

성영훈 지음

KS
한국 과학

인사말

프로그래밍에서 중요한 것 중의 하나는 자신이 생각하는 아이디어를 순서에 따라 논리적으로 실행할 수 있는 알고리즘을 구현하는 것입니다.

엔트리와 같은 교육용 블록 프로그래밍을 처음 접하는 사람들은 하나의 블록으로 실행하는 프로그램은 쉽게 이해하나 여러 가지 블록을 하나의 오브젝트 또는 여러 가지 오브젝트에 동시에 적용하는 것은 많이 어려워합니다.

이 책에서는 제시된 문제를 해결하기 위해 필요한 요소들을 생각하고 분류하고 더 작은 요소로 나누어 보는 연습을 할 수 있도록 요소들을 시각화하여 표현하고 있습니다. 이를 통해서 학습자는 전체적으로 만들고자 하는 흐름을 한 눈에 파악할 수 있고 여러 가지 오브젝트, 이벤트들이 동시에 진행되어도 어떠한 흐름으로 진행되는지 보다 쉽게 파악할 수 있습니다. 또한 가상의 이야기를 챗터 마지막 부분에 추가하여 보다 재미있게 학습할 수 있도록 하였습니다.

창의적이고 논리적인 사고를 향상시키기 위해서는 생각을 떠올리는 방법, 그 생각을 자신의 목적에 맞게 과정별로 제시하는 방법을 배우는 것이 무엇보다 중요합니다.

특히 저자의 경험으로는 생각을 떠올리는 방법, 즉 새로운 아이디어는 보다 자유로운 환경에서 더욱 잘 떠오른다고 생각합니다.

따라서 평소 강가를 걷거나 바다를 보고 느끼는 연습을 많이 해보기를 바랍니다. 또한 예술적 감각을 높이기 위해 악기를 연주한다든지 그림을 그려보는 활동을 많이 해 보기를 권합니다.

끝으로 이 책을 쓰기 위해 영감과 지혜를 주신 나의 하나님께 감사드리고 사랑하는 현, 우, 은 그리고 작은 것까지 꼼꼼히 짚어주는 아내에게 감사의 말을 전합니다.

책 구성과 배움 순서

학습 수준

이 책은 기본적인 엔트리 프로그래밍, 초코파이보드 학습을 통해 기초적인 블록 프로그래밍이 가능한 사람을 대상으로 하고 있으나 챕터별로 자세한 설명을 추가하고 있으므로 코딩을 처음 배우는 분들도 학습이 가능합니다.

책의 구성

★ 문제가 무엇인가요?

WHAT'S PROBLEM?

챕터에서 제시하는 문제를 해결하기 위해서 필요한 정보들을 제시하였습니다.

★ 문제를 분해해 봅시다.

DECOMPOSITION

제시된 정보와 요소들을 기반으로 프로그램 구현에 필요한 오브젝트와 이벤트들에 대해서 상세히 학습합니다. 또한 알고리즘 구현을 위해 이벤트들을 시각적으로 표시하여 학습자들이 전체적인 알고리즘을 시각적으로 인지할 수 있도록 구성하였습니다.

★ 하나씩 연결해 봅시다.

STICK TOGETHER

오브젝트에 따른 이벤트들을 연결하여 작성한 알고리즘을 실제 블록 프로그래밍으로 구현하는 방법에 대해서 설명하고 있습니다.

[SUMMARY 써머리~] 챕터에서 다루는 주요한 학습내용을 다시금 정리하였습니다.

★ 미션을 해결해 봅시다.

MISSION

이전 단계에서 배운 학습요소들을 가지고 가상의 재미있는 스토리를 기반으로 실제 학교현장, 사랑하는 자녀, 방과후 학생, 프로그램을 처음 경험하는 사람들과 함께 학습할 수 있는 미션을 제시하였습니다.

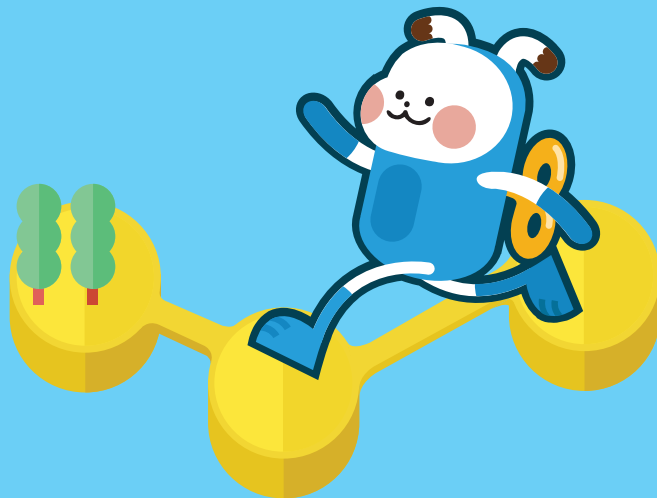
목 차

01. 우주선을 띄워라	7
엔트리와 초코파이보드 다루기	
	
02. 태양계 탐험장비	31
오브젝트, LED 다루기	
	
03. 우주를 줄게	55
터치, 센서블록 활용하기	
	
04. 잃어버린 기억	87
조이스틱 다루기	
	
부록1. 기초 배우기	113
엔트리와 초코파이보드 알아보기	
	
부록 2. 심심풀이 확장블록	141
	

01

우주선을 띄워라 !

엔트리와 초코파이보드 다루기





문제가 무엇인가요?

‘으앙~’

어디선가 아기의 울음소리가 들려옵니다. 엄마는 이때 어떤 생각을 하게 될까요? 아기의 울음소리 크기를 들어보기도 하고 얼굴 표정을 보면서 찡그리고 있는지 웃고 있는지를 살펴보기도 할 것입니다.

아니면 아기의 손 모양을 보거나 만져보기도 하고 아기의 배를 만져보고 혹시 배탈이 났는지도

살펴볼 것입니다. 그렇다면 이러한 장면 속에서 어떤 중요한 원리가 숨어있을까요?

해답은 이러한 문제가 발생했을 경우 우리가 취하는 태도에 있습니다. 하나는 아기의 입장이고 다른 하나는 엄마의 입장입니다. 컴퓨터, 코딩, 프로그래밍 얘기는 시작하지 않고 뜬금없이 아기 울음소리일까요?



좀 더 살펴봅시다.

[아기의 입장]

2~3살 된 아기는 엄마가 어디에 있는지 늘 찾습니다. 놀이를 하거나 밥을 먹거나 잠을 자거나 하는 일상의 모든 순간에서 말입니다. 왜냐하면 엄마는 아기가 필요로 하는 모든 것을 제공하는 존재, 바로 생명의 근원이라고 볼 수 있기 때문입니다.

그렇다면 이제 아기가 엄마를 찾아보는 순간들을 다시 한 번 절차대로 생각해 봅시다.

- 아기는 인형을 가지고 놀다가 뱃속에서 배고프다는 신호를 느낍니다.
- 배고픔을 느끼면 먹을 것이 있는지 눈으로 찾습니다.
- 먹을 것을 발견하지 못했을 때는 어떻게 하나요? 새로운 상황이 발생했습니다.
- 자신이 할 수 있는 방법들을 찾아봅니다. 무엇이 있을까요?

가지고 있던 인형을 던진다, 먹을 것을 찾아서 기어간다, 울거나 소리를 지른다.

- 위에서 제시한 3가지 방법들의 공통점은 무엇일까요?
맞습니다. 바로 생명의 근원인 엄마에게 신호를 보내는 것입니다.
- 아기가 신호를 보내면 주방에서 일하던 엄마는 그 신호를 듣습니다.
- 본능적으로 엄마는 어떻게 할까요? 바로 아기에게 달려오게 됩니다.



이제 엄마의 입장을 살펴봅시다.

[엄마의 입장]

- 엄마는 주방에서 일을 하고 있습니다.
- 거실에서 놀던 아기가 갑자기 ‘으앙’ 울음을 터트립니다.
- 앞의 이야기처럼 울음소리를 듣고 엄마가 하는 행동을 예상해 보면 다음과 같습니다.

- 아기의 얼굴을 살펴봅니다.
- 아기 울음소리의 크기도 들어봅니다.
- 아기의 배도 만져봅니다.
- 아기의 손과 발의 모습들을 살펴봅니다.
- 거실에 걸려있는 시계도 살펴봅니다.
- 아기 주위에 있는 장난감도 꼼꼼히 살펴봅니다.

- 엄마는 아기가 보내오는 신호들, 즉 정보를 가지고 머릿속에서 어떤 판단을 내릴까요?
또한 판단을 내리는데 필요한 추가 정보는 무엇일까요?
- 아기가 보내는 신호 외에 추가적인 정보도 있습니다. 바로 시계입니다.
- 엄마는 시계를 보고 점심이 되었음을 알게됩니다.

그렇습니다. 엄마가 판단하기를 아기의 울음소리와 기타의 정보를 분석해 볼 때 바로 ‘배고픔’에서 오는 신호라고 최종적으로 판단을 내리는 것입니다.

물론, 현실에서 주변의 신호를 받아 판단을 내리는 행위들은 늘 일어나는 일이지만 이렇게 문제를 해결해 나가는 과정을 다음과 같이 순서대로 요약하여 적어보았습니다.

SUMMARY

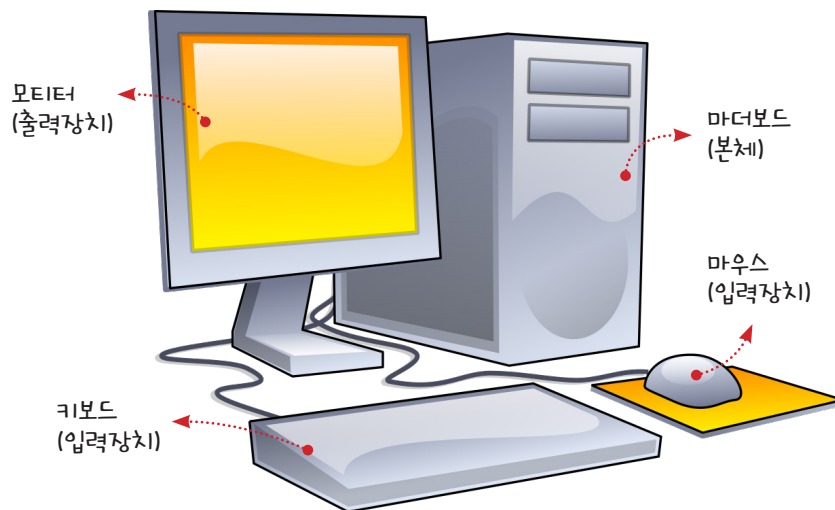
- ① 아기는 엄마에게 신호를 보냅니다.
- ② 엄마는 아기가 보내는 신호를 받아서 정보를 수집합니다.
- ③ 방금 받은 정보이외에 기존에 있던 정보들도 같이 분석합니다.
- ④ 엄마는 아기의 ‘배고픔’을 알게 됩니다. 즉 아기와 엄마가 ‘연결’되는 것이지요.
- ⑤ 엄마는 아기에게 ‘필요한 음식을 가져다주는 행동’을 합니다.
- ⑥ 아기의 배고픔은 사라집니다.

피지컬 컴퓨팅(Physical Computing)의 원리

피지컬 컴퓨팅의 원리도 이와 같이 아기와 엄마의 관계로 비유할 수 있습니다. 바로 아기는 센서(Sensor)에 해당되고 엄마는 센서를 받아들이는 마더보드(MotherBoard)라고 할 수 있습니다.

센서는 다양한 일을 할 수 있지만 간략하게 표현하면 어떤 정보를 받아들이거나 제공하는 역할이라고 볼 수 있습니다. 마더보드는 센서가 제공하는 정보를 받아 문제해결에 필요한 처리과정을 거쳐서 결국 다양한 형태와 방식으로 다시 제공하는 것, 즉 출력하는 일을 합니다.

센서와 마더보드 둘다 중요하지만 피지컬 컴퓨팅에서 무엇보다도 중요한 것은 바로 '연결'입니다. 사실 아기는 아직 스스로 생각하고 행동할 수 있는 능력이 부족해서 엄마의 도움을 필요로 합니다. 마찬가지로 센서도 자신이 가지고 있는 기능, 즉 센서가 받아들이는 정보를 충분히 제공하기 위해서는 마더보드와 연결되어야 합니다. '연결'의 의미에 대해 좀 더 알아 봅시다. 컴퓨터와 연결되어 있는 일반적인 장치들을 살펴보면 다음과 같습니다.

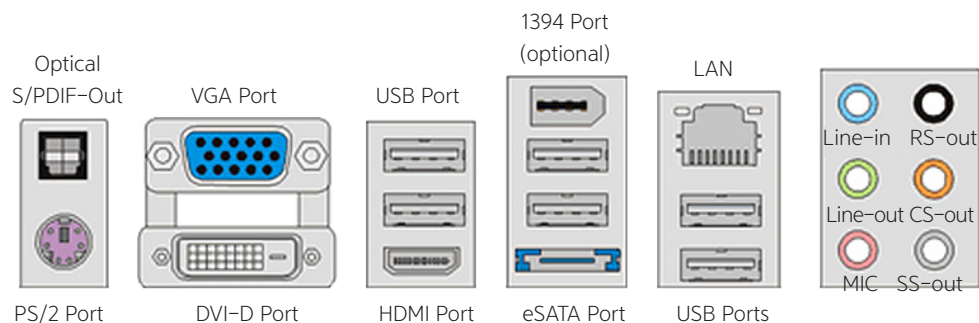




우리가 일반적으로 컴퓨터의 본체라고 말하는 마더보드(연결되어 정보를 주고받을 수 있는 장치)는 크게 입력장치와 출력장치로 구분할 수 있습니다. 그림에 표시된 입력장치는 마우스, 키보드가 있고 출력장치는 모니터입니다.

이러한 입출력장치는 마더보드와 어떻게 연결될까요? I/O 인터페이스라고 부르는 입출력(Input/Output) 인터페이스(Interface)를 통해서 연결됩니다.

즉 각각의 입출력장치를 마더보드와 연결하여 정보를 주고받기 위해서는 일정한 규약 또는 규칙이라 부르는 프로토콜(Protocol)이 필요합니다. 다시말해 입출력장치와 마더보드는 프로토콜이 정의한 규격에 의해 디자인된 모양이나 형태로 ‘연결’되어야 합니다.



[그림 1-1] 마더보드 Super I/O 인터페이스

(그림출처 <http://www.pcbuyer beware.co.uk/hardware/motherboards-cases-psus/4/>)

[그림1-1]은 컴퓨터의 마더보드에 연결하는 입출력 포트(Input/Output Ports)로 다양한 인터페이스 규격에 맞게 디자인 되어있습니다. 일반적으로 USB 포트에는 키보드와 유선 마우스를 연결하고 VGA 포트, DVI-D 포트 또는 HDMI 포트에는 모니터를 연결합니다.

컴퓨팅의 원리

이 책에서 이야기하는 '컴퓨팅'은 컴퓨터가 정보를 처리하는 과정을 의미합니다. 다시 말해서 '입력-처리-출력'의 과정이며 이것을 '컴퓨팅(Computing)'이라고 정의할 수 있습니다.



[그림 1-2] 마더보드 Super I/O 인터페이스

[그림1-2]와 같이 컴퓨터와 I/O 인터페이스를 통해 연결되어 있는 입력장치에서 정보를 입력받고 입력받은 정보는 마더보드의 중앙처리장치인 CPU에서 다양한 연산과정(계산과정)을 거쳐서 다시 I/O 인터페이스를 통해 연결된 출력장치로 정보를 전송하여 최종적으로 정보를 출력하는 과정이라고 얘기할 수 있습니다.

일반적으로 컴퓨터는 각종 계산을 처리하는 연산장치인 CPU와 정보를 저장하는 메모리, 입출력장치들로 구성되어 있다고 하였습니다. 현실속에 이러한 컴퓨터를 어디에서 찾아 볼 수 있을까요?

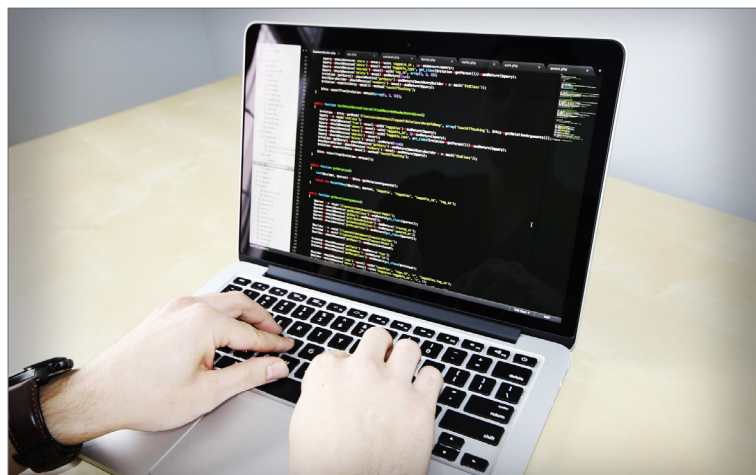


[그림 1-3] 다양한 스마트 기기들



[그림 1-3]과 같이 생활 속 다양한 스마트기기들 중에도 각종 정보를 입력받아 그 정보를 처리하고 제어할 수 있는 컴퓨팅 기기들이 있습니다. 대표적인 기기에 스마트폰이 있습니다. 여기서 한 가지 궁금한 점이 있습니다. 이러한 장치들을 제어하고 정보를 처리하는 일은 어디에서 할까요?

정답은 바로 소프트웨어입니다. 컴퓨팅의 원리에서 입력, 처리, 출력의 과정 중 다양하게 연결된 기기들을 제어하는 ‘생명의 근원’이 바로 ‘소프트웨어’라고 할 수 있습니다.



[그림 1-4] 소프트웨어

그렇다면 다양한 장치들, 즉 하드웨어를 움직일 수 있는 생명의 근원인 소프트웨어에 대해서 좀 더 살펴보겠습니다.



문제를 분해해봅시다!

우리가 알고 있는 소프트웨어는 어떤 것들이 있을까요?

쉽게 설명하면 컴퓨터 전반에 연결된 하드웨어를 제어하고 다양한 프로그램 실행의 기반이 되는 운영체제인 Windows10, iOS, OSX등이 있고, 프로그램인 MS OFFICE, 한글, 엑셀, 웹브라우저 등이 있습니다. 사실 어플리케이션(Application)이라고 불리는 프로그램은 소프트웨어의 한 분류에 속한다고 볼 수 있습니다.

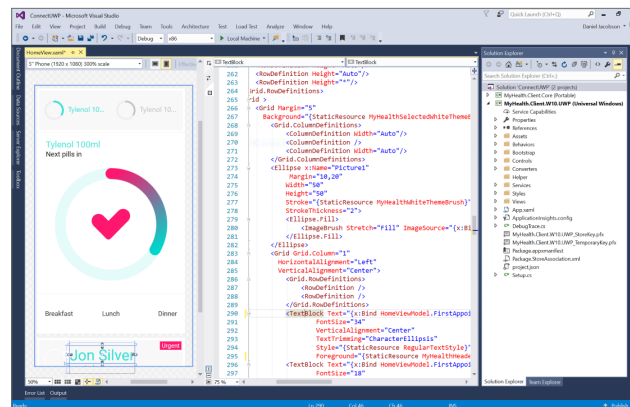
다양한 프로그램을 만들기 위해서는 컴퓨터가 이해하는 언어를 사용해야 합니다. 그것을 우리는 일반적으로 프로그래밍 언어(Programming Language)라고 합니다.

```
#include <iostream>
using namespace std;

int main()
{
    cout << "Hello, World!";
    return 0;
}
```

[그림 1-5] C++ 언어로 나타낸 ‘Hello, World!’ 출력 프로그램

프로그래밍 언어도 실제 생활에서 사용하는 언어처럼 ‘언어의 형태’를 가지고 있습니다. 즉, 그 언어가 가진 문법에 맞게 프로그램을 작성해야 컴퓨터가 이해하고 실행할 수 있습니다. [그림 1-5]와 같이 텍스트 기반의 언어인 C++은 읽기에도 복잡하고 언어의 사용문법도 어려워 처음 컴퓨터 프로그램을 접하는 사람들이 이해하기에는 힘든 점이 많습니다.



