니다.  을 클릭해서 새로운 배경을 선택하세요.

새로운 배경이 무대에 나타납니다.







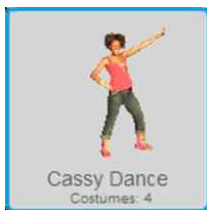
(10) 스프라이트 추가

스크래치에서는 무대에 보이는 캐릭터들을 '스프라이트(Sprite)'라고 부릅니다.
새로운 스프라이트를 더하려면 이 버튼들을 클릭하세요.



새 스프라이트 버튼들:

-  스프라이트 모음에서 선택하기
-  직접 그리기
-  내 PC 그림 중에서 선택하기
-  웹캠에서 사진 찍기



스프라이트를 더하려면 버튼을 클릭하고 전체 목록에서 사람들을 선택한 후 "Cassy Dance"를 선택합니다.

가져온 캐릭터는 마우스로 잡아 끌어서 원하는 곳에 가져다 놓으세요.



(11) 기능 알아보기

이제 스프라이트가 무엇인지 알게 되었을 것입니다. 아래 나오는 것들을 따라 해 보거나 다른 기능들을 자유롭게 살펴보세요.

소리 더하기

소리 탭을 클릭하세요.



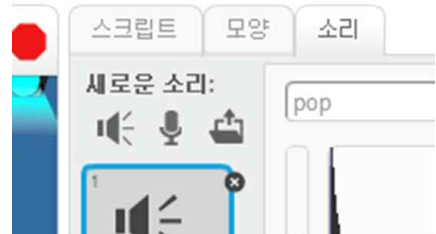
버튼을 클릭하여 소리를 선택하세요.



버튼을 클릭하여 소리를 녹음하세요.



버튼을 클릭하여 내 PC에 있는 소리파일 (mp3, aif, wav 등)을 가져오세요.



스크립트 탭을 클릭하여 소리내기 블록을 가지고 오세요. 블록 안쪽 작은 화살표를 클릭하여 소리를 선택하세요.



모양 바꾸기

스프라이트는 다양한 모양을 가지고 있습니다. 현재 모양을 바꾸려면 모양 탭을 클릭한 후에 다른 모양을 선택하세요.



동작 효과 만들기

스프라이트의 모양을 바꿔서 동작효과를 만들 수 있습니다. 스프라이트 탭을 클릭하세요. 모양을 연속해서 바꿔주는 스크립트를 만드세요.



(12) 도움말



프로젝트 이름을 쓰세요

더 알고 싶다면 도움말을 클릭하세요.

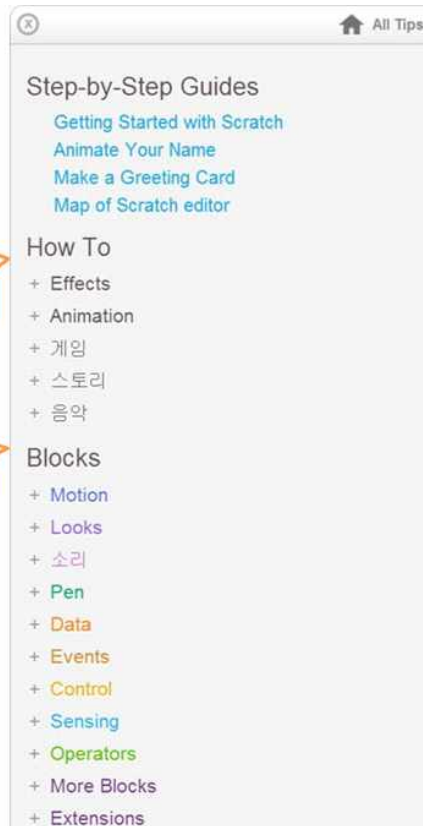


도움말에서 프로젝트에 이용할 수 있는 다양한 예제들을 찾을 수 있습니다.