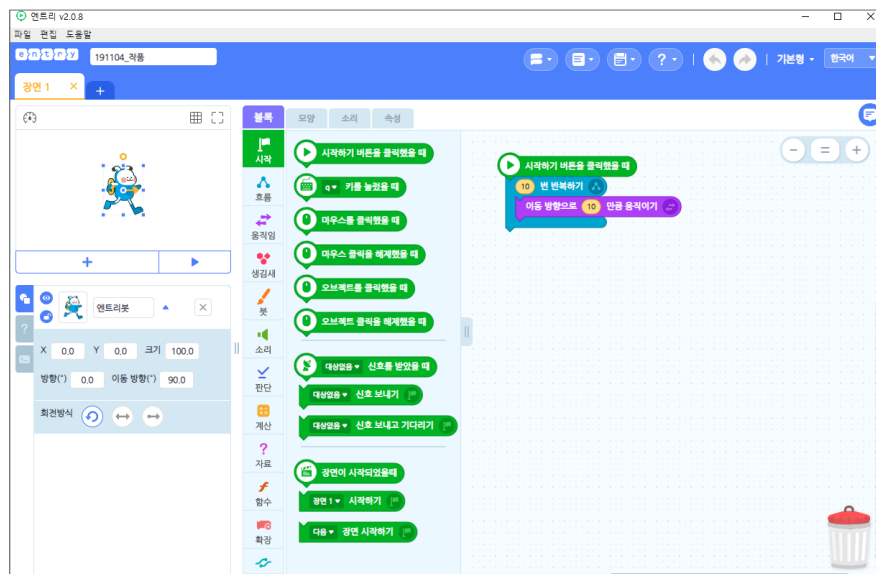
[그림 1-6] Visual Studio 2017 프로그램(출처: www.visualstudio.com)



교육용 프로그래밍 언어

텍스트 기반의 프로그램을 처음 접하는 사용자들은 컴퓨터가 정보를 수집하고 처리하는 방식을 이해하는데 상당한 시간과 노력이 필요하여 학습이 진행될수록 흥미를 잃어고 포기하기가 쉽습니다.

반면 그래픽 기반의 프로그래밍 언어는 프로그램에 필요한 요소들을 끌어다 놓아 디자인하고 프로그램 작성을 위한 편집기도 디자인된 요소들을 서로 연결하여 활용할 수 있도록 제공하여 텍스트 기반의 프로그램보다 훨씬 쉽게 프로그램을 작성할 수 있습니다.



[그림1-7] 블록형 프로그래밍 언어 엔트리

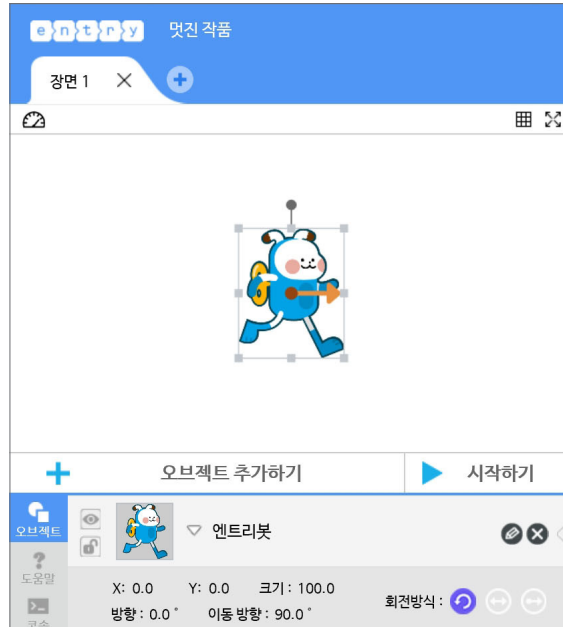
[그림1-7]과 같이 블록형 프로그래밍 언어 엔트리는 간단한 블록들을 오른쪽의 블록조립소에 끌어 놓으면 실제 프로그램이 작성되는 원리입니다. 책에서 제시하고 있는 다양한 예제들을 학습하기 위해서는 엔트리의 기본적인 사용법 즉 블록을 끌어다 놓고 스크립트를 작성하고 실행하고 자신이 작성한 프로그램을 저장하고 불러오는 등의 기본적인 사용법에 대해서는 먼저 학습하기를 바랍니다.



하나씩 연결해봅시다!

오브젝트 다루기

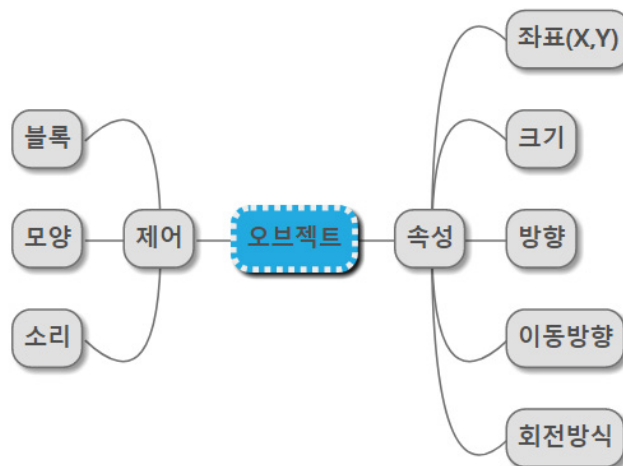
엔트리에 접속해서 기본적인 환경에 대해 알아보시다. 먼저 눈에 띄는 것이 바로 [그림1-8]에 있는 오브젝트(Object)입니다.



[그림1-8] 블록형 프로그래밍 언어 엔트리

오브젝트들이 움직이는 공간을 배경이라고 합니다. 배경도 하나의 오브젝트에 속합니다. 오브젝트는 다음과 같이 여러 가지 속성을 가지고 있습니다.

[그림 1-9] 블록형 프로그래밍 언어 엔트리(출처: playentry.org)



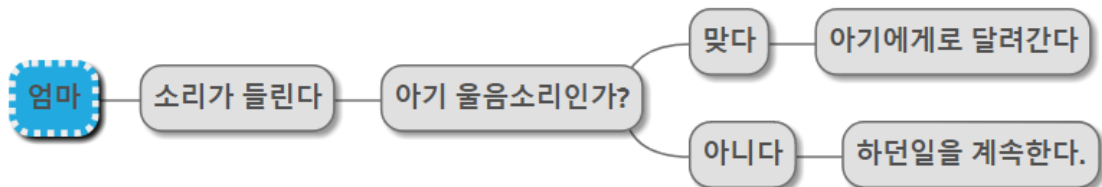
오브젝트의 속성은 오브젝트 자체의 좌표, 크기, 방향, 이동방향, 회전방식을 포함합니다. 이 속성값은 사용자가 지정할 수 있습니다. 오브젝트를 제어하기 위해서는 블록, 모양, 소리 탭에 있는 요소들을 에디터에 끌어다 놓고 원하는 대로 수정하여 프로그래밍해야 합니다.



이벤트 다루기

이전 내용에서 아기의 울음소리를 기억하시나요?

엄마의 입장에서 생각할 때 ‘아기의 울음소리가 들리면 아기에게로 달려간다.’는 것을 프로그래밍 한다고 가정하면 다음과 같은 것들을 생각해 볼 수 있습니다.



[그림 1-10] 문제해결 절차 생각하기

엄마는 주방에서 일을 하고 있습니다. 만약 소리가 들린다면 다음과 같은 조건들을 생각해 볼 수 있습니다.

- 아기의 울음소리인가?
 - 맞다: 아기에게로 달려간다.
 - 아니다: 하던 일을 계속한다.

여기에서 아기의 울음소리를 확인하는 것 이것을 ‘이벤트’라고 합니다.

컴퓨터와 관련시켜 생각해 보면 컴퓨팅(Computing) 과정에서는 다양한 입출력장치를 통해 정보를 수집하거나 제공하고 연산장치(CPU)를 통해서 정보를 처리합니다.

이벤트는 이러한 정보처리 과정에서 다양한 입출력장치나 연산장치에 정보가 입력되거나 출력되었을 때를 의미합니다.

예를 들어 이벤트는 ‘키보드의 화살표 왼쪽 키를 눌렀을 때’, ‘1부터 100까지 숫자를 다 더했을 때’와 같이 다양한 조건들이나 상황이 발생하는 것을 의미합니다.

사실 [그림 1-10]에서 표현하고 있는 일련의 절차들은 아기가 울었을 때 해결할 수 있는 방안을 의미하는 것이며 우리는 이러한 절차를 ‘알고리즘(Algorithm)’이라고 부릅니다.

스크립트로 표현하기

그렇다면 어떻게 이러한 알고리즘을 프로그래밍으로 만들 수 있을까요?

간단한 엔트리 프로그램을 활용하여 알아보시다.

엔트리 프로그래밍의 핵심은 해결하고자하는 문제를 알고리즘으로 만들고 그 절차대로 프로그래밍하는 것입니다.

다시 말하면 문제해결에서 제시된 이벤트를 오브젝트와 연결하는 것입니다.

이제 실제로 프로그래밍을 해보겠습니다.

우선, 오브젝트 이전에 있는 엔트리봇

오브젝트는 사용하지 않으므로

삭제 버튼을 클릭하여 오브젝트를 삭제합니다.



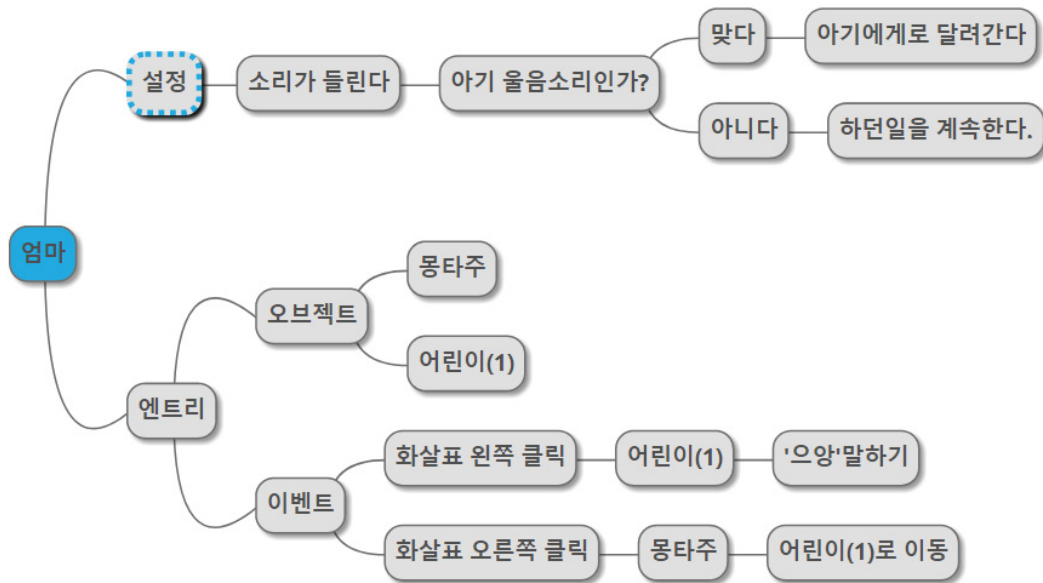
‘오브젝트 추가하기 ’ 클릭

→ '사람' 클릭

→ '몽타주', '어린이(1)' 오브젝트를
선택하여 추가합니다.



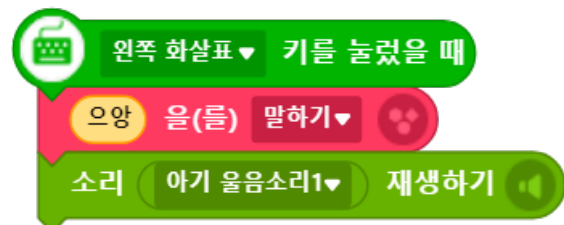
이제 만들어진 알고리즘을 바탕으로 오브젝트와 이벤트에 대해서 생각해 봅시다.



우선 프로그램에서 필요한 오브젝트는 몽타주, 어린이(1)입니다.

필요한 이벤트는 크게 화살표 왼쪽, 오른쪽 키를 클릭했을 경우 발생합니다. 왼쪽 화살표를 클릭하면 어린이(1) 오브젝트에서 '오양'말하기를 실행하고 오른쪽 화살표를 클릭하면 몽타주 오브젝트가 어린이(1) 오브젝트로 이동하게 만듭니다.

따라서 어린이(1) 오브젝트의 블록 프로그래밍은 오른쪽 그림과 같이 왼쪽 화살표 키를 눌렀을 때 '오양' 말하고, '아기 울음소리'를 재생하도록 만듭니다.



아기(어린이(1) 오브젝트)의 울음소리가 들리면 오른쪽 화살표를 클릭합니다. 그러면 몽타주 오브젝트가 아기쪽으로 이동합니다.



여기서 한 가지 궁금한 점이 있지 않으신가요?

실제 생각한 알고리즘은 바로 아기가 울면 엄마는 자동으로 움직이게 하라는 것이었습니다.

그렇다면 여러분은 어떻게 프로그래밍 할 것인가요?

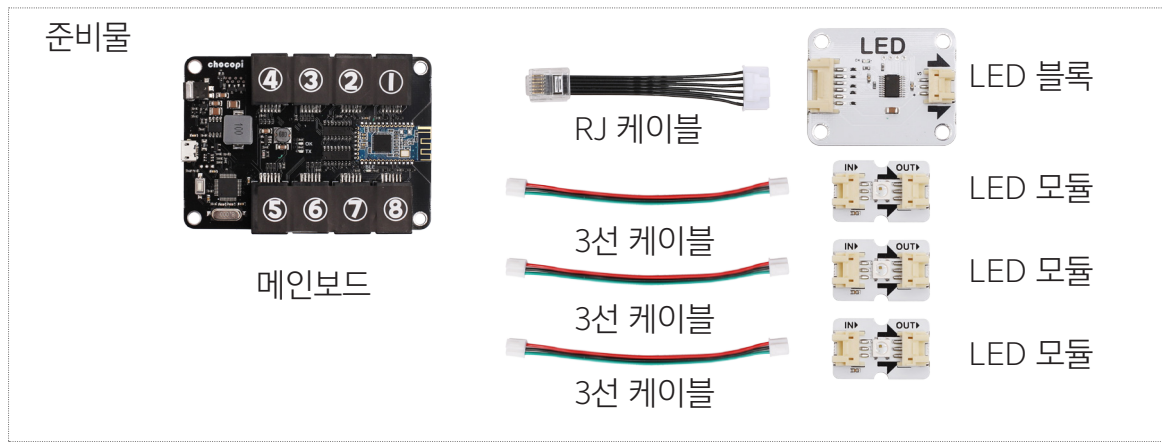


미션을 해결해봅시다!

이전 섹션에서 다루었던 내용을 기초로 하여 자신만의 아이디어를 추가해서 프로그램을 더욱 발전시킬 수 있는 섹션입니다.

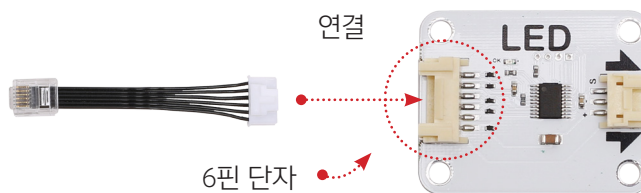
엔트리와 초코파이보드의 연결

초코파이보드를 제어하기 위해 다음의 과정을 한 번 따라해 보세요.

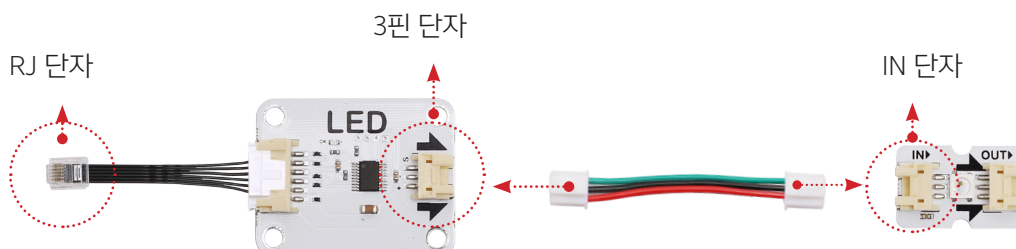


초코파이보드, LED 블록, LED 모듈 및 연결 케이블을 준비합니다. LED 센서와 연결되는 커넥터는 3가닥의 줄로 구성된 3선 케이블이고, LED 센서블록과 초코파이보드 연결은 까만색의 RJ케이블로 연결합니다.

① LED 블록의 6핀 단자에 RJ 케이블을 연결합니다.

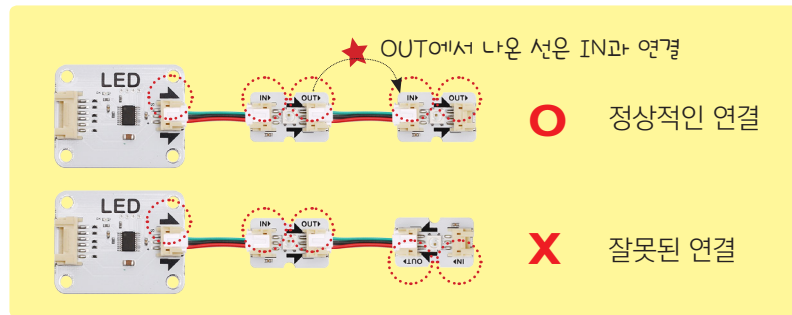
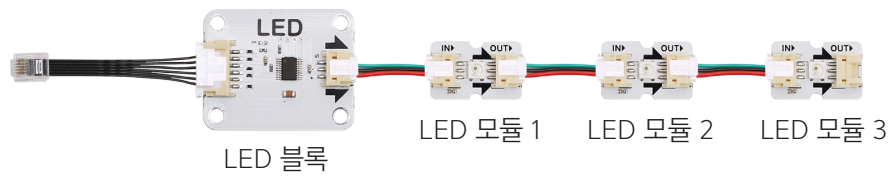


② 3선 케이블을 이용하여 LED 블록과 LED 모듈의 IN 단자를 연결합니다.



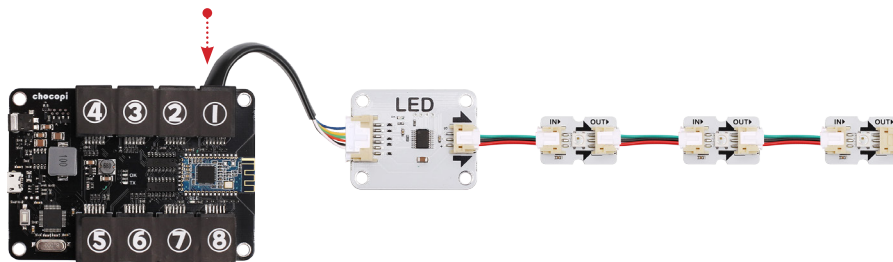


LED 모듈을 두 개 더 연결합니다. LED 블록, LED 모듈에는 IN, OUT 방향이 있습니다. 주의하여 연결해야 합니다.



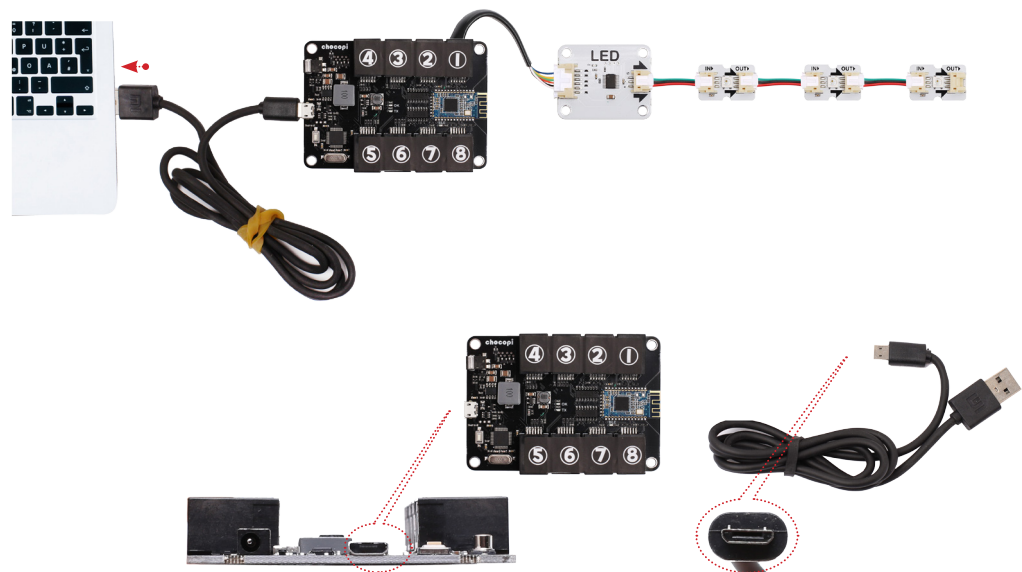
③ LED 블록을 초코파이 보드에 연결합니다.

초코파이보드에는 1번부터 8번까지의 포트가 있습니다. 1번 단자에 연결합니다.



④ 초코파이보드를 컴퓨터와 연결합니다.

초코파이보드를 컴퓨터와 연결하기 위해 USB 케이블을 사용하여 컴퓨터의 USB 포트에 연결합니다.



엔트리 프로그램 실행

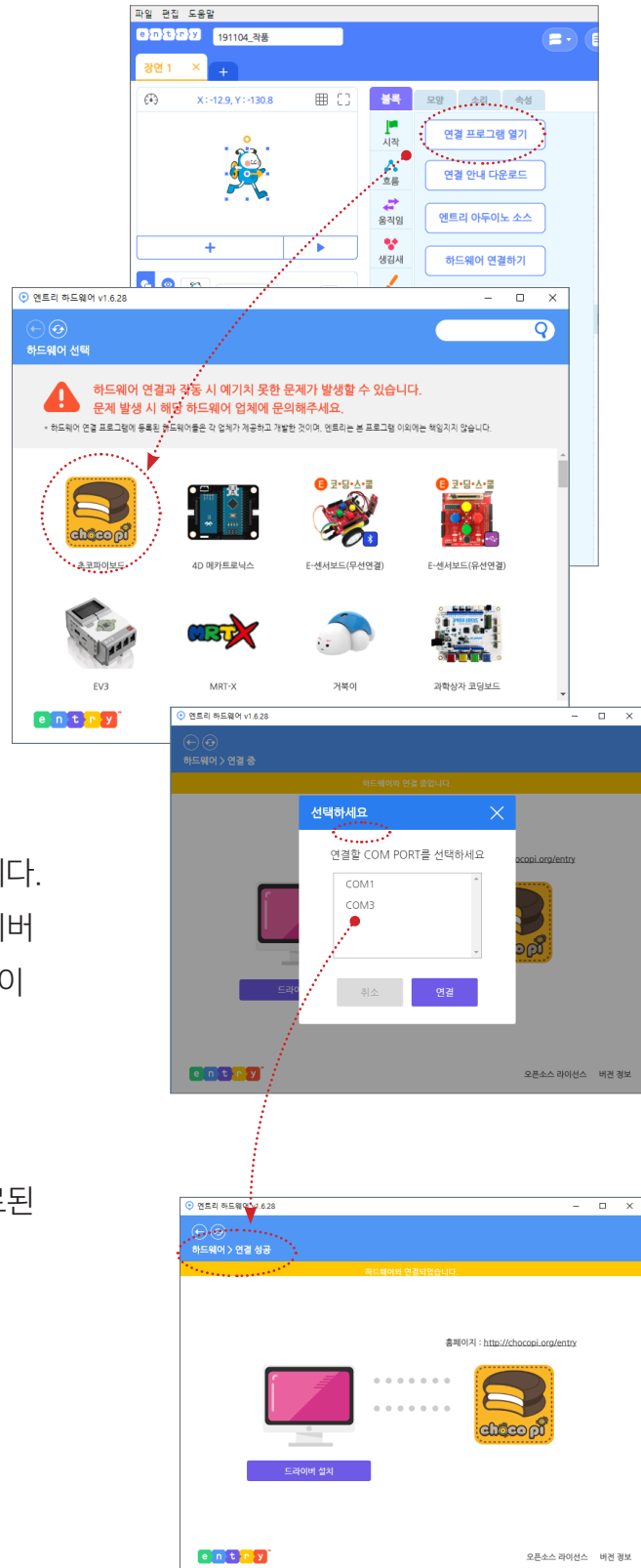
엔트리 프로그램에서 블록탭 맨 아래에 있는 하드웨어를 클릭하고 연결 프로그램 다운로드 버튼을 클릭하여 엔트리 하드웨어 프로그램을 설치합니다.

연결 프로그램 열기를 클릭합니다.

초코파이 보드를 선택합니다.

초코파이 보드와 엔트리 프로그램의 통신 역할을 하는 COM 포트를 선택합니다.
(※ Windows 8 이하 버전인 경우 드라이버 설치를 해야 리스트에 COM 포트 목록이 나타나고 선택할 수 있습니다.)

'연결 성공' 메시지가 보이면 연결이 완료된 상태입니다.



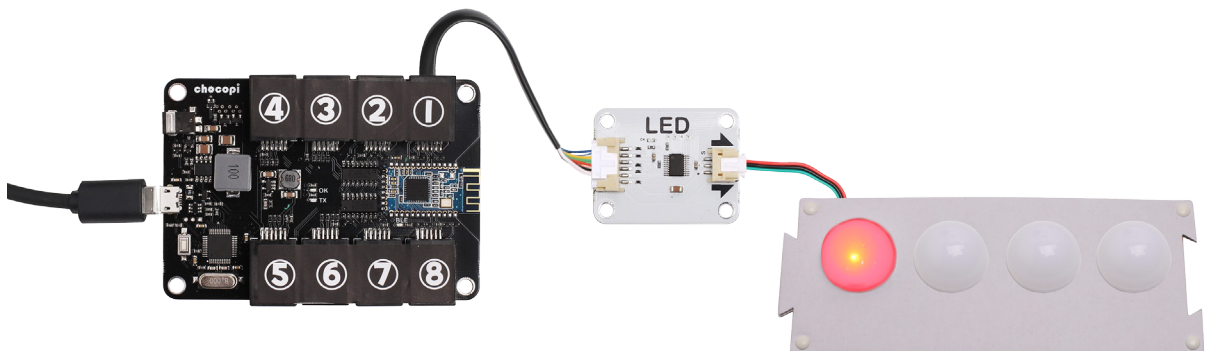


프로그래밍 하기

어린이(1) 오브젝트를 선택하고 하드웨어 블록에서 LED 블록을 가져다가 아래와 같이 프로그래밍합니다.



왼쪽 화살표 키를 눌렀을 때 초코파이보드의 1번 포트에 연결된 LED 블록의 첫 번째 LED 모듈이 빨간색불이 들어옵니다. LED가 밝아 눈이 부실 수 있습니다. 시력을 보호 할 수 있도록 가까이 오래 보지않도록 늘 유의합니다.



MISSION 01 우주선을 띄워라

지금은 서기 3030년입니다. 지구라는 별도 있지만 이 곳 지구는 수많은 은하계 속에 속한 하나의 행성 일 뿐이라는 것을 알게 되었습니다.

따라서 어디에 있는지 모르는 다양한 미지의 속으로 탐험하기 위해서 우리가 만든 우주선을 타고 다른 행성으로 여행을 가려고 합니다.



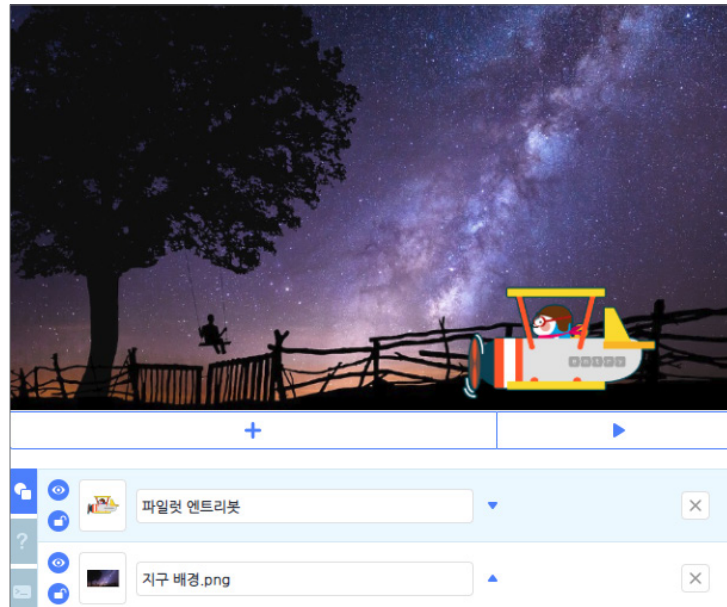
① 우주선 이벤트 디자인하기

- 스페이스 키를 누르면 우주선에 연결된 LED 3개에 차례대로 빨강, 노랑, 초록 불이 들어오게 해 봅시다.
- 우주선이 움직이고 있다는 것을 보여주기 위해서 우주선을 위아래로 흔들리게 해 봅시다.



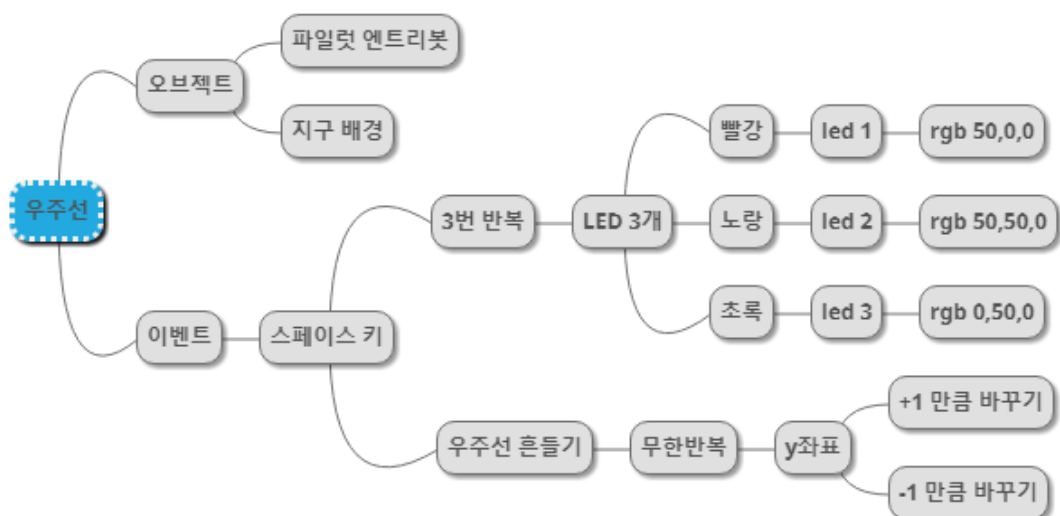
② 오브젝트 추가하기

파일럿 엔트리봇, 지구배경 오브젝트가 포함되어 있는 소스파일을 엔트리로 불러옵니다.
(엔트리 파일은 초코파이 보드 홈페이지에서 다운받을 수 있습니다.)

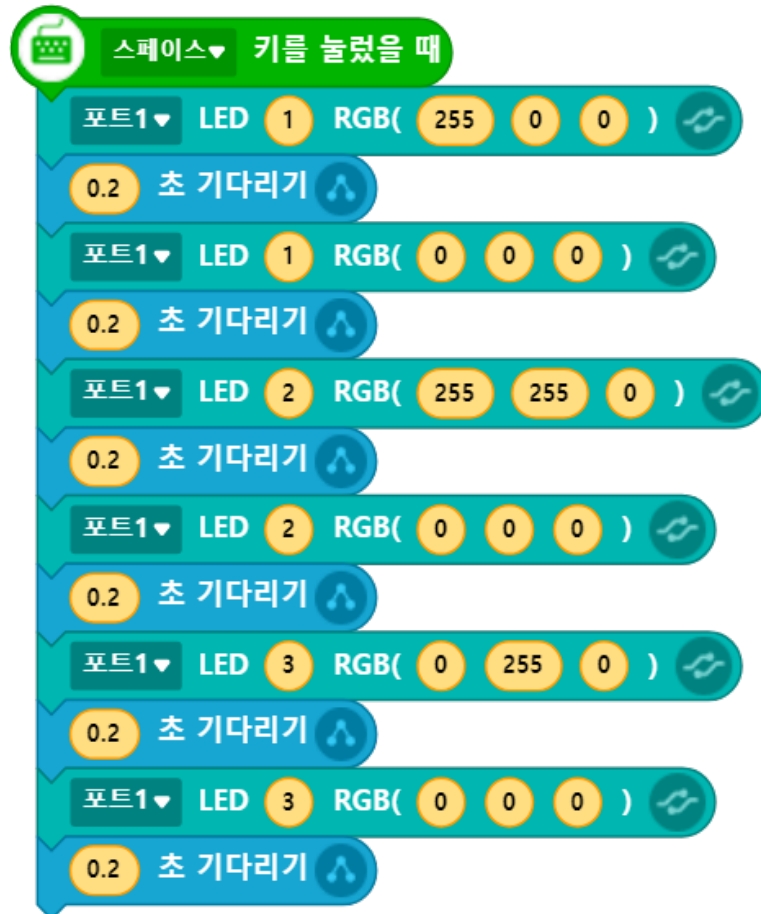


③ 오브젝트와 이벤트 연결하기

제시된 문제를 해결하기 위한 알고리즘을 간략하게 정리하면 다음과 같습니다. LED가 깜빡인 후 우주선 흔들기를 합니다.

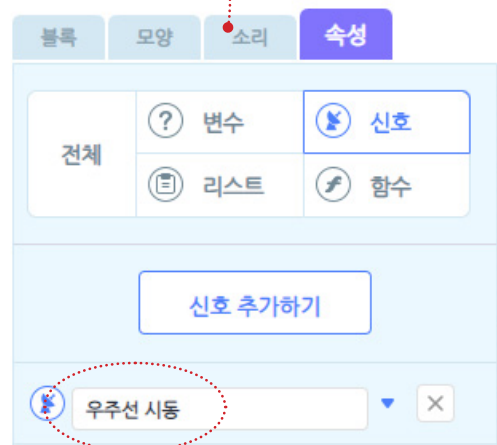
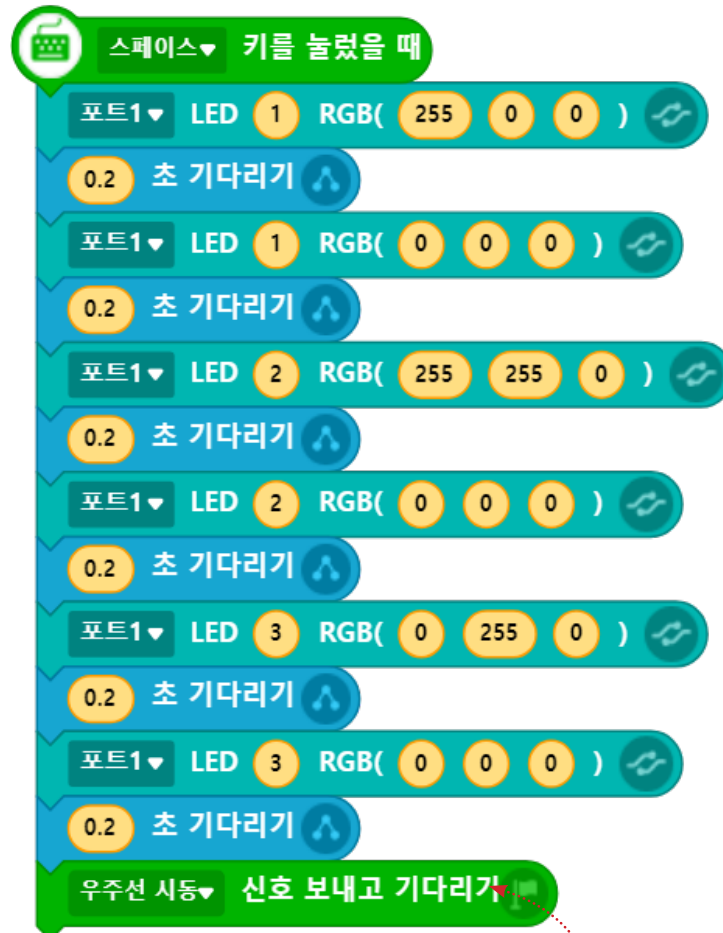


스페이스 키를 눌렀을 때 LED 세 개가 차례대로 빨강, 노랑, 초록 불이 들어오도록 합니다.
불빛이 깜빡이게 하기 위해서 다음과 같이 블록 프로그래밍 합니다.
'지구 배경'오브젝트를 선택하고 프로그래밍 합니다.

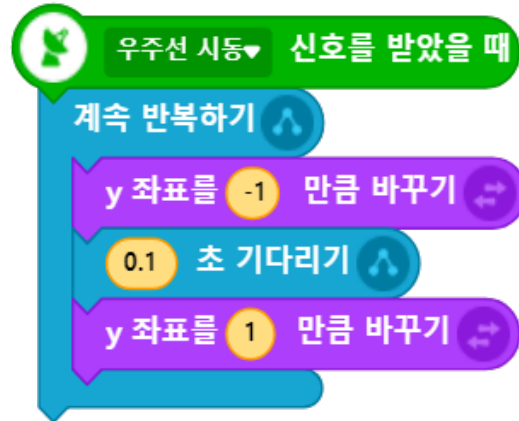




LED 시동이 끝나고 나면 ‘파일럿 엔트리봇’ 오브젝트에 ‘우주선 시동’이라는 신호를 전달할 수 있도록 하여 우주선에 시동이 걸리게 표현합니다. 그러기 위해서는 ‘우주선 시동’ 신호를 추가하고 LED 블록 아래에 다음의 블록을 추가합니다.



우주선에 시동이 걸리면 자동으로 위아래로 조금씩 움직이게 해야합니다.
그러기 위해서는 y좌표를 위아래로 이동하는 방법을 적용하여 다음과 같이 블록 프로그래밍 합니다. '파일럿 엔트리봇' 오브젝트를 선택하고 프로그래밍 합니다.



④ 프로그램 테스트와 디버깅

시작 버튼을 클릭하여 프로그램을 실행하여 봅시다.



스페이스 키를 누르면 LED에 불빛이 차례대로 들어오고 우주선에 시동이 걸리는 것을 볼 수 있습니다.

여기에서 3번 LED의 색이 아직도 빨간색으로 나타난 것을 볼 수 있습니다.

구글에서 'RGB 10진수 255 색상표'로 검색해 보면 표시하고 싶은 색상과 관련된 코드정보를 찾아 볼 수 있습니다.