스크래치의 블록들에 대한 자세한 설명을 볼 수도 있습니다.

도움말에서 프로젝트에 이용할 수 있는 다양한 예제들을 찾을 수 있습니다.

스크래치의 블록들에 대한 자세한 설명을 볼 수 있습니다.



(13) 저장과 공유




로그인

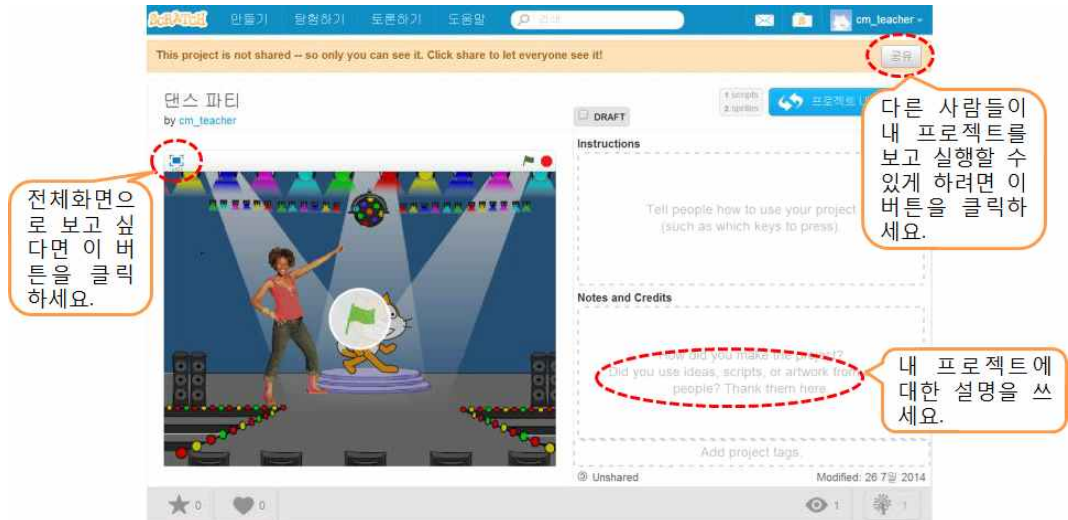
로그인을 하면 프로젝트 파일을 온라인에 저장해 둘 수 있습니다.

(내 PC에 파일을 저장하고 싶다면 파일 메뉴에서 “내 컴퓨터에 프로젝트 다운로드 하기”를 클릭하면 됩니다.)

로그인을 했다면 다음과 같이 현재의 프로젝트를 저장할 수 있습니다.



그리고 다시 작업을 하려고 할 때는 내작업실  을 클릭하세요. 그러면 작업실에 저장해둔 프로젝트들을 볼 수 있다. 작업하고자 하는 프로젝트를 선택하여 스크립트보기 버튼을 누르면 작업환경이 나타난다. 작업중에 이를 실행하고자 한다면 “프로젝트 페이지보기”  공유하기  프로젝트 페이지 보기 를 클릭하면 그러면 다음과 같은 프로젝트 페이지가 나옵니다.



내 프로젝트를 공유하면 다른 사람들이 방문해서 내 프로젝트를 보고 댓글을 남기거나 내 프로젝트를 이용해서 새로운 프로젝트를 만들 수 있습니다.

2) 문제풀이 및 도전문제

- (1) 고양이가 걸어가면서 "야옹" 소리를 내게 해보자.
- (2) "s"를 누르면 고양이가 움직이고 "스페이스"를 누르면 멈추게 해보자.
- (3) 앞 도움말 예제에서 여자가 춤을 추는 동작을 완성하자.
- (4) 강아지가 마우스가 클릭할 때 마다 드럼소리가 울리면서 앞으로 뛰어가는 애니메이션을 만들어 보자.

3) 프로그래밍적인 사고하기

배가 부르다. 사료를 먹을지 사냥을 할지 고민한다. 쥐 사냥한다.
잠을 잔다. 배가 고프다.