⑤ 프로그램 개선하기

LED 빛을 순서대로 깜빡이기 위해 만든 블록 프로그래밍을 살펴봅시다.

계속해서 반복되고 있는 것들을 찾아보고 더 간단하게 만들 수 있는 방법은 없을까 생각해봅시다.

아래의 블록들을 활용해서 프로그램을 개선해 봅시다.



MISSION 02 나만의 우주선

이제 진짜 우주선을 띄우기 위해 필요한 것들을 생각해 봅시다.

- 우주선을 움직이기 위해서 필요한 것들
 - 화살표 위, 아래 키
 - 오브젝트의 y좌표 이동

화살표 위, 아래 키를 누르면 우주선이 위, 아래로 움직일 수 있도록 프로그램을 만들어 봅시다.



02

태양계 탐험장비

오브젝트, LED 다루기





문제가 무엇인가요?

이전 장에서 여러분은 우주선을 위, 아래로 움직이도록 프로그래밍해 보았습니다. 이러한 프로그래밍의 제작과정을 다시 요약해 보면 다음과 같습니다.

1. 해결하고자 하는 문제를 분석합니다.
2. 문제에서 제시된 정보를 가지고 필요한 자료를 수집합니다.
3. 수집된 자료에서 일정한 패턴을 찾고 분류합니다.
4. 해결하기 위한 방안을 순서대로 정리하여 하나의 알고리즘으로 만듭니다.
5. 만들어진 알고리즘에서 제시한 절차대로 필요한 이벤트를 구성합니다.
6. 구성된 이벤트에 따라서 필요한 오브젝트를 분류하고 정리합니다.
 - 엔트리 자체의 오브젝트를 수집하고 찾아봅니다.
 - 초코파이보드에 연결하는 센서를 찾아서 구성하고 조립합니다.
7. 각각의 오브젝트에 이벤트별로 연결된 알고리즘을 다시 연결합니다.
 - 오브젝트별로 관련된 이벤트를 처리하기 위해 필요한 정보나 요소를 생각합니다.
 - 이벤트별로 처리되는 내용들이 자신이 만든 알고리즘대로 구현되는지 확인합니다.
8. 알고리즘에 따라 오브젝트별로 이벤트를 블록 프로그래밍 하면서 필요한 내용을 추가 또는 삭제하여 프로그램을 구조화 시킵니다.
9. 프로그램을 실행합니다.
10. 개선할 사항이나 오류를 찾아봅니다.
11. 프로그램을 개선하여 최종 결과물로 만듭니다.



문제를 분해해봅시다!

그렇다면 해결하기 위한 방안은 곧 이벤트라고 할 수 있습니다.

파일럿 엔트리봇이 위로 올라갈수록 지구로부터 점점 멀어지니까 보다 현실적으로 만들기 위해서 점점 파일럿 엔트리봇의 크기가 작아지도록 하고 아래로 내려올수록 크기는 점점 커지게 해 봅시다.

이벤트 다루기

제시한 문장을 다시 한 번 분석해서 절차대로 나누어 보면 다음과 같습니다.



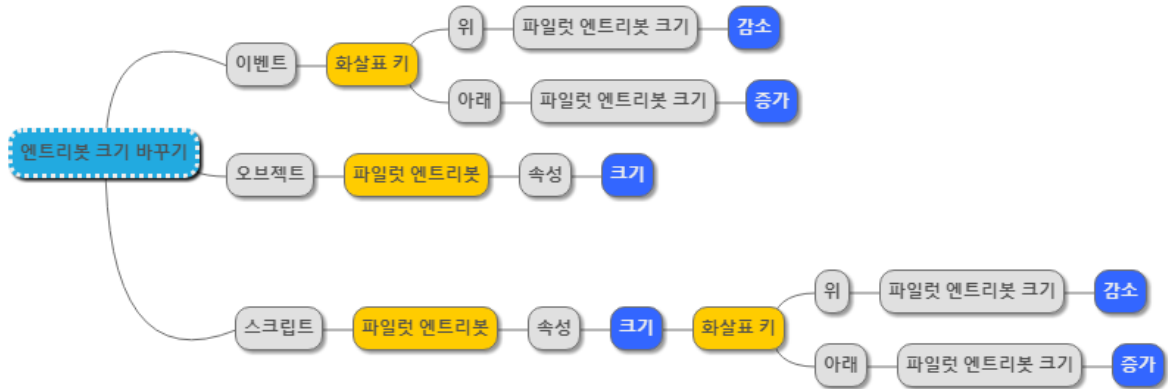
[그림 2-1] 문제해결 절차 생각하기

[그림 2-1]은 문제해결 과정을 시각적으로 표현했습니다. 이를 풀어서 설명하면 다음과 같습니다. 필요한 오브젝트는 '파일럿 엔트리봇'이고 이와 관련된 이벤트는 '화살표 키'입니다. 이해를 돕기 위해서 같은 색으로 표시하였습니다.

파일럿 엔트리봇 오브젝트의 속성을 살펴보면 크기를 조정하는 블록이 있습니다. 따라서 이벤트에서 제시된 위, 아래 방향키를 누르는 것과 오브젝트의 크기를 변경하는 속성을 연결하면 됩니다.

스크립트로 표현하기

여러분이 생각한 것을 다시 구조화한 후 스크립트로 표시해 봅시다.



[그림 2-2] 문제해결 절차 추상화(Abstraction)

[그림 2-2]와 같이 구조화된 이벤트와 오브젝트를 서로 연결한 것을 아래 ‘스크립트’로 표현할 수 있습니다.

오브젝트 블록 프로그래밍

그렇다면 실제 엔트리에서는 어떻게 구성할 수 있을까요? ‘파일럿 엔트리봇’ 오브젝트에 다음과 같이 추가할 수 있습니다.



다른 방법은 없을까요? 여러분이 생각한 더 좋은 방법이 있으면 적어봅시다.



하나씩 연결해봅시다!

오브젝트간 상호작용

제시된 문제를 해결 방안에 따라서 이벤트를 만들고 해당 오브젝트에 관련된 이벤트를 연결하여 스크립트로 작성하여 보았습니다. 그렇다면 다른 오브젝트와 상호작용하는 방법에 대해서도 알아보시다.

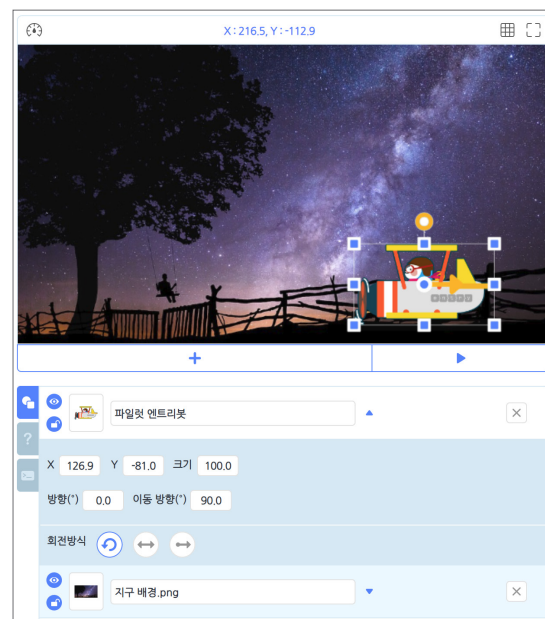
① 데이터의 표시

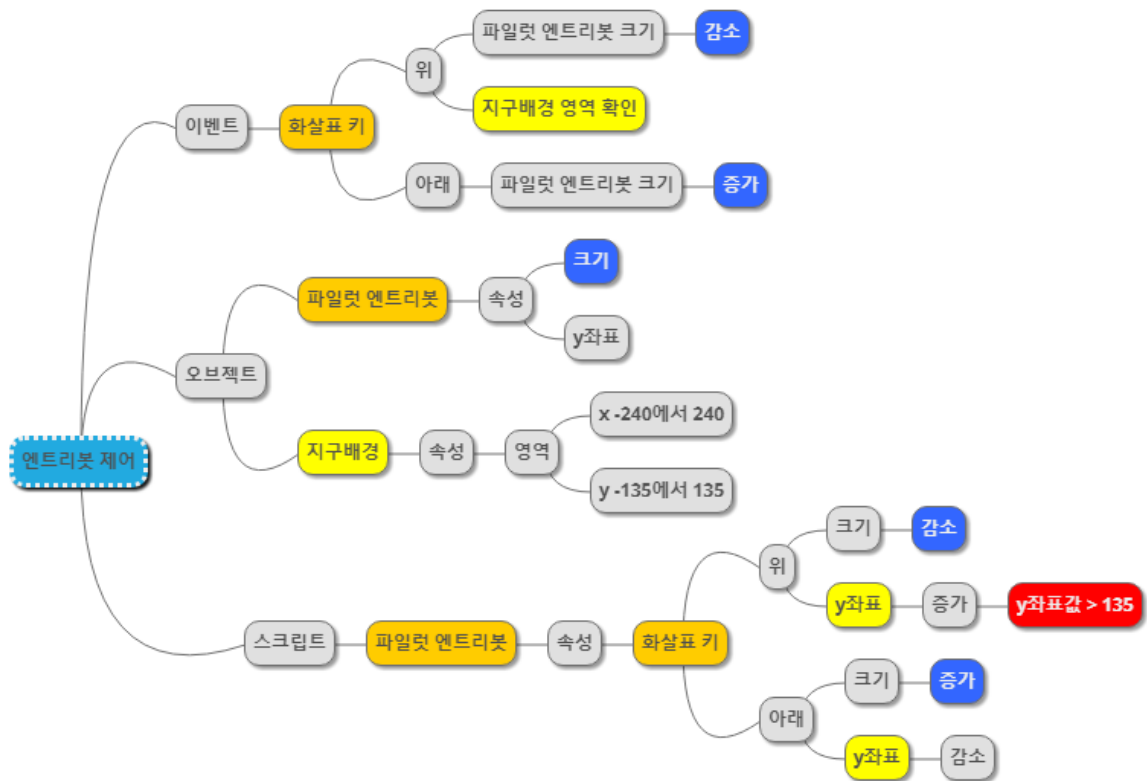
배경에 삽입된 오브젝트들을 상호작용 시키려면 오브젝트 각각의 속성에 따라 활용할 수 있는 데이터를 수집하고 분석하는 능력이 필요합니다. 간단한 예제를 통해서 알아보시다.

우선 ‘파일럿 엔트리봇’이 위로 올라갈 경우 크기가 점점 작아지게 하였습니다.
그런데 만약 파일럿 엔트리봇이 계속 올라가서 배경을 벗어나게 되면 어떻게 해야 할까요?
이러한 것은 어디서 확인해 볼 수 있을까요?

엔트리의 배경 크기는 480*270입니다.
x좌표의 영역은 -240에서 240까지,
y좌표의 영역은 -135에서 135까지입니다.

만약 ‘파일럿 엔트리봇’이 배경 영역을 벗어나면 어떻게 확인해야 할까요?
맞습니다. ‘파일럿 엔트리봇’의 y좌표값이 135보다 커지면 영역을 벗어난 것입니다.
다시 문제해결을 위해 순서에 따라서 알고리즘을 다음과 같이 요약해 보았습니다.





[그림 2-3] 해결방안 요약

[그림]에서와 같이 지구배경의 영역을 확인하기 위해서 관련된 오브젝트와 이벤트를 노란색으로 표기하였습니다. 알고리즘을 요약한 스크립트를 살펴보면 다음과 같은 절차를 가지고 있는 것을 알 수 있습니다.

화살표 키를 누른다

ㄱ. 위쪽 화살표를 누른 경우

- 오브젝트의 크기를 감소시킨다
- y좌표를 증가 시킨다
 - y좌표값이 135보다 크다면 영역을 벗어난 것으로 간주한다.
 - (y좌표값을 -80으로 정해서 아래로 위치하도록 한다)

ㄴ. 아래쪽 화살표를 누른 경우

- 오브젝트의 크기를 증가시킨다.
- y좌표를 감소 시킨다.



프로그램을 구현할 때는 이러한 오브젝트의 값을 확인할 필요가 있습니다. 따라서 다음과 같은 방법으로 y좌표의 값을 임시로 표시되도록 블록을 추가합니다.



‘우주선 시동’이라는 신호를 받았을 때

 블록을 통해서 임시로 y좌표값을 표시해서 프로그램을 만드는 입장에서 보다 쉽게 그 값을 확인할 수 있도록 합니다.

② 오브젝트 상호작용

이제, ‘파일럿 엔트리봇’이 y좌표값 135를 벗어나면 다시 원래의 위치로 돌아오게합니다.



③ 초코파이보드와 연결하여 생각하기

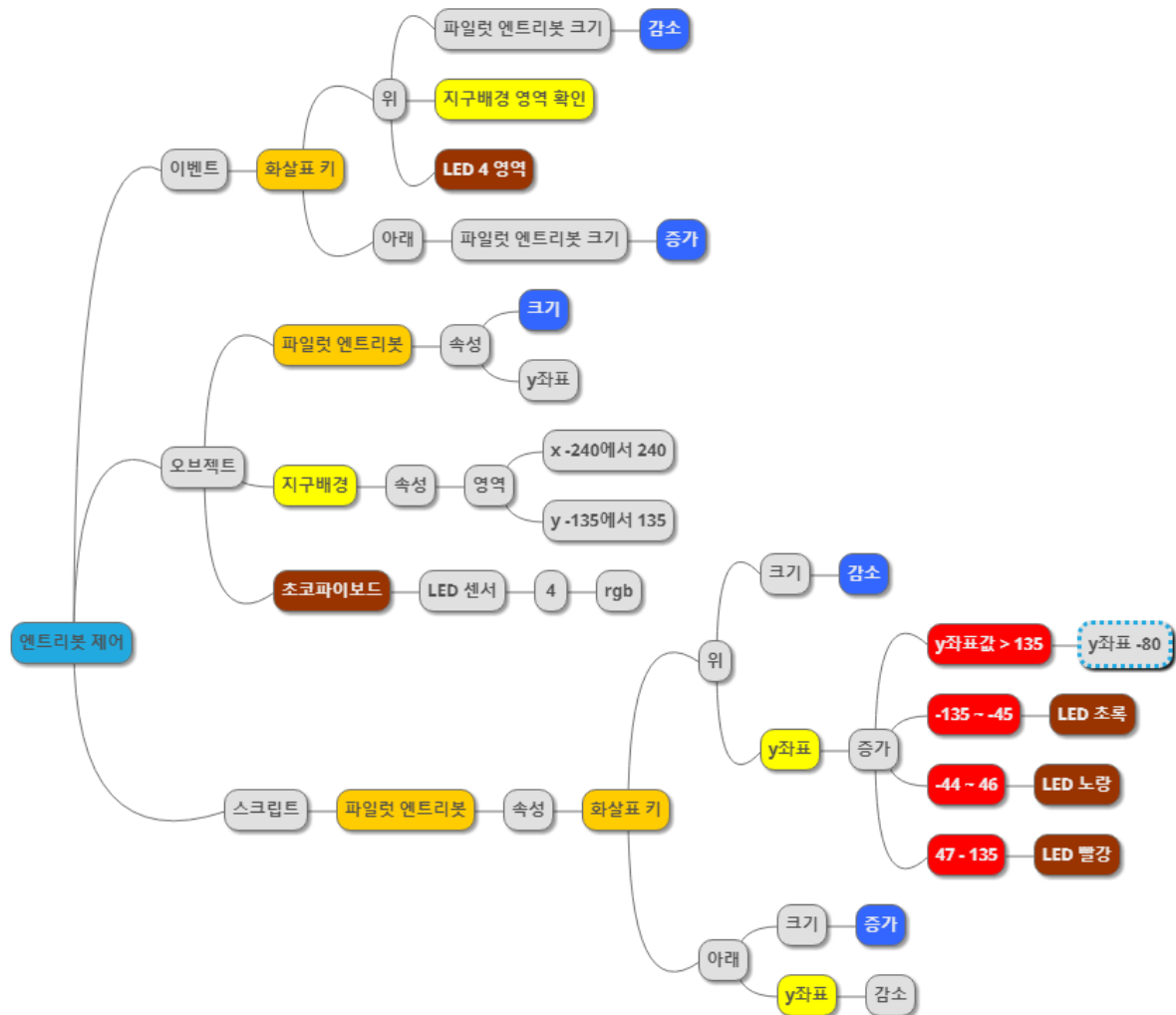
그렇다면 시동을 거는 LED 센서들과 연결하여 ‘파일럿 엔트리봇’의 y좌표 값의 범위에 따라서 4번째 LED의 색이 다음과 같이 바뀔 수 있도록 해 봅시다.

y 좌표 범위	LED 색상	RGB 코드값
-135 ~ -45	초록	0, 255, 0
-44 ~ 46	노랑	255, 255, 0
47 ~ 135	빨강	255, 0, 0

Color	Color HEX	Color RGB
	#000000	rgb(0,0,0)
	#FF0000	rgb(255,0,0)
	#00FF00	rgb(0,255,0)
	#0000FF	rgb(0,0,255)
	#FFFF00	rgb(255,255,0)
	#00FFFF	rgb(0,255,255)
	#FF00FF	rgb(255,0,255)
	#C0C0C0	rgb(192,192,192)
	#FFFFFF	rgb(255,255,255)

[표 2-1] 파일럿 엔트리봇 범위에 따른 LED 4번 색상

범위에 따른 LED 색깔을 다르게 표현하기 위해서 다음과 같이 정리해 보았습니다.



[그림 2-4] LED 범위에 따른 알고리즘 추가



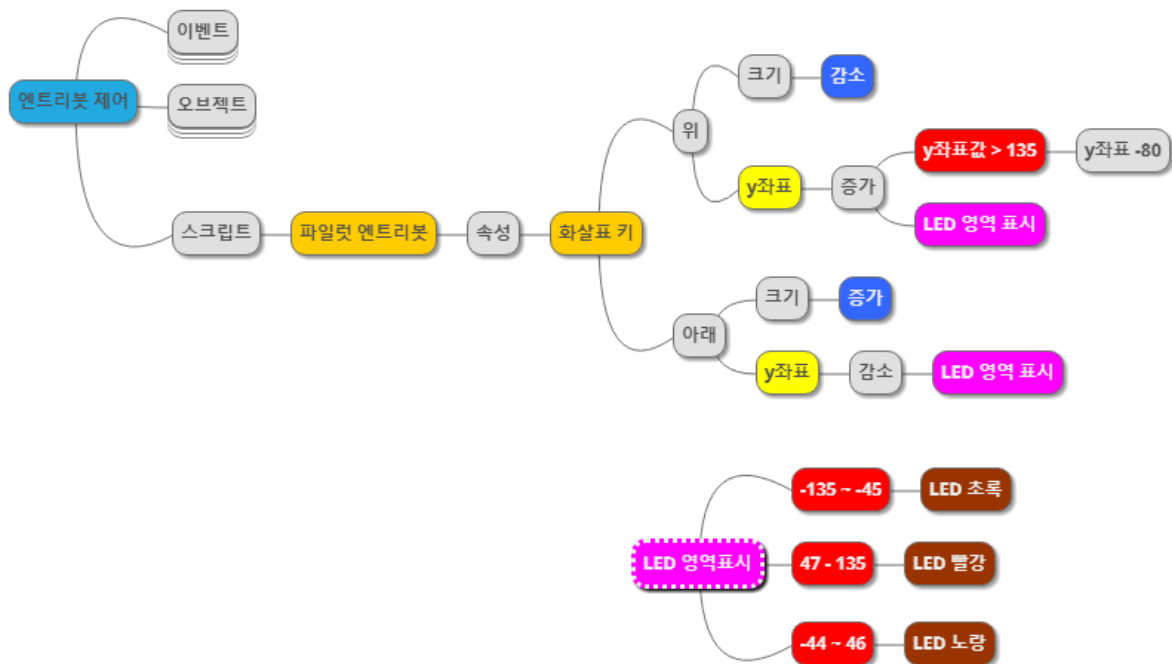
[그림 2-4]에서 갈색으로 표시된 부분은 초코파이보드와 관련된 내용입니다.

스크립트를 살펴보면 y좌표가 증가할 때 y좌표의 범위에 따라 LED 값을 제어해야 합니다.

그러나 여기에서 한가지 더 생각할 것이 있습니다. LED 색깔은 y좌표가 증가할 때 뿐만아니라 ‘감소’할 때도 적용이 된다는 것입니다.

즉, 화살표 키를 아래로 눌렀을 때도 위의 알고리즘이 같이 적용되어야 하는 것을 의미합니다. 이것을 어떻게 해결할 수 있을까요?

그렇습니다. 특정한 이벤트가 동시에 적용되기 위해서는 그 이벤트가 발생할 경우 미리 정의해 놓은 프로시저(Procedure)를 호출하여 같은 작업을 반복하여 수행하면 됩니다.



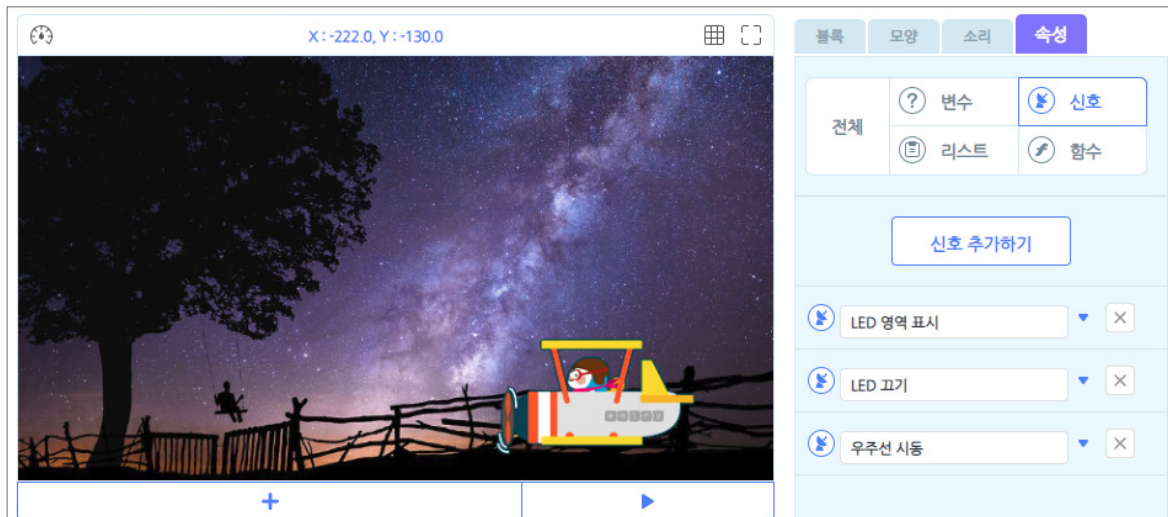
[그림 2-5] LED 영역 표시 프로시저 호출

[그림 2-5]와 같이 반복되는 작업을 호출할 수 있는 ‘LED 영역표시’를 만들어 y좌표 값이 변할때마다 해당 프로시저를 확인하는 알고리즘을 추가하였습니다.

실제 프로그래밍에서는 어떻게 적용되는지 알아보겠습니다.

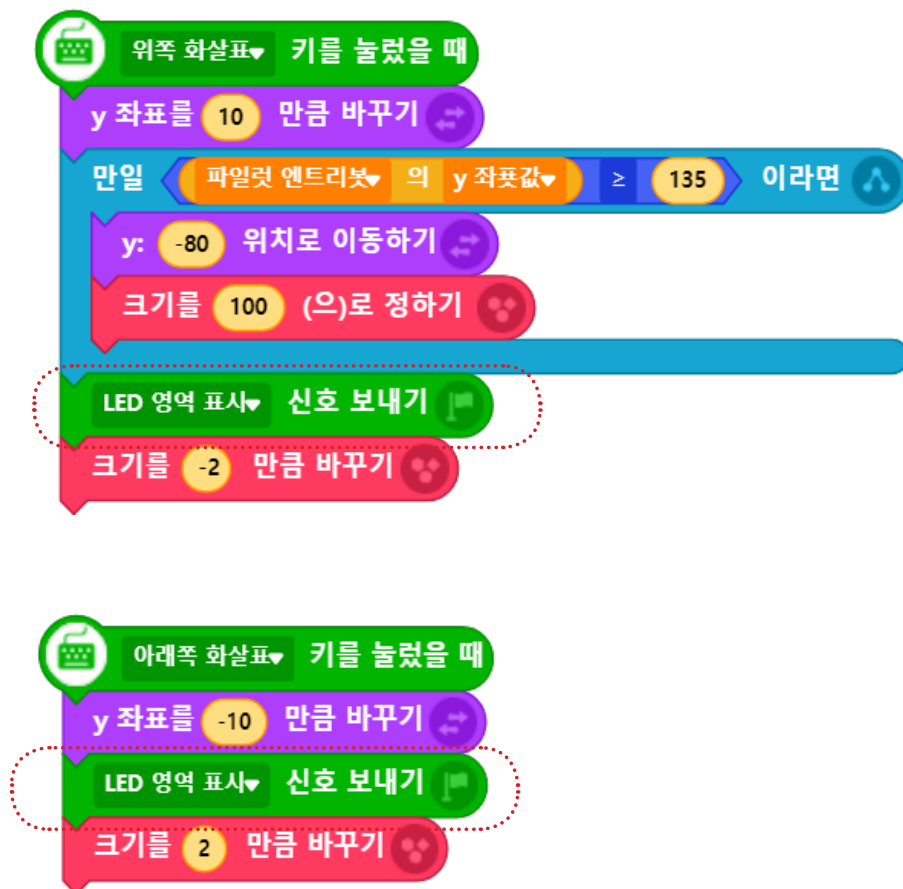
먼저 해당 이벤트가 발생할 경우 호출할 ‘신호’를 만들어 줍니다.

‘파일럿 엔트리봇’을 선택한 다음 속성에서 ‘LED 영역표시’ 신호를 추가합니다.



[그림 2-6] LED 영역 표시 신호 만들기

‘파일럿 엔트리봇’에 위쪽 화살표키를 눌렀을 때, 아래쪽 화살표 키를 눌렀을 때 ‘LED 영역표시’ 신호 보내기 블록을 추가합니다.





다음으로 '파일럿 엔트리봇'의 y좌표 영역범위에 따라 LED 색깔이 표시되도록 아래의 블록을 추가합니다.

LED 영역 표시 신호를 받았을 때

만일 <파일럿 엔트리봇의 y 좌표값> > -135 그리고 <파일럿 엔트리봇의 y 좌표값> ≤ -45 이라면

포트1 LED 4 RGB(0 255 0)

초록 음(를) 말하기

만일 <파일럿 엔트리봇의 y 좌표값> > -44 그리고 <파일럿 엔트리봇의 y 좌표값> ≤ 46 이라면

포트1 LED 4 RGB(255 255 0)

노랑 음(를) 말하기

만일 <파일럿 엔트리봇의 y 좌표값> > 47 그리고 <파일럿 엔트리봇의 y 좌표값> ≤ 134 이라면

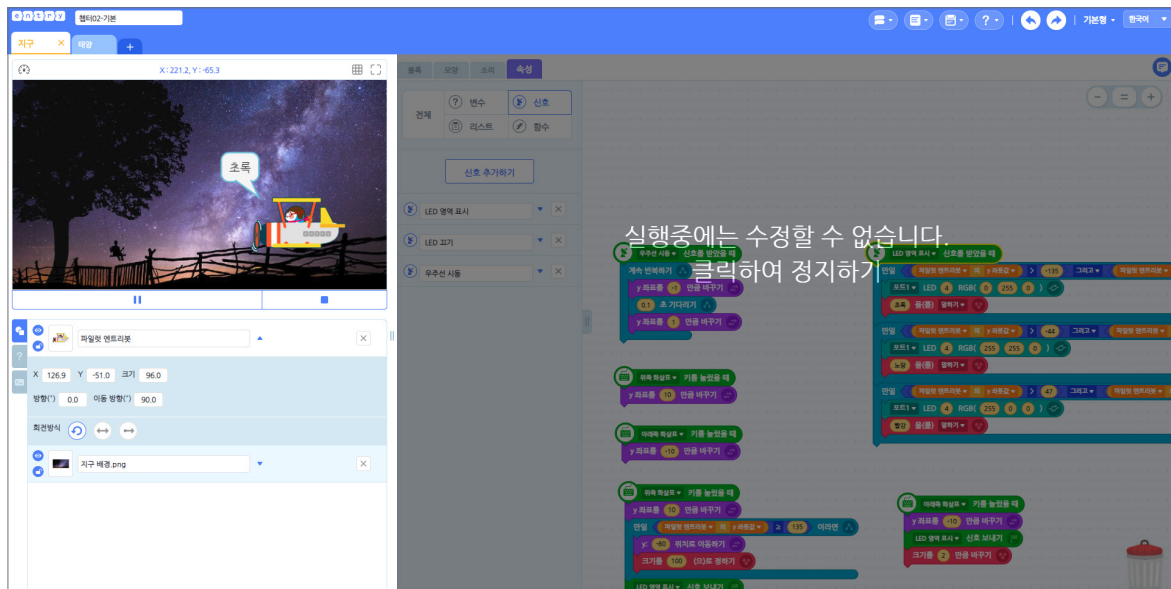
포트1 LED 4 RGB(255 0 0)

빨강 음(를) 말하기

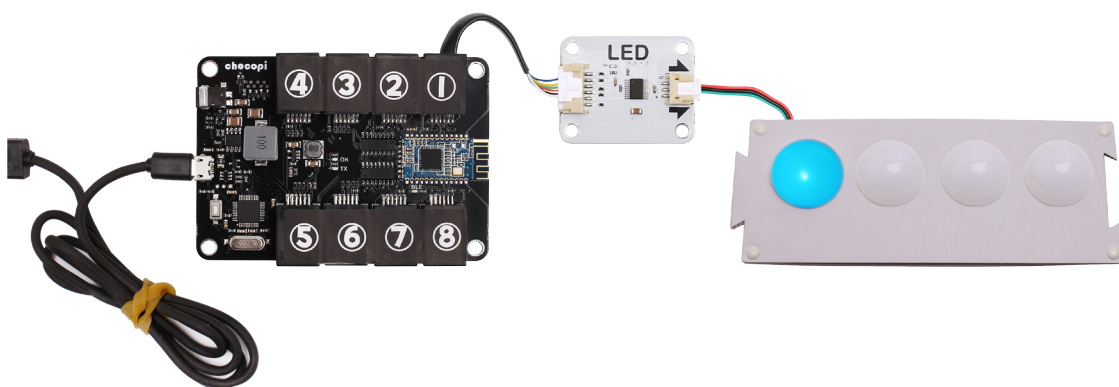
④ 프로그램 테스트와 디버깅

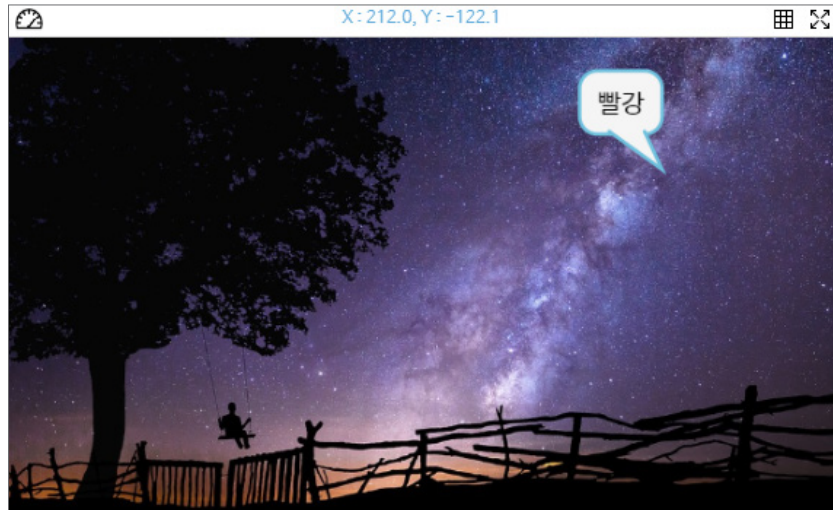
이제 실제 프로그램을 실행하여 테스트해 봅시다.

스페이스키를 누르면 초코파이보드의 LED 모듈 1,2,3에 차례대로 지정한 색깔의 불이 켜집니다. 다음으로 ‘파일럿 엔트리봇’이 시동이 걸리면서 위아래로 약간씩 흔들리는 것이 보입니다. 화살표를 위, 아래로 이동시키면 다음과 같은 실행화면을 볼 수 있습니다.



[그림 2-7] 프로그램 실행 테스트 화면





그런데 화살표를 계속 위로 올려서 y좌표값이 135를 벗어나면 '파일럿 엔트리봇'이 다시 원위치로 돌아오는 것을 확인할 수 있습니다. 오브젝트를 계속 위로 올리는 것을 반복하다 보면 '파일럿 엔트리봇'이 점점 작아져서 마침내 보이지 않게 됩니다.

이러한 문제는 어떻게 해결할 수 있을까요?

오브젝트의 크기를 원래대로 돌려놓아야 합니다.

따라서 크기를 100 (으)로 정하기 블록을 위쪽화살표 키를 눌렀을 때 블록 아래에 다음과 같이 추가로 삽입합니다.



SUMMARY 써머리~

- 제시된 문제를 해결하기 위해서는 먼저 절차에 따라 시각적으로 표현하는 훈련이 중요합니다. 만들어진 이벤트를 스크립트로 표현할 때 오브젝트에 연결된 이벤트를 연결해서 스크립트로 표현합니다.
- 오브젝트간 상호작용은 오브젝트의 속성을 이용합니다.
- 피지컬컴퓨팅에서도 위와 같은 방법을 적용합니다. 센서들과 오브젝트를 연결하는 것도 이벤트 중심으로 연결하여 구성합니다.
- 프로그래밍 테스트를 통해서 개선해야할 사항들을 확인합니다.

개선해야할 사항을 적어봅시다.



미션을 해결해봅시다!

이전 섹션에서 다루었던 내용을 기초로 하여 가상의 이야기를 바탕으로 따라 해보거나 자신만의 아이디어를 추가해서 더욱 발전시켜볼 수 있는 섹션입니다.

MISSION 01 우주장비 움직이기

엔트리봇이 타고 있는 파일럿은 지구를 떠나 태양을 탐험하기 위해 가고 있습니다. 새로운 행성을 찾아가기 위해서는 반드시 여러 가지 장비들이 필요합니다.

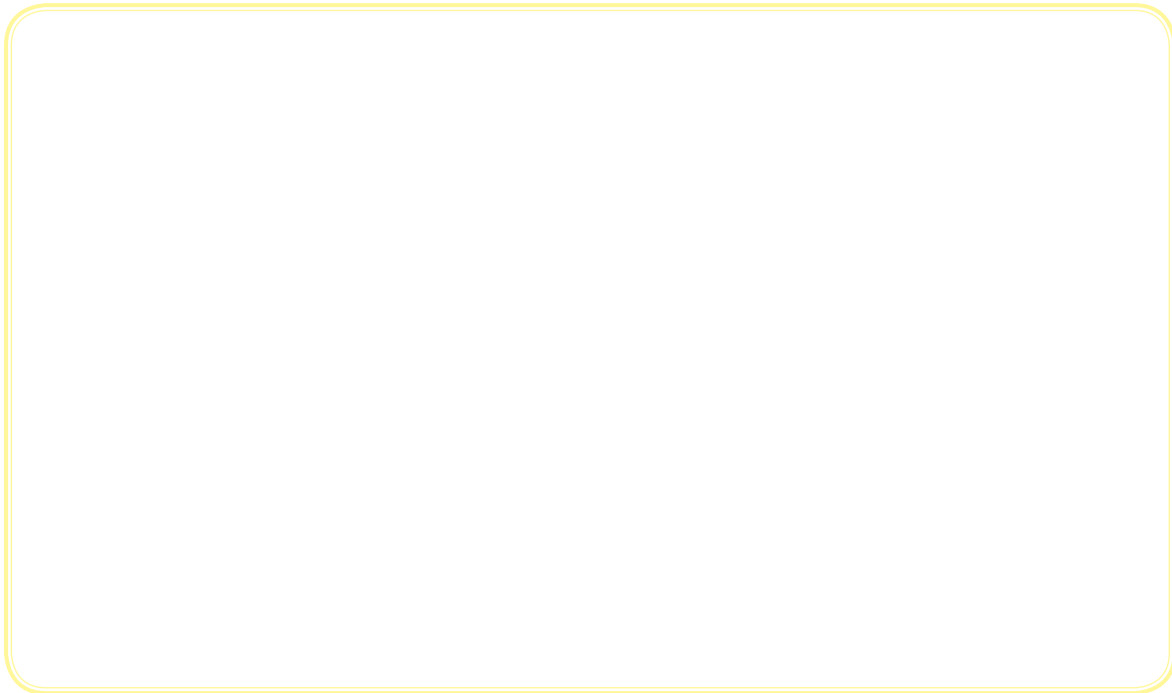
여러분이 생각하는 장비는 어떠한 장비일까요? 그림으로 그려보아도 좋고 여러분의 생각을 적어봅니다.

① 우주선 장비 이벤트 디자인하기

우주선에서 드론을 띄워서 태양을 탐험하는 방법을 생각해 봅시다.

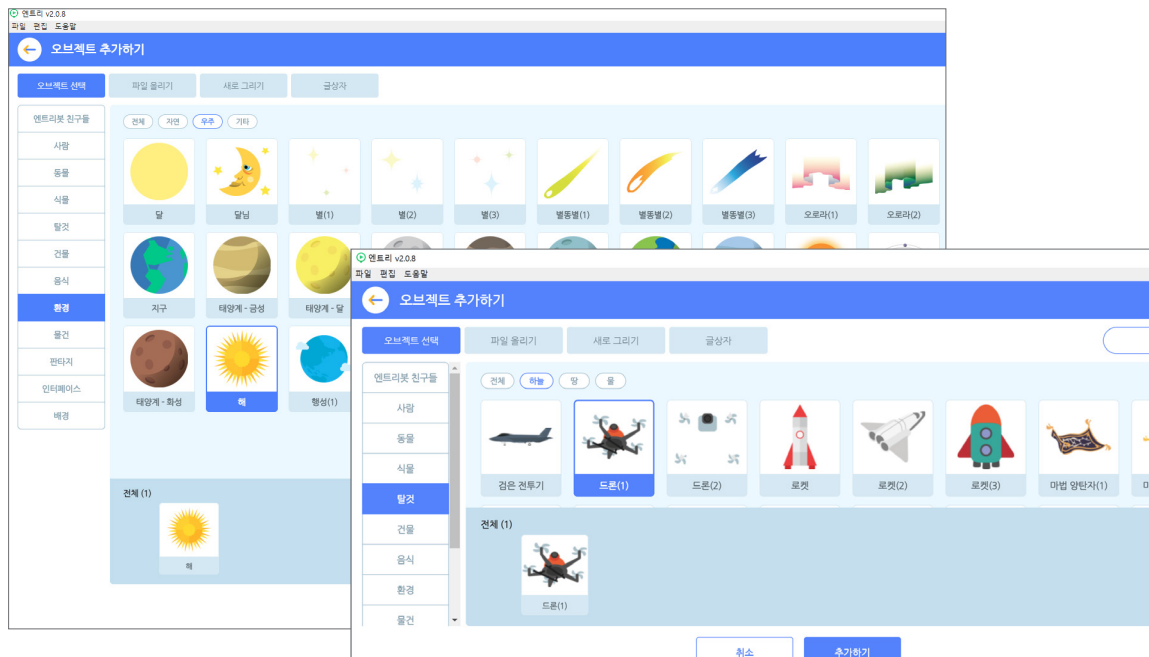
- 태양은 1초에 6도만큼 회전하도록 합니다.
- 숫자 1을 누르면 파일럿 엔트리봇에서 드론이 나타나도록 합니다.
- 숫자 0을 누르면 드론이 사라지게 합니다.
- W,A,S,D 키를 활용하여 드론을 움직일 수 있도록 합니다.
- 숫자 9를 누르면 드론이 파일럿 엔트리봇으로 돌아오도록 합니다.
- 태양과 부딪히면 LED 1번에서 빨간색 불을 켜고 위험을 알립니다.