(1) 고양이가 하는 행동을 생각해보고 그 순서를 적어보자.

(2) 이것을 스크래치로 표현해보자. (예)



(3) 위의 흐름에서 고양이가 쥐를 만났을 때의 상황을 같이 묘사해 보자.

- 쥐가 화면에 나타나야 한다.
- 쥐와 고양이가 대화를 나눈다.
- 쥐: 안녕하세요(후들 후들). 고양이: 안녕하지 못해. 난 배고파. 미안하지만 내 식사가 되어 주어야겠다.
- 쥐가 화면에서 사라져야 한다.

(4) 이와 같은 방법을 응용하여 자신의 하루 일과를 표현해 보자.

- 하루의 일과를 나열하기
- 반복적인 일도 표현해보기
- 자신의 캐릭터를 만들어 보기
- 아침시간과 일과시간, 그리고 저녁시간으로 나누어 배경을 바꾸어 보기
- 등장인물을 부모님, 친구, 선생님을 만들어서 대화식으로 장면을 표현해 보기.

NOTE

3. 프로그래밍

1) 생각열기

- 가. 밖에서 운동을 하고 들어온 아이에게 엄마는 어떻게 반응하나?
 나. 로봇이 있다면 운동을 하고 들어온 아이에게 어떤 반응을 하도록 해야 할까?
 다. 로봇이 엄마처럼 반응하려면 어떤 기능을 가지고 있어야 할까?
 라. 그 기능들이 현재 기술로 가능할까?

2) 지능.인공지능.프로그래밍

- 가. 사람의 지능을 컴퓨터에서 가능하도록 하는 것이 인공지능이다.
 나. 인공지능 기술의 저변에는 기본적인 제어 기능을 가지고 있다.
 다. 그 제어 기능을 표현한 것이 프로그래밍이다.

| | 지능적 기능 | 프로그래밍적 기능 |
|-------------|---|---|
| 얼굴을 보고 판단하기 | 땀-운동-갈증-물 | x |
| 물 가져오기 | 물이 있는 곳:냉장고 냉장고 위치 파악 나의 위치 파악 냉장고가 있는 곳으로 이동 냉장고에서 물을 꺼냄 마실 컵을 가지고 옴 물을 따라 줌 | 물이 있는 곳:냉장고 =>물이 필요한 것 명시(알려줌) |
| | | 냉장고 위치 파악 =>나의 위치에 대비한 냉장고 위치 알려줌 |
| | | 나의 위치 파악 =>미리 알려주어 함 |
| | | 냉장고가 있는 곳으로 이동 =>(예) 90도 좌회전->3미터 이동->90도 우회전 |
| | | 냉장고에서 물을 꺼냄 =>(예)오른 팔을 90도 올림->손으로 손잡이 잡음->0.01마력의 힘으로 후진 10cm 당김->물이 어디 있는지 알려줌->물을 꺼냄->냉장고문을 잡고 다시 전방으로 0.001마력의 힘으로 밟 |
| | | 마실 컵을 가지고 옴 =>컵이 필요하다는 것을 알림.->컵이 있는 위치를 알림(예. 식탁위)->컵을 잡고 ->앞선 이동 경로를 역으로 알려준 명령대로 움직임 |
| | | 물을 따라 줌 =>컵 든 손을 90도 들고->물병이 든 손을 들어올림->병을 기우리게 함->컵에 물이 차면 병을 역으로 돌림 ... |

4. 제어하기

프로그래밍에서 제어는 가장 핵심적인 요소이다. 이는 생각의 흐름을 표현하는 것으로 3가지 방법이 있다. (1) 순차적인 제어 (2) 조건제어 (3) 반복제어 이다.