위에서 제시한 문제를 해결할 수 있도록 구조화하여 표현해 봅시다.



② 오브젝트 추가하기

오브젝트 추가하기 > 환경에서 ‘해’를 추가합니다. 마찬가지로 탈것 > 하늘에서 ‘드론’을 추가합니다.

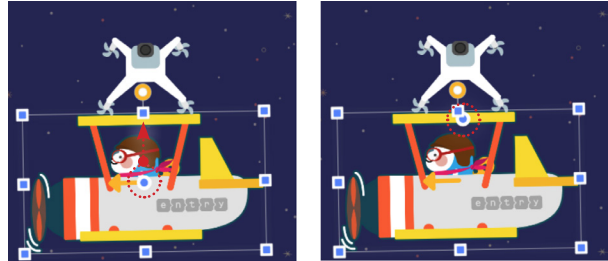




아래 화면과 같이 오브젝트들을 배치합니다.

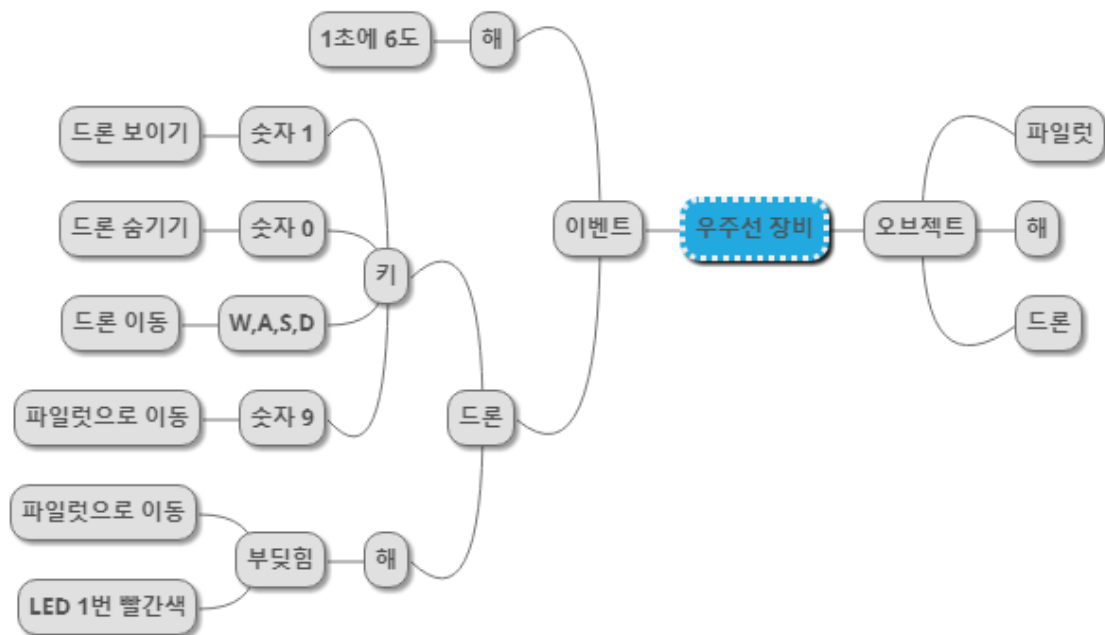


드론을 파일럿에 붙어있게 하기 위해 파일럿의 중심점을 아래와 같이 위쪽으로 이동시킵니다.



③ 오브젝트와 이벤트 연결하기

제시된 문제를 해결하기 위한 알고리즘을 간략하게 정리하면 다음과 같습니다.

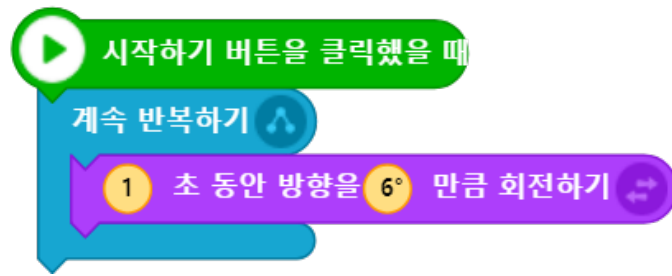


[그림 2-8] 이벤트 요약

해 오브젝트는 1초에 6도씩 회전시킵니다. 드론 오브젝트와 연결된 이벤트는 크게 드론의 모습, 드론의 이동, 해 오브젝트와의 부딪힘 확인, 파일럿으로 이동하는 이벤트로 구성되어 있습니다.

또한 초코파이보드를 이용해서 드론이 해 오브젝트와 부딪치면 LED 1번이 빨간색 불이 켜지게합니다.

먼저 시작버튼을 클릭하면 해가 자동으로 회전하도록 표현해 봅시다.



1초동안 6도만큼 회전하면 60초에 360도를 회전하게 되므로 해는 1분에 한바퀴를 회전하게 되는 셈입니다. 실제 속도를 계산한다면 그야말로 엄청나겠군요. 그러나 실제 태양이 그렇게 회전하지 않습니다. 태양의 자전속도는 1초에 1.997km입니다.

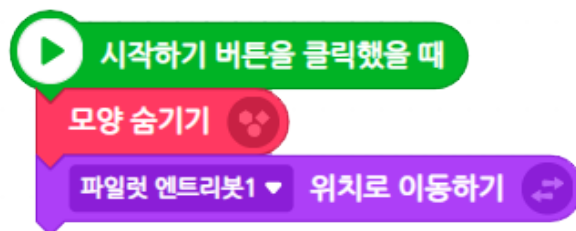
태양 행성에 들어가기 전에 태양에 대해서 좀 더 알아보겠습니다. 상세한 내용은 위키백과의 내용을 참조하였습니다.

- 수명 123억 6500만년
- 태양계의 중심에 있고 실제 공전 속도는 217km/s 이며 상대속도는 20km/s
- 평균 지름은 1.392×10^6 km 지구의 109배
- 부피는 지구의 1,300,000배로 1.41×10^{18} km³
- 태양이 지구로 보내준 햇빛 에너지로 식물이 광합성을 일으켜 성장하기 때문에 지구상의 동식물의 생존이 가능함

그러나 수집된 정보는 2018년의 정보일 뿐, 현재는 서기 3030년, 태양 속에 어떠한 생물이 존재하는지 아무도 모르는 일입니다.

따라서 우리는 특수하게 제작된 드론을 띄워야 합니다.

시작버튼을 클릭하면 먼저 드론을 보이지 않게 해야 합니다. 그리고 드론의 위치가 항상 파일럿에 붙어 있도록 설정해줘야 합니다. 드론 오브젝트에 적용한 블록 프로그래밍을 참고하세요.

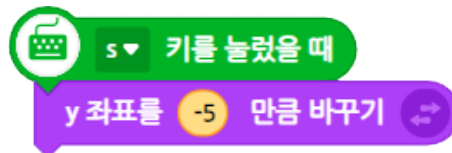
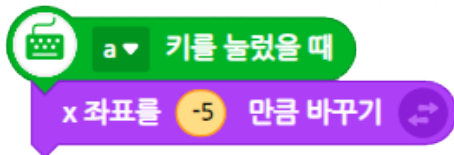




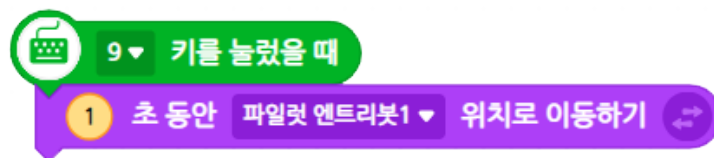
다음으로 숫자 1번을 누르면 드론이 표시되고 숫자 0을 누르면 드론이 표시되지 않도록 합니다.



드론을 이동시키기 위해서 W,A,S,D키를 활용하여 다음과 같이 이동할 수 있는 블록들을 추가합니다.

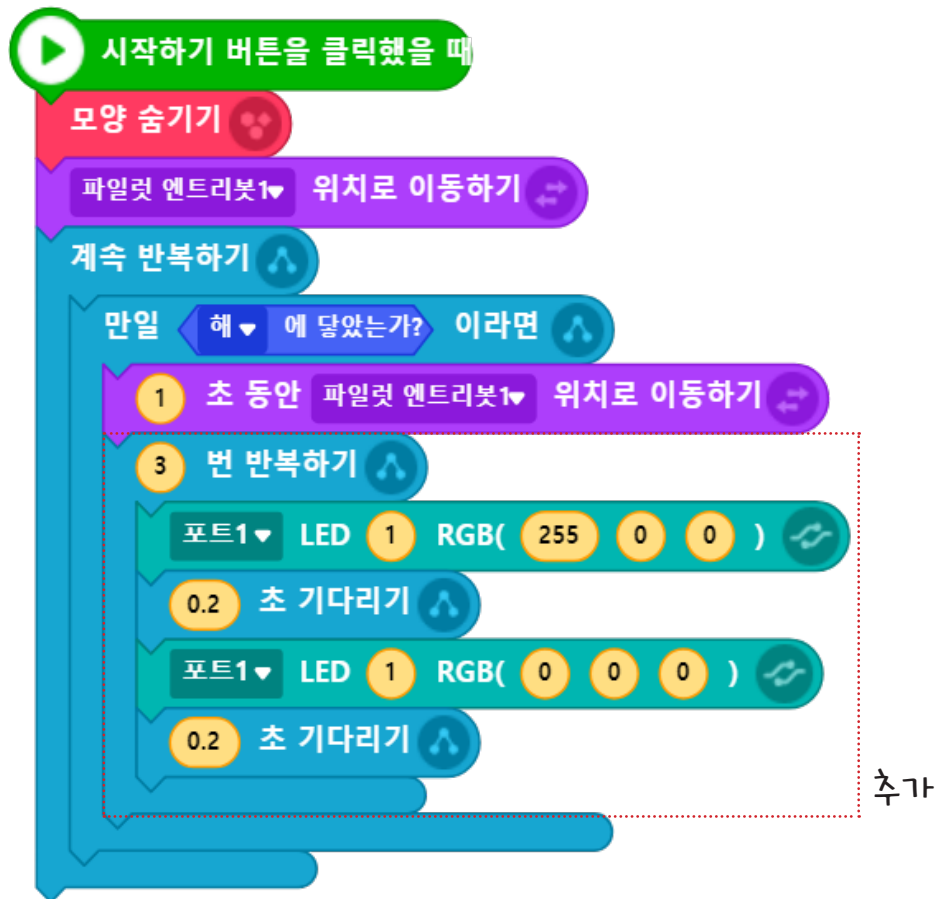


숫자 9를 누르면 파일럿으로 이동하기 위한 프로그램을 만들어 봅시다.



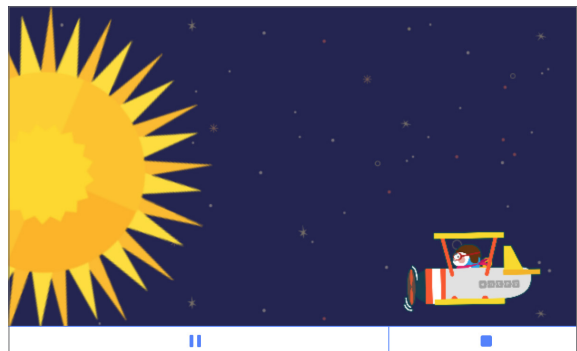
이제 W,A,S,D 키를 활용하여 드론이 해로 가까워지게 합니다. 만약 해와 부딪히면 다음과 같이 프로그래밍하여 드론을 파일럿으로 돌아오게 합니다.



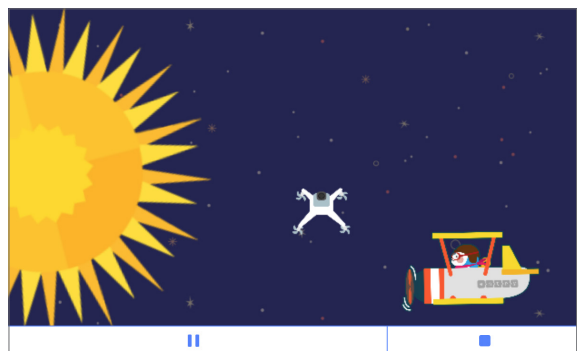


④ 프로그램 테스트와 디버깅

시작 버튼을 클릭하여 프로그램을 실행하여 봅시다.
다음과 같이 태양이 회전하는 것이 보일 것입니다.



숫자 1을 누르면 드론이 나타나고 W,A,S,D키로 이동시킬 수 있습니다.





⑤ 프로그램 개선하기

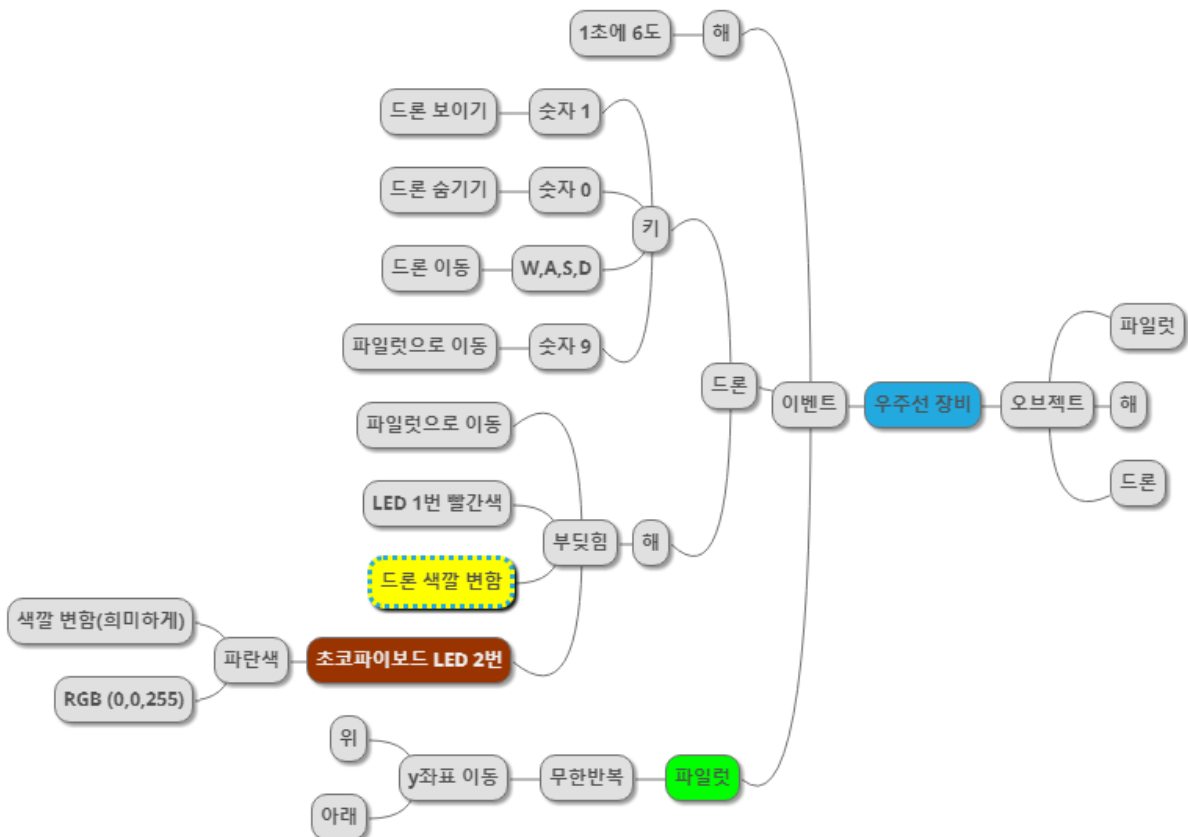
이번에는 다음의 몇 가지 사항을 추가해서 프로그램을 보완해 봅시다.

- 파일럿이 운행 중이면 자동으로 위아래로 움직이게 만듭니다.
- 드론이 태양과 부딪히면 드론의 색깔이 변하게 합니다.
- 드론의 상태를 표시하기 위해 초코파이보드 LED 2번에서 파란색 불빛을 점점 희미하게 합니다.

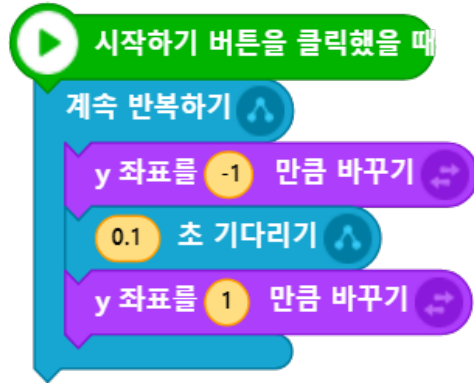
어떻게 프로그래밍 해야 할까요?

새로운 조건이 추가되었을 때 우리가 흔히하는 실수는 추가된 이벤트들에 대해서 알고리즘을 수정하지 않고 바로 블록들을 끌어다 놓으려고 한다는 것입니다. 간단한 프로그램은 쉽게 수정이 가능하지만 프로그램이 점점 복잡해질수록 여러분은 본인이 무엇을 추가하고 수정하는지 이해 못하는 경우가 많이 있습니다. 따라서 다시 시각적으로 한 번 이벤트들을 추가 보완해서 알고리즘을 개선해 보시기 바랍니다.

추가된 문제를 해결하기 위해서 파일럿의 움직임, 드론 색깔 변화, 초코파이보드 LED 2번 색깔 제어 부분을 추가해서 다음과 같이 완성하였습니다.



먼저 파일럿이 운행 중에 움직임이 자연스러워 보이도록 프로그래밍 해 봅시다.

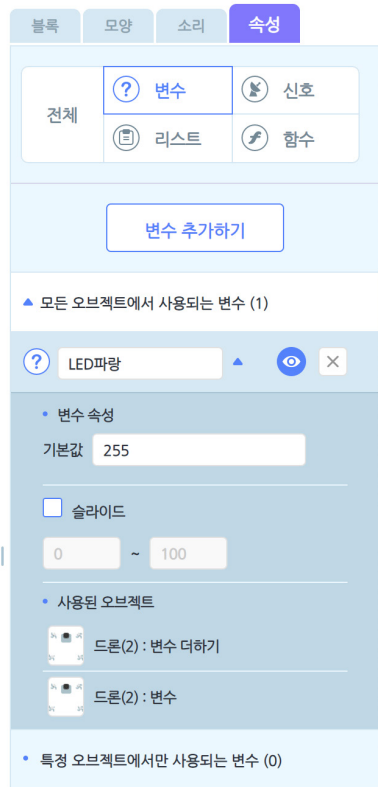


드론이 ‘해’에 부딪혔을 때
드론의 색깔을 변하게 해봅시다.



드론이 ‘해’에 부딪혔을 때 드론의 상태를 나타내는 초코파이보드의 LED 2번의 파란색이 점점 희미해지게 해 봅시다.