(1) 순차적인 제어

- 어떤 일을 할 때의 순서를 그대로 표현하는 방법이다.
- 순차적인 작업의 나열은 앞 뒤의 순서가 중요할 경우도 있고 순서에 상관없는 일의 경우도 있다.
- 순서가 중요한 작업은 반드시 우선순서를 지켜서 블록을 배치해야 한다.

(2) 조건제어

- 주어진 조건을 만족할 때와 그렇지 못할 경우를 나누어서 작업을 하도록 제어한다.
- 스크래치에서 제공하는 조건제어는 다음과 같은 2가지 경우이다.

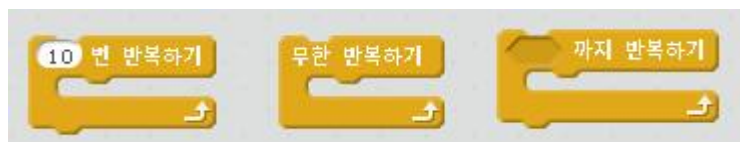


- 조건은 연산 카테고리 있는 관계연산자(<,=,>,그리고, 또는, 가(이) 아니다)가 올 수 있다.



(3) 반복제어

- 반복제어는 조건이나 주어진 범위내에서 주어진 작업을 반복적으로 할 수 있도록 제어한다.
- 이때 주어진 조건은 만족(참일 때)할 때 이고, 범위도 그 참/거짓을 판별할 수 있도록 주어진다.




- 반복할 수 있는 범위에는 아래와 같은 사칙연산 뿐 만 아니라 다양한 형태로 그 범위를 지정할 수 있다.

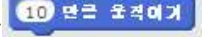


1) 몸 풀기: 단순한 동작 제어 및 반복제어


가. 문제: 고양이가 현재 방향으로 100걸음 이동했다가 방향을 바꾸어 제자리로 돌아오게 하자.


(1) 새로운 프로젝트(웹-만들기)

(2) 스크립트 이벤트팔레트-  드래그

(3) 동작팔레트-  드래그- 값을 100으로 수정

(4) 동작팔레트-  드래그 - 값을 180으로 수정

(5) 잠시 쉬었다 감. 제어 팔레트 -  드래그

(6) 동작팔레트-  드래그- 값을 100으로 수정

(7) 동작 확인해 보기


나. 응용문제: 앵무새가 벽을 만나면 돌아서 날아오게 하기


- 동작: 앵무새가 날아가는 것을 표현해야 한다.

- 벽을 만나면 되돌아오는 동작을 만들어야 한다.

(1) 원래의 고양이 스프라이트를 오른쪽 마우스를 클릭하여 "삭제"한다.



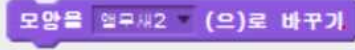
(2) 새로운 스프라이트 버튼  를 클릭하여 동물 카테고리에서 앵무새를 선택한다. 선택된 앵무새 스프라이트를 **모양** 탭을 선택하면 앵무새가 2가지가 나온다. 먼저 나온 앵무새 스프라이트를 parrot-a를 "앵무새1"로 이름을 고치고 두 번째 스프라이트를 parrot-b를 "앵무새2"로 고친다.

(3) 스크립트 이벤트팔레트-  드래그


(4) 제어 팔레트에서 무한반복  루프 블록을 드래그

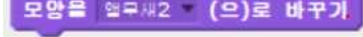
(5) 무한반복 블록 안에 모양을 제어하는 블록을

드래그.




블록 안에서 오른쪽 단추버튼을 클릭하여 원하는 “앵무새1” 스프라이트를 선택한다.


(6) 동작 후에 1초간 기다리는 제어팔레트  블록 드래그

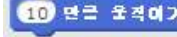
(7) 두 번째 스프라이트 보이기 위해  블록 드래그.

스프라이트를 선택하는 단추를 눌러 “앵무새2” 스프라이트 선택.

(8) 동작 후에 1초간 기다리는 제어팔레트에서  블록 드래그

(9) 동작 확인 해보기. 그리고 이제 동작과 함께 앵무새가 움직이게 해 보자.