파란색은 RGB(0,0,255)로 표시합니다. 드론이 부딪힐 때마다 희미하게 하려면 255값이 점점 작아지면 됩니다. 한 번 부딪힐 때마다 10씩 줄어들게 해 봅시다. 그러기 위해서는 숫자를 저장하는 변수가 필요합니다. 속성탭에서 변수를 클릭하고 변수명을 'LED파랑'이라고 입력한 뒤 기본값은 255로 설정합니다.



RGB의 값에 LED파랑 변수를 삽입하여 수정합니다.
LED파랑은 기본값이 255로 설정되어 있으므로
‘해’ 오브젝트에 부딪힐 때마다 10씩 감소하게 됩니다.

MISSION 02 나만의 우주장비

파일럿은 태양 행성을 탐사할 준비가 되었습니다.

드론을 움직여서 태양으로 가까이 갈 수 있게 되었고 드론의 상태를 확인하기 위해 초코파이보드의 LED 센서를 이용했습니다.

이 프로그램에 어떤 요소를 추가하면 보다 게임처럼 만들 수 있을까요?

아래의 아이디어를 보고 여러분만의 프로그램을 만들어 주시기 바랍니다.

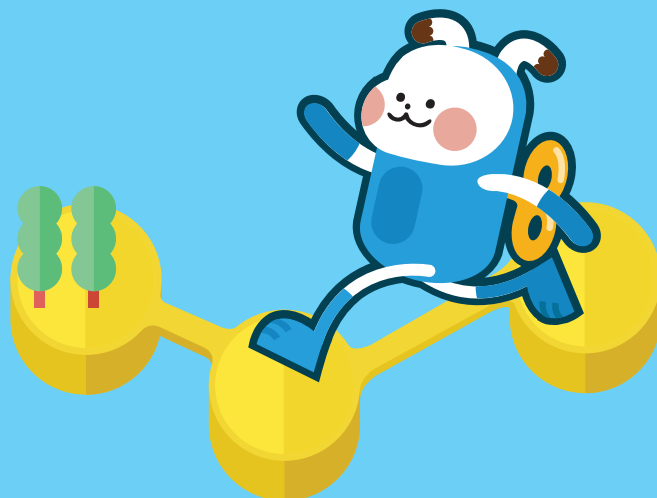
- 드론의 횟수를 정하자
- 파일럿이 배경을 벗어나지 못하게 하자
- 나도 모르는 외계의 존재를 넣어보자

이제 상상은 여러분의 몫입니다.

03

우주를 줄게

터치, 센서블록 활용하기



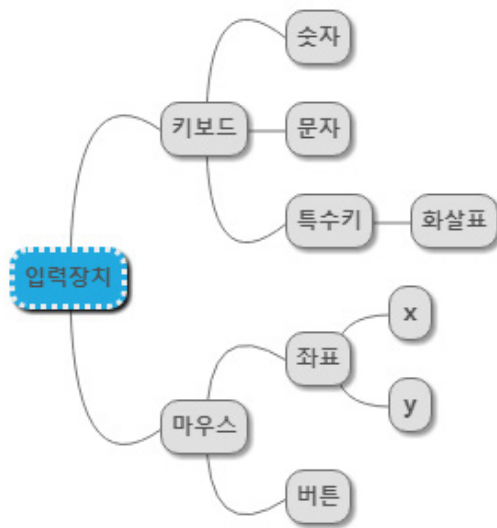


문제가 무엇인가요?

컴퓨팅의 과정은 입력, 처리, 출력의 과정이라고 설명하였습니다.

즉 정보를 수집하는 과정이 입력과정이 되겠지요. 컴퓨터가 받아들일 수 있는 정보의 종류는 여러 가지가 있습니다.

일반적으로 많이 사용하는 키보드와 마우스에 대해서 생각해 봅시다.



[그림 3-1] 키보드, 마우스 입력장치 속성

[그림 3-1]과 같이 컴퓨터는 키보드를 통해서 숫자, 문자 또는 특수키인 화살표 이동키 등을 입력받을 수 있습니다. 또한 마우스 장치를 통해서 마우스의 x, y좌표와 버튼의 클릭 여부를 입력 받을 수 있습니다.

그렇다면 엔트리 프로그램에서는 키보드와 마우스 입력과 관련하여 어떠한 블록들을 사용해서 정보를 수집하게 되는 것일까요?



문제를 분해해봅시다!

먼저 키보드 관련 입력정보를 처리하는 방법에 대해서 알아봅시다. 관련된 엔트리 블록은 다음과 같습니다.



키를 눌렀을 때

블록을 통해 사용자가 입력 한 키의 정보를 수집하거나

안녕!

을(를) 묻고 대답 기다리기 ?

을 통해서 오브젝트가 어떠한 질문을 던지고

이에 대한 정보를 **대답** 이라는 블록에 저장하는 방법이 있습니다.

이벤트 다루기

그렇다면 위의 두 가지 블록을 연결하여 생각해 봅시다.

'a'키를 누르면 오브젝트가 현재의 온도를 물어보고 비교하는 프로그램을 만들려면 어떠한 것들이 필요할까요?



먼저 숫자를 입력받는 인물로는 '락커', 배경으로는 '생일파티' 오브젝트를 선택했습니다. 이벤트는 키보드의 숫자를 입력받는 것으로 설정합니다. 사용자는 숫자만 입력하고 문자는 입력하지 않는다고 가정하였습니다.

스크립트로 표현하기

전체적인 스크립트는 다음과 같이 진행합니다.

- 시작버튼을 클릭하면 컴퓨터는 'a'키가 입력되는 것을 기다립니다.
 - 'a'키가 입력되면 입력된 숫자값을 가지고
 - 초코파이보드의 센서 블록에 입력되는 온도값과 비교합니다.
 - 입력된 숫자값과 센서 블록이 측정한 현재의 온도를 비교하여 온도가 높거나 낮음에 따라서 '온도높다' 또는 '온도낮다' 신호를 전달하여 프로시저를 호출합니다.
 - 신호전달 받은 각각의 프로시저는 설정된 배경을 바꾸고 해당되는 음악을 재생합니다.

오브젝트 블록 프로그래밍

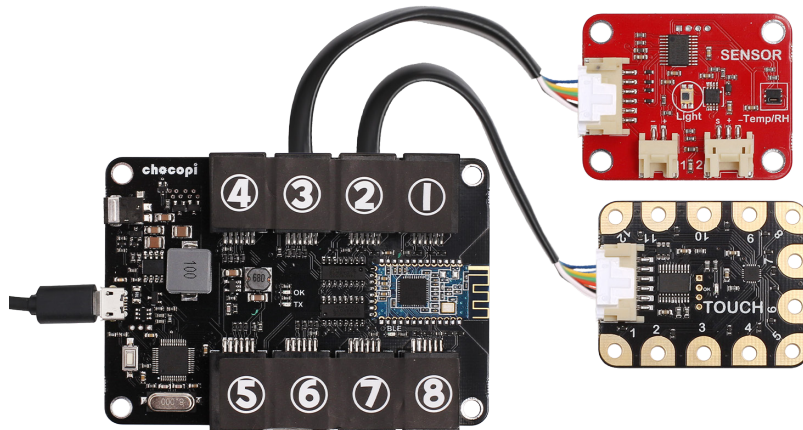
이제 생각한 알고리즘을 블록 프로그래밍으로 차례대로 전환해 봅시다.

생일파티와 락커 오브젝트를 다음과 같이 디자인합니다. 물론 여러분의 취향대로 다른 오브젝트를 가져와서 구성해도 좋습니다.

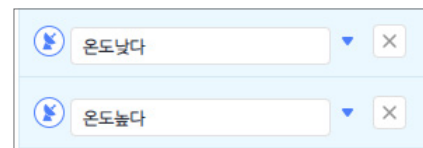




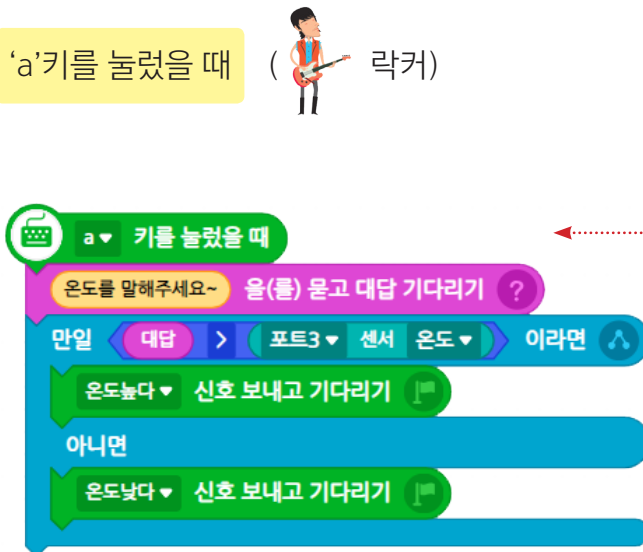
초코파이보드의 센서 블록을 3번 포트에 연결한 다음 엔트리와 연결합니다.



'온도높다'와 '온도낮다' 신호(프로시저)를 추가합니다.



'a'키를 눌렀을 때 (락커)



'a'키를 눌렀을 때 다음과 같이 프로그래밍 합니다.

'a'키를 누르면 락커는 '온도를 말해주세요~'라고 물어보고 키보드의 입력을 기다립니다. 사용자가 숫자만 입력한다는 가정하에 대답에 들어온 변수값과 '초코파이보드'의 센서 블록의 온도값과 비교합니다.

입력된 온도값이 센서의 온도값보다 높으면 '온도높다' 신호를 전달합니다.

입력된 온도값이 센서의 온도값보다 낮으면 '온도낮다' 신호를 전달합니다.

여기서 잊지 말아야 할 것은 신호는 전체의 오브젝트에 전달하는 것입니다. 즉 락커 오브젝트뿐 아니라 배경에 있는 오브젝트에도 해당 신호(프로시저)를 동시에 전달합니다. 신호를 전달 받으면 다음과 같이 배경을 바꾸고, 음악을 재생시킵니다.



락커 오브젝트에서는 어떠한 작업을 해야되는지 좀 더 알아보니다.

'온도 높다' 신호를 받았을 때 (락커)



위의 프로그램 코딩이 어떻게 구성되어 있는지 살펴봅시다.

'온도높다' 신호를 받았을 때 오른쪽 그림과 같이 'rock_happy' 사운드를 재생합니다.



만약 사운드가 재생 중일 때 배경을 변화시키기 위해서 소릿값이 0보다 큰동안 입력한 온도와 현재의 온도를 비교해서 말해주는 블록을 넣습니다.



다음으로 락커오브젝트의 속성에 있는 모양탭, 소리탭에는 [그림 3-2]와 같은 내용이 있어야 합니다.

각 탭에 요소를 추가하는 것은 엔트리 홈페이지에서 알아보시기 바랍니다. 소리파일은 본인이 갖고 있는 것을 사용할 수 있습니다.



[그림 3-2] 락커 오브젝트 모양탭, 소리탭에 추가할 요소들



모양을 추가했다면 기타치는 사람 모양을
0.2초마다 번갈아 가면서 바뀌도록 프로그래밍
합니다.



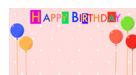
'온도낮다' 신호를 받았을 때 (락커)



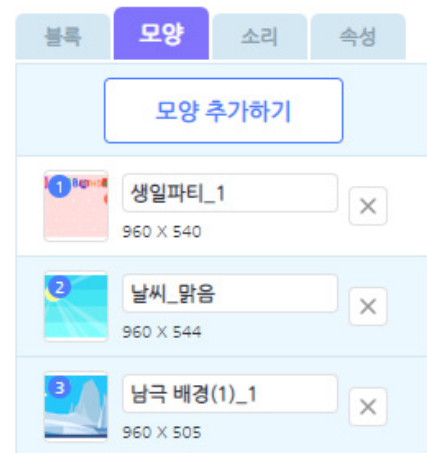
마찬가지로 락커 오브젝트에 '온도낮다' 신호에 대해서도 다음과 같이 블록
프로그래밍합니다.



'온도 높다','온도 낮다' 신호를 받았을 때 (생일파티)



일단 락커 오브젝트에서 '온도높다', '온도낮다' 이벤트를
처리하게 되면 사운드가 재생되고 있을 것입니다.
따라서 배경에서는 해당 신호를 받았을 때 사운드의
소릿값을 확인하여 배경을 바꿔주는 것입니다.
또한 배경 오브젝트에는 오른쪽 그림과 같은 배경이
추가되어 있어야 합니다.



'생일파티' 배경 오브젝트에 모양을 추가 한 후 '온도높다', '온도낮다' 신호를 받았을 때 해야할 이벤트를 처리하는 블록 프로그래밍을 합니다.



여러분이 생각하는 좀 더 나은 이벤트는 없을까요?

제시된 문제들을 가지고 게임으로 만들기 위해서는 어떠한 것들이 필요할까요?



하나씩 연결해봅시다!

오브젝트간 상호작용

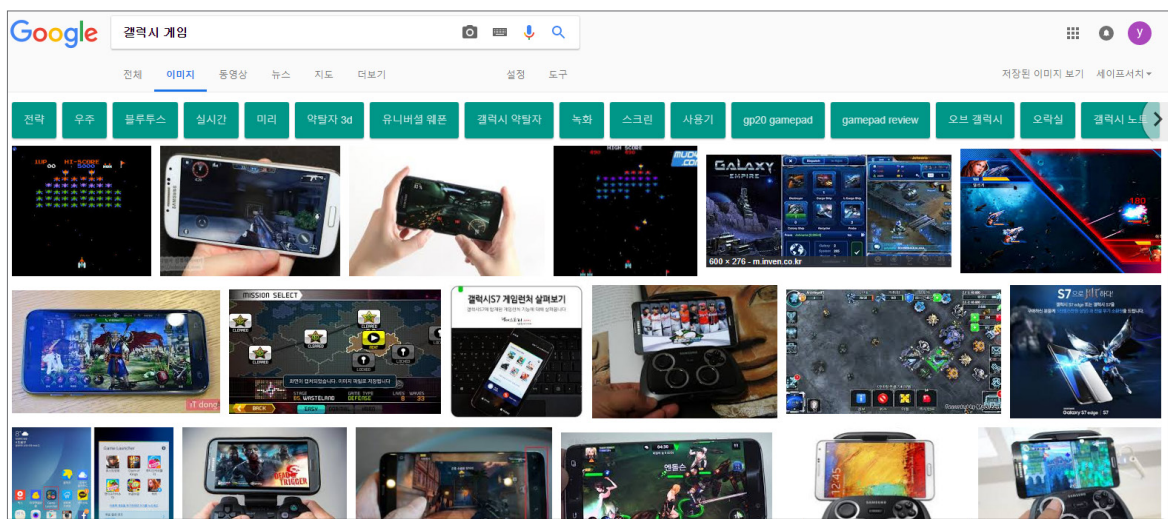
온도를 입력하고 입력된 온도와 현재 온도를 비교해서 오브젝트의 모양을 바꾸거나 사운드를 제어하는 프로그램을 만들어 보았습니다.

오브젝트간에 상호작용하는 방법에 대해서 자연스럽게 배울 수 있고 프로그램을 좀 더 재미있게 하기 위해서 게임 요소를 넣어봅시다.

게임하면 어떤 것이 떠오르나요? 제가 가진 게임의 추억을 소개하겠습니다. 1980년대였어요. 그 때만해도 친구들은 대부분 태권도, 미술, 피아노학원 다닐 때 였는데 우리 동네에는 마침 컴퓨터학원이 생겨서 저는 그 곳을 다녔습니다. 키보드 자판을 통해 무엇인가를 입력하면 화면에 글자가 나타는 것이 참 신기했습니다.

그런데 어린 나를 잡아끈 것은 흥미로운 컴퓨터학원보다 그 건물 1층에 있는 오락실이었습니다. 오락실에서 '50원'짜리 동전을 넣고 게임 한 판을 하기 위해서 일주일 내내 기다렸던 생각이 납니다. 그 때 당시 재미있었던 게임이 바로 '갤럭시'라는 게임이었습니다.

그렇다면 어떤 게임이었는지 한 번 검색해 봅시다.



[그림 3-3] 구글에서 갤럭시 게임으로 검색한 화면

그러면 갤럭시 게임을 좀 더 자세히 살펴봅시다. 물론 앞서서 공부했던 온도를 입력 받는 프로그램을 온도 맞추기 게임으로 바꾸기 위해 필요한 정보를 찾고 수집하는 단계입니다.



[그림 3-4] Galaxian 게임 화면

(출처: <https://mlpforums.com/topic/147618-your-all-time-favorite-classic-arcade-games/>)

[그림 3-4]를 자세히 살펴보고 게임에 들어가는 요소를 정리하면 대략 다음과 같습니다.

- 일단 최고점수를 의미하는 HIGH SCORE가 있습니다.
- 게임에 투입되는 비행기는 모두 세 대입니다.
- 적들 모두 세 개의 형태를 가지고 있습니다. 각각의 형태별로 공격 유형이 다릅니다.
- 적은 특수한 그물 아이템을 사용하여 우리편 비행기를 잡아서 끌어올릴 수 있습니다.
- 납치된 비행기를 가지고 있는 적을 파괴하면 그 비행기가 우리편으로 내려와서 둘이 합쳐서 공격할 수 있습니다.

수집된 정보를 다시 간단하게 요약하면 어떻게 정리할 수 있을까요?

- 비행기는 세 대
- 최고점수가 있음
- 적은 우리를 공격하거나 우리편 비행기와 부딪히면 파괴됨
- 특수한 아이템을 사용할 수 있음

위와 같이 크게 네 가지의 유형으로 다시금 정리하였습니다.



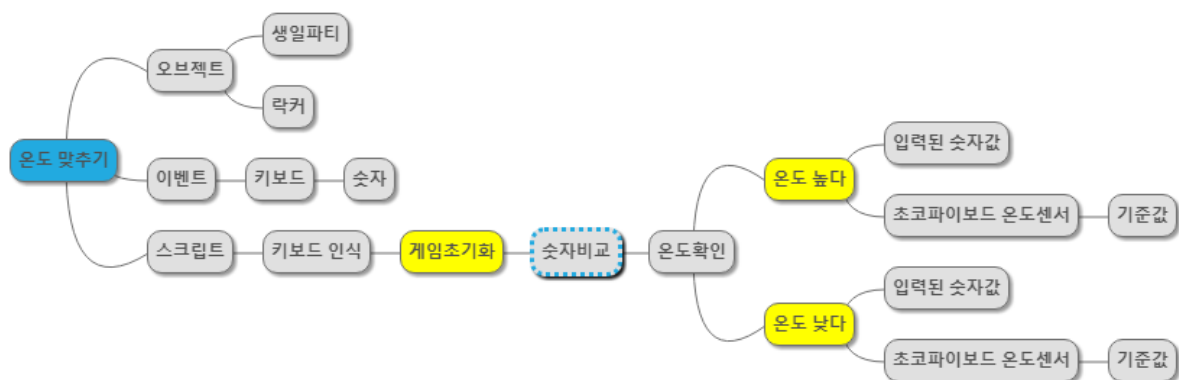
마찬가지로 앞서 공부한 온도를 입력받는 프로그램을 온도 맞추기 게임으로 바꾸려면 무엇이 필요할까요?

- 2번 포트에 연결된 터치센서의 1번을 터치하여 게임을 시작
- 락커 생명은 3개이고 이것을 LED의 1, 2, 3번에 빨간색으로 표시함
- 현재온도보다 낮게 입력하면 락커 생명이 하나 줄어들고 점수가 10점 감점됨
- 생명이 줄어들 때마다 LED 번호 3, 2, 1번 순서대로 꺼짐
- 락커의 생명이 0이되면 게임 종료
- 현재온도보다 높게 입력하면 점수가 10점 올라감

설정한 문제를 해결하기 위해서 추가로 처리해야 될 정보들을 생각해 봅시다.