(10) 먼저, 이벤트팔레트-  드래그 하고, 무한루프 블록을 드래그 한다.

(11) 앞선 예제와 같이 동작 팔레트에서  블록을 드래그



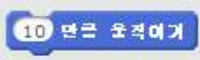
(12) 여기까지 작성된 스크립트를 실행해 보자.

(13) 벽에 부딪히면 튕기기를 위한 제어  블록 드래그



다. 스프라이트의 새로운 배경에서 숲을 선택하여 배경을 꾸며보자.

라. 바다 백사장 위를 게가 마우스를 따라 움직이게 해보자.

- (1) 새로운 스프라이트에서 게를 선택. 게의 모양을 2가지 이름을 넣는다.
- (2) 배경은 바닷가로 선택한다.
- (3) 무한루프 블록을 가지고 와서 앵무새 애니메이션 방법과 같이 게의 움직임을 애니메이션 한다. (두 스프라이트를 사용하는 대신  로 해도 됨.
- (4) 또 다른 루프 블록을 드래그 함. 무한반복 블록에 마우스 방향을 따라가는 블록  을 드래그 함.
- (5)  으로 스프라이트 움직이게 함.

마. 도전문제: 나비와 박쥐를 그림을 이용해 박쥐가 나비를 따라 가는 애니메이션 동작을 만들어 보자.

2) 반복하기

가. 종료조건을 가지는 반복: 앵무새나 게가 벽에 부딪히면 동작을 멈추게 해보자.



- (1) 무한루프 블록이 아니라 **조건루프** 블록을 선택하고 조건은 **관찰팔레트** 안에서 스프라이트가 어디에 닿았는지 **에 닿았는가?** 검사하는 블록으로 제어할 수 있다.

나. 문제: 나비 2마리가 불규칙하게 날아다니는 것을 만들어 보자.

- 동작: 스프라이트가 설정한 각도만큼 회전하게 하기
- 연산: * 설정한 범위 안의 난수 만들기
 - * 왼쪽의 값과 오른쪽 값이 같은지 알아보기
- 형태: * 스프라이트가 다른 스프라이트보다 앞에 보이게 하기
 - * 스프라이트가 다른 스프라이트보다 설정한 값만큼 뒤로 물러나 보이게 하기
- 제어: if(조건) then (동작) else (동작)



- (1) 나비가 경계점에 오면 방향을 돌리게 한다. 나비는 스프라이트 정보에서 좌우로만 움직이게 한다. 나비 날개 짓을 동작으로 보이게 한다.



(2) 나비가 불규칙하게 날게 한다. 날아가는 시간도 불규칙하게 한다.



(3) 배경을 언덕이 있는 것을 선택하고, 나무 스프라이트를 무대에 가져온다. 가져온 나무와 나비가 만났을 때, 나비가 나무 앞으로 혹은 뒤로 날아다니는 효과를 표현한다.



다. 도전문제: 어항에 물고기 3마리가 자유롭게 헤엄치는 동작을 표현해 보자.

- 이때 수초를 심어서 물고기가 수초 앞뒤로 다니도록 한다.
- 또한 물고기가 멀리 가는 느낌이 나도록 물고기 크기를 변경해 보자.
- 물방울을 만들어 올라오게 만들어 보자.
- 물고기가 천천히, 그리고 빠르게 등 불규칙하게 움직이게 해보자.
- 물방울이 다른 크기로 여러 개 발생되도록 해보자.

3) 변수 만들고 사용하기

가. 문제: 경주게임 만들기

- 등장인물: 심판, 박쥐, 앵무새
- 배경: 도착지점
- 게임내용
 - * 배경: 새로운 스프라이트 그리기를 이용해 도착지점을 나타내는 막대기(이름저장-도착지점)를 만든다.
 - * 심판: 출발을 위해 심판이 보이게 하고 5,4,3,2,1을 1초간 말하고 "출발"을 방송과 동시에 "출발"이라고 0.2초간 말한다. 그리고 0.2초 정도 지나고 사라진다.
 - * 박쥐/앵무새: 각각 날아가는 동작을 만들고, 처음 출발지점을 (-200, 0), (-200,50) 각각 이동시킨다. "출발" 신호와 동시에 앞에서 만든 도착지점에 도달할 때까지 이동하면 된다. 먼저 도착한 새는 도착했음을 방송한다.
 - * 심판: 심판은 그 방송을 듣고 모습을 나타내어 먼저 도착한 새를 "앵무새 승!" 혹은 "박쥐 승!"이라고 말하고 동작을 종료한다.

나. 도전문제: 나비를 잡아먹는 앵무새

- 등장인물: 나비3마리, 앵무새
- 나비: 나비는 불규칙적으로 날아다닌다. 크기는 실감 있게 작게 한다.
- 앵무새: 앵무새는 50초 동안 나비를 잡으려 다닌다. 시간 측정을 위해 타이머를 설치해야 한다.
- 게임기능:
 - * 앵무새는 나비를 향해 가야 한다.
 - * 앵무새가 나비를 잡으면 나비는 사라져야 한다.
 - * 나비는 일정한 거리(50)이내에 접근하면 앵무새가 날아오는 반대방향으로 도망가야 한다.
 - * 여러 마리의 나비를 쫓아다니는 앵무새가 하나의 앵무새를 잡으면 나비는 사라진다.

4) 예제로 풀어보는 프로그래밍 제어

가. 문제-벽돌깨기 게임 : 1단계

- 정통적인 게임 중에 하나인 벽돌깨기 게임을 만들어 보자.
- 판을 좌우로 화살표로 이동한다.
- 공이 목표물을 맞추면 점수가 올라가도록 한다.
- 공은 벽을 만나면 튕겨나온다.
- 공이 판을 만나면 입사각 대비 반사각을 계산하여 공의 방향을 바꾸어준다.
- 공이 바닥에 닿으면 게임이 끝난다.

(1) 먼저 필요한 스프라이트를 만든다.

- 무대 : 저장소에서 적당한 배경을 가지고 온다.
- 움직이는 공 : 저장소에서 선택한다.
- 목표물 : 저장소에서 적당한 목표물을 가지고 온다.
- 움직이는 판 : 그림판에서 그린다. 그리고 적당한 아래쪽에 위치한다.
- 바닥 : 긴 막대를 만들어 무대의 맨 아래쪽에 위치시킨다.
- 각각의 스프라이트를 이름을 부여한다.

(2) 판의 움직임을 위한 스크립트 만들기

- <이벤트>카테고리에서 시작버튼을 클릭했을 때를 가지고 온다.
- 시작위치를 아래쪽에 적당한 위치로 초기화 한다. <동작>카테고리에서 판의 위치를 정한다.
- 판의 방향을 지정하기 위해 <데이터>카테고리에서 (변수만들기)를 선택하여 변수 "판의방향"을 만든다. 이 변수의 값은 "준비/오른쪽/왼쪽"으로 주어진다.
- <데이터>카테고리에서 변수의 값을 초기화 한다. "판의방향"을 "준비" 상태로 한다.
- 판의 움직임은 게임이 진행되는 동안 무한히 반복됨으로 <제어>카테고리에서 무한 반복하기를 가지고 온다.
- <제어>카테고리 내에서 판의 움직임을 좌/우 방향키에 따라 움직이므로 이에 대한 제어블럭을 가지고 온다. 즉 "판의방향"이 "오른쪽"이면 판의 위치를 변경해야 한다. 조건블럭을 가지고 와서 조건제어를 위해 <연산>카테고리에서 ($=$)블럭을 가지고 온다.
- 그리고 판의 위치를 변경하기 위해 <동작>카테고리에서 <x좌표 —만큼 바꾸기> 블럭으로 좌표 변경한다. x좌표를 +5 만큼 움직인다.
- "판의방향"이 "왼쪽"일때는 같은 방법으로 구성한다. x좌표를 -5 만큼 움직

인다.

- <이벤트>카테고리에서 이벤트 발생할 때의 블록에서 "오른쪽 화살표"키를 눌렀을 때를 선택하고 <데이터>카테고리에서 "판의방향"를 오른쪽으로 정한다.
- "왼쪽 화살표"키를 눌렀을 때도 같은 방법으로 구성한다.



(3) 공의 움직임을 위한 스크립트 만들기

- <이벤트>카테고리에서 시작버튼을 클릭했을 때를 가지고 온다.
- 공의 위치를 시작위치를 중앙(0,0)에 오도록 한다.
- 공이 <바닥>에 닿을 때까지 지속적으로 움직이게 만들기 위해서 <제어>카테고리에서 <...까지 반복하기>하기 블록을 가지고 온다.
- 공은 5만큼 움직이게 만들기 위해 <동작>카테고리에서 <...만큼 움직이기>블록을 가지고 반복제어 안에 둔다.
- 공은 벽에 부딪히면 튕긴다. <동작>카테고리에서 <벽에 닿으면 튕기기>블록을 가지고 온다.
- 공이 판에 닿으면 입사각 대비 반사각을 계산해야 한다. 이는 목표물에도 같이 적용되어야 하므로 "방송"기능을 사용한다.
- 공이 판에 닿았는지 여부는 조건제어를 이용한다. <제어>카테고리에서 <만약 ..라면> 블록을 가지고 온다. 조건에 <판에 닿았는가>를 삽입하고 참이면 "충돌"이라는 방송을 하게 한다. <데이터>카테고리에서 <방송하기>블록에서 새로운 메시지로 "충돌"를 만든다.

- <이벤트>카테고리에서 <"충돌"(방송)을 받았을때> 반사각은 계산하여 공의 위치를 변경한다. <동작>카테고리에서 <..도 방향 보기>를 위치시킨다.
- 반사각은 "180-(방향)"이다. <동작>카테고리에 공의 스프라이트 방향을 나타내는 <방향>블럭이 있다.
- 효과를 위해 공이 판에서 튕기면 <pop 재생하기>로 소리를 낸다.
- 공이 바닥에 닿으면, 즉, 반복제어 블록이 "거짓"으로 끝나면 <제어>카테고리에서 <모두 멈추기>블럭으로 모든 스프라이트를 멈추게 한다.



(4) 지금까지 완성된 게임을 실행해 보자.

나. 벽돌깨기 게임 : 2단계

- 공이 판에 튕기면 공의 속도가 조금씩 빨라진다.
- 목표물을 맞추면 점수가 올라간다.
- 목표물 충돌을 판과의 충돌과 구분하여 방송한다.

(1) 공의 움직임을 위한 스크립트 변경하기

- 공의 속도를 나타내는 변수 "공속도"를 만든다.
- 점수를 나타내는 변수 "점수"를 만든다.

- "점수"를 0으로 초기화 한다.
- "공속도"를 초기화 5로 한다.
- "목표물충돌" 방송을 받았을 때, 점수를 +10점으로 증가한다.



(2) 2단계 실행해 보자.

다. 벽돌깨기 게임 : 3단계

- 목표물을 여러 개 만든다. 같은 목표물을 복제하여 만든다. 복제할 동안 목표물을 보이지 않게 한다.
- 목표물의 위치를 랜덤하게 배치한다.
- 복제가 완료되면 목표물을 보이게 하고 공과의 충돌을 감지한다.
- 충돌이 감지되면 목표물 수를 감소시키고 목표물을 제거한다.
- 필요한 변수만 보이게 한다.

(1) 공의 움직임을 위한 스크립트 변경하기

- 공에 대한 충돌에서 목표물과의 충돌여부를 제거한다. 이 충돌은 목표물에서 공과의 충돌로 변경하여 구현한다.

(2) 목표물을 위한 스크립트 작성하기

- 목표물의 수를 나타내기 위해 변수 "목표물수"를 생성한다.

- 목표물수를 0으로 하고 <숨기기>블럭을 가지고 온다.
- 10개의 목표물을 만들기 위해 <..번 반복하기>블럭을 가지고 온다.
- 반복제어에서 하나의 목표물을 만들 때(복제) 마다 "목표물수"를 증가시킨다.
- 생성되는 목표물을 랜덤하게 위치하게 한다. y축은 화면의 상단에 나타날 수 있도록 제한을 둔다.
- 목표물 복제는 <제어>카테고리에 있는 <...복제하기> 블록이다.
- <이벤트>카테고리에서, 복제된 목표물이 생성되면 <보이기>로 목표물을 보이게 한다.
- 목표물이 <공과 닿을 때 까지 기다>린다.
- 목표물에 닿으면 "목표물충돌" 방송을 한다. 이 방송으로 공스크립트에 있는 이벤트가 작동한다.
- 목표물수를 하나 감소시키고 목표물을 제거한다.



공 스크립트



목표물 스크립트

(3) 3번째 단계 실행해보기

NOTE

5. 연산해 보기

1) 간단한 연산해 보기

가. 문제: 간단한 더하기 문제를 출제하고 사용자로부터 답을 기다린다. 답이 맞으면 "맞았습니다"라고 하고 틀리면 "틀렸습니다"라고 한다.

나. [문제 가]에서 계속해서 문제를 풀 수 있도록 다음 문제를 내도록 해보자.

다. [문제 나]에서 10문제까지 풀고 맞는 항의 개수를 알려주는 프로그램을 작성해 보자.

라. 도전문제: 문제를 더하기뿐만 아니라 뺄셈, 곱셈, 나눗셈도 같이 나올 수 있는 프로그램을 작성해 보자.

2) 반복 연산

가. 문제: 어떤 수를 입력받아서 1부터 그 수까지 더하는 프로그램을 작성해 보자.

나. [문제 가]에서 홀수와 짝수의 합을 구해보자.

다. 도전문제: 구구단을 작성해 보자.

3) 재귀호출(Recursion)

가. 재귀적 방법

재귀호출이란 자기 자신을 참조하거나 호출하여 문제를 해결하는 방법을 말한다. 이를 가장 잘 표현한 것이 점화관계이다. 그 예로서 팩토리얼 $fact(n)$ 을 나타낼 때, 다음과 같이 두 가지 방법으로 구할 수 있다.

(1) 반복적 방법 : $3! = 3 \times 2 \times 1$

(2) 재귀적 방법 : $3! = 3 \times 2! \leftarrow 2! = 2 \times 1!$

따라서 $n!$ 은 $(n-1)!$ 값을 구하면서 단계적으로 진행한다.

단계1 : $fact(n)$ 이 수행되면 내부에서 $fact(n-1)$ 을 호출하고

단계2 : 다시 내부적으로 $fact(n-2)$ 를 호출 수행

단계3. fact(1)=1 까지 계산

나. 프로그램

- 필요한 변수

- 입력을 위한 변수 : input. 예. 5!를 구하고자 한다면 5가 input의 값이 됨
- 팩토리얼 결과값 : output
- 내부적으로 입력값을 하나씩 감소할 때 필요한 변수 : mid



다. 도전문제 : 피보나치 수열의 값을 알려주는 프로그램을 작성해 보자. 피보나치수열을 다음과 같이 생성된다. $F_1 = 1, F_2 = 1, F_n = F_{n-2} + F_{n-1} (n \geq 3)$

NOTE