- 락커의 생명을 담아 둘 변수 '락커생명'이 필요합니다.
- 점수가 올라가거나 내려가기 위한 '점수'변수도 필요합니다
- 락커의 생명을 LED에 표시하기 위해 'LED번호'도 필요합니다.

이제 게임설정에서 제시된 문제를 해결하기 위한 이벤트들을 정리하여 봅시다.

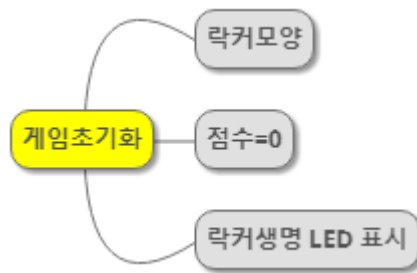


[그림 3-5] 게임 추가 설정

전체적으로 추가된 알고리즘을 살펴보면 다음과 같습니다.

먼저 키보드의 'a'를 누르거나 초코파이보드 터치블록의 터치부 1번을 만지면 게임을 시작합니다. [그림 3-5]에서는 키보드 인식만 표시하였습니다.

게임을 시작하면 게임에 필요한 값들을 초기화하는 '게임초기화'를 호출하여 실행합니다. 이후에 입력된 숫자를 비교하는 알고리즘은 동일합니다.



[그림 3-6] 게임 초기화 이벤트 처리

'게임초기화'신호를 받았을 때 (락커)



게임초기화 신호를 받으면 소리가 재생되고 있는 경우 정지해야합니다.

그다음으로 락커의 모양을 처음 설정한 모양으로 바꾸고 점수를 0으로 초기화 합니다.

락커의 생명을 초코파이보드의 LED로 표시하는데 처음 시작할 때는 LED 1, 2, 3번이 모두 빨간색으로 켜지게 합니다.

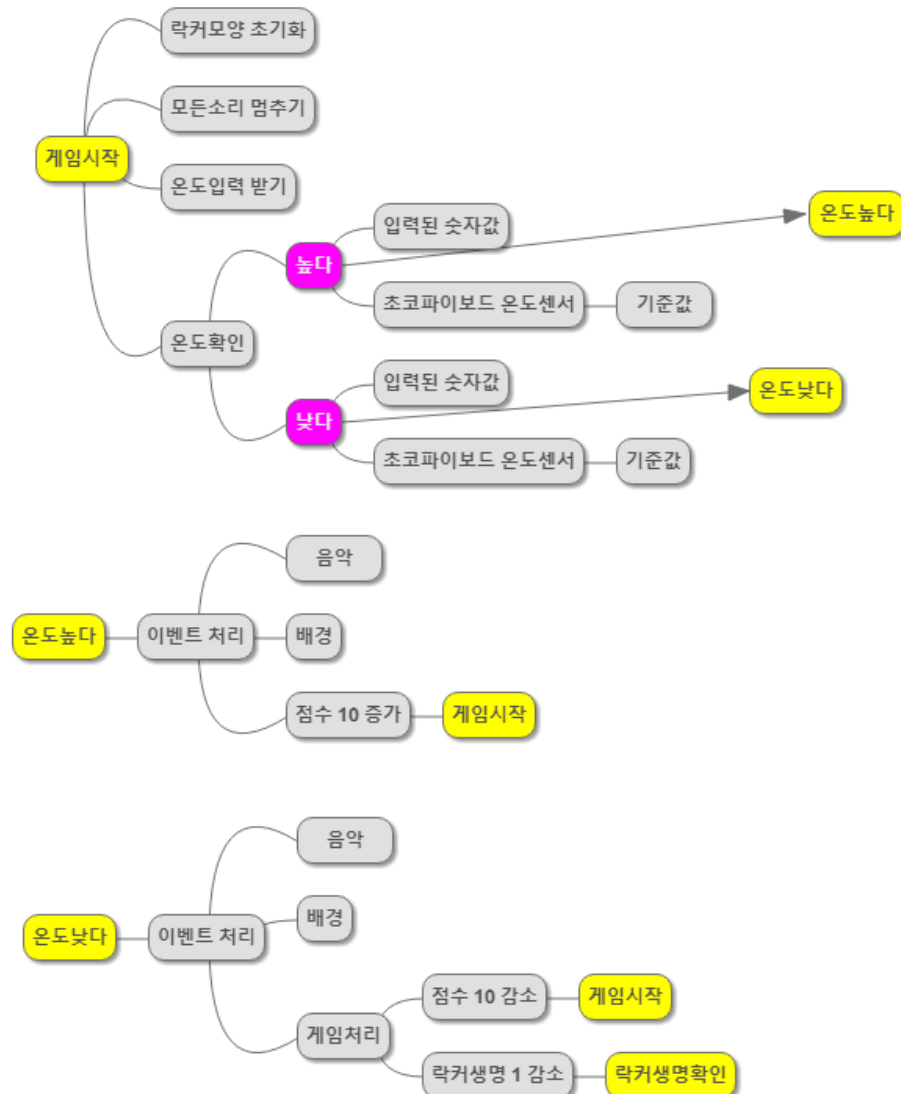
락커생명은 3으로 설정했기 때문에 총 3번을 반복하여 LED번호 변수를 하나씩 올려서 LED 번호순서대로 빨간색으로 켜 줍니다.



[그림 3-7] 게임 초기화 블록 프로그래밍



다음으로 입력한 온도에 따라서 비교할 수 있는 ‘온도높다’, ‘온도낮다’ 신호와 관련하여 추가된 이벤트 처리내용을 정리하면 다음과 같습니다.



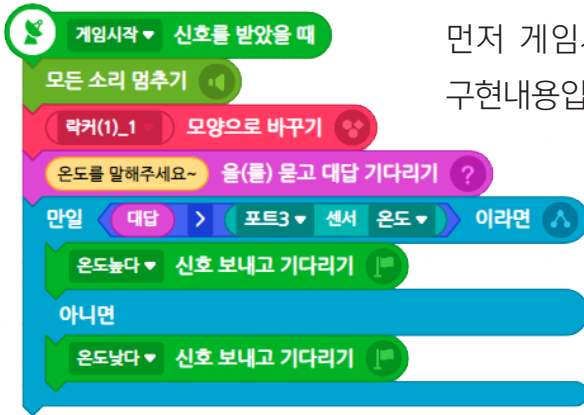
[그림 3-8] 게임시작 신호와 온도높다, 온도낮다 신호간의 관계도

[그림 3-8]과 같이 게임시작신호를 받았을 때 락커의 모양을 기본으로 바꾸고 모든 소리를 멈춥니다. 그 다음 입력받은 숫자와 센서 블록으로 측정한 현재의 온도를 비교합니다. 온도가 높고 낮음에 따라서 온도높다와 온도낮다 신호를 보냅니다.

- 온도높다 신호를 받으면 기존의 음악과 배경은 수정되는 것이 없고 점수를 10점 올리고 다시 게임시작을 호출합니다.
- 온도낮다 신호를 받게되면 기존의 음악과 배경은 수정되는 것이 없습니다.
- 점수를 10점 감점 시킵니다.
- 락커의 생명을 1을 감소 합니다. 그리고 락커생명확인 신호를 보냅니다.

앞에서 정리한 이벤트들이 락커 오브젝트에서 어떻게 블록 프로그래밍되는지 살펴 봅시다.

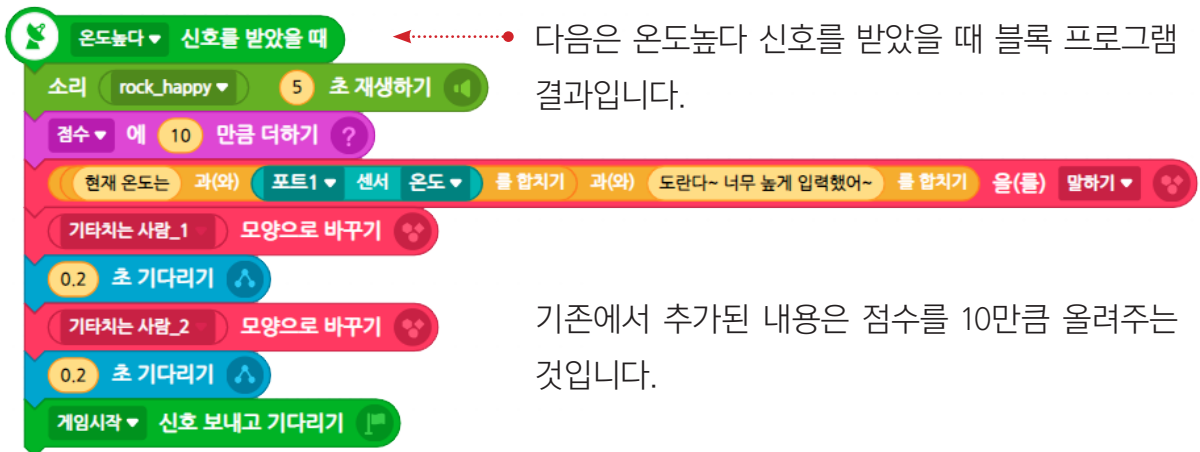
'게임시작' 신호를 받았을 때 (락커)



먼저 게임시작 신호를 받았을 때의 블록 프로그래밍 구현내용입니다.

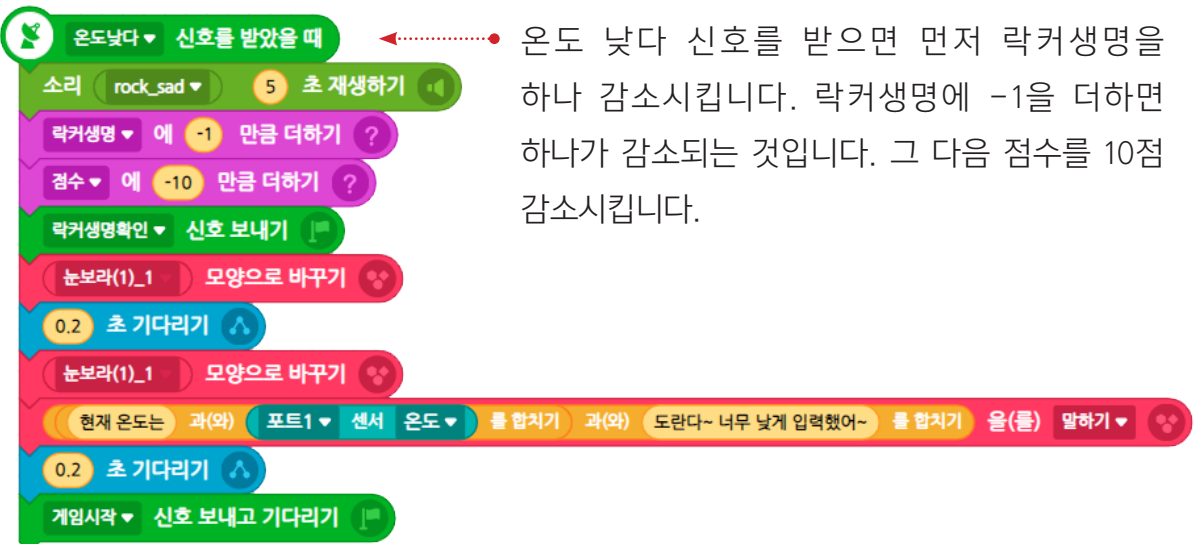
입력된 숫자값(대답)이 현재 센서의 온도값보다 크면 '온도높다', 작으면 '온도낮다'를 호출하게 되어 있습니다.

'온도높다' '온도낮다' 신호를 받았을 때 (락커)



다음은 온도높다 신호를 받았을 때 블록 프로그램 결과입니다.

기존에서 추가된 내용은 점수를 10만큼 올려주는 것입니다.



온도 낮다 신호를 받으면 먼저 락커생명을 하나 감소시킵니다. 락커생명에 -1을 더하면 하나가 감소되는 것입니다. 그 다음 점수를 10점 감소시킵니다.



그 다음 순서로 락커생명확인 신호를 전달합니다. 크게 두 가지의 이벤트를 처리해야 합니다.

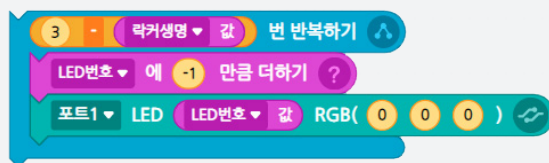
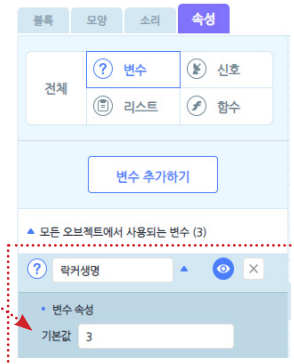
'락커생명확인 신호'를 받았을 때 (락커)

· 락커생명이 하나씩 줄어들 때마다 이것을 실제로 표현하기 위해 초코파이보드에 연결된 LED가 마지막에 3번부터 차례로 꺼지도록 합니다.

· 락커생명의 값이 0이면 게임이 종료되며 다음장면으로 넘어갑니다.

특히 두 번째 이벤트 처리하는 과정을 살펴보면 다음과 같습니다.

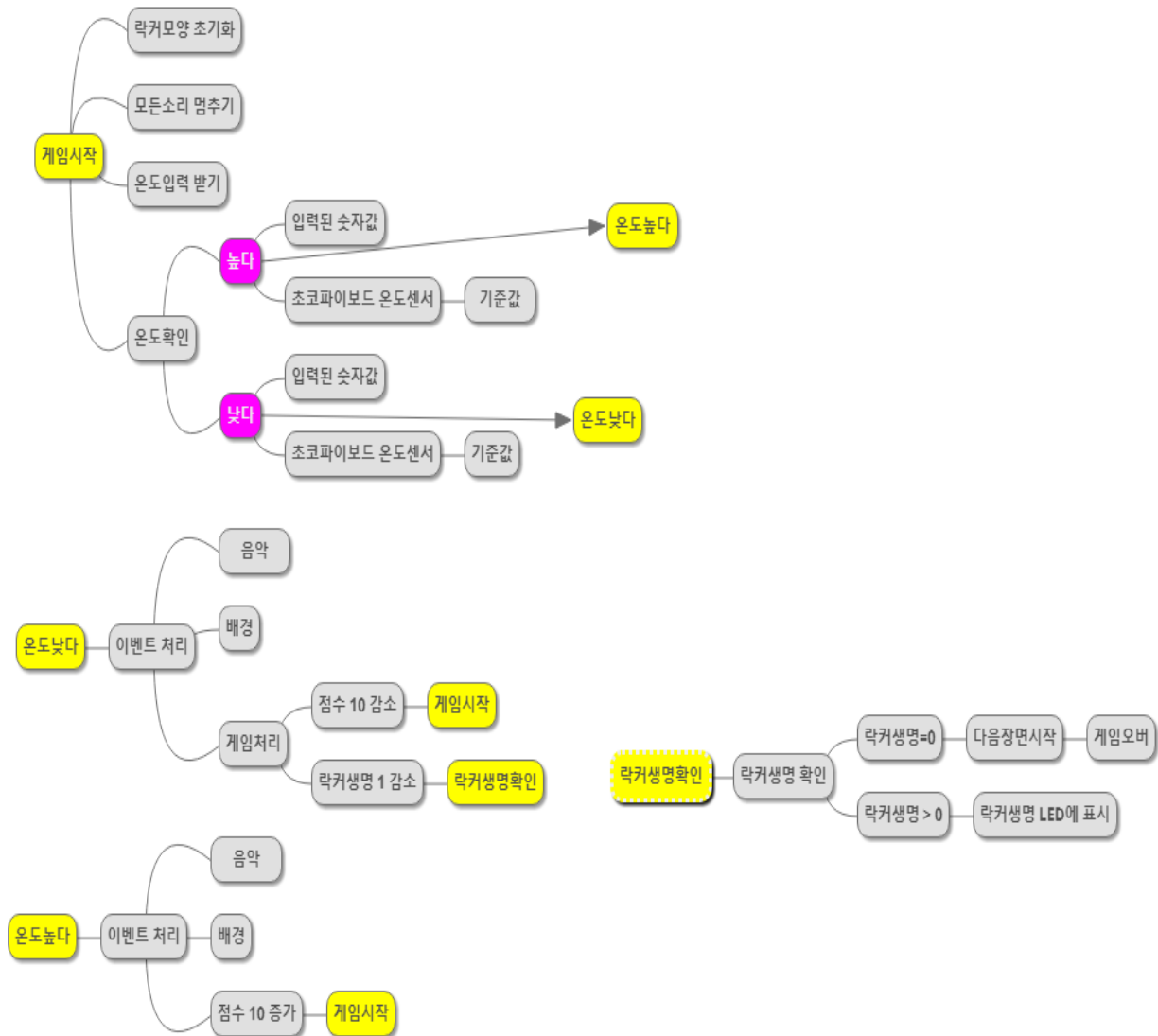
- LED번호의 값을 4로 설정합니다.
- 락커의 생명값이 3(변수 기본값)이기 때문입니다.
- 락커생명값의 변화에 따른 LED 번호 값
 - 최초 락커생명은 3이고 앞 페이지의 온도 낮다 신호를 받으면 1이 줄어 2가 됩니다. 위 코딩 블록 3-2는 1이 됩니다. 즉 1번만 반복하면 됩니다. 따라서 LED번호는 3이됩니다.



락커생명 ▼ 값	3 - 락커생명 ▼ 값 번 반복	LED번호 ▼ 에 -1 만큼 더하기 ?	LED번호 ▼ 값
2	1번 반복	4-1	LED 3 꺼짐
1	2번 반복	4-1-1	LED 2 꺼짐
0	3번 반복	4-1-1-1	LED 1 꺼짐

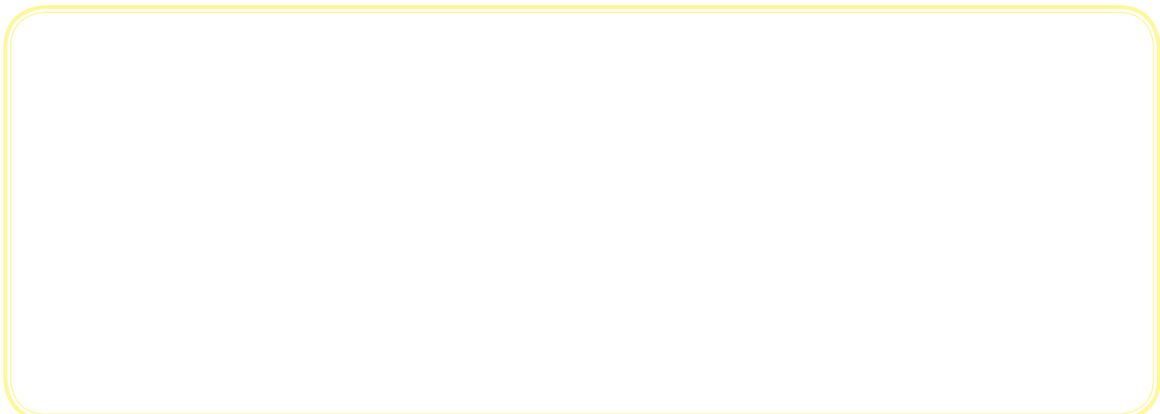
초코파이 보드의 센서 블록을 통해 측정한 현재의 온도보다 낮은 값을 입력할 때마다 LED가 하나씩 꺼져 락커의 생명이 줄어드는 것을 눈으로 확인합니다. LED가 모두 꺼지면 게임이 종료됩니다.

다음은 이제까지 추가로 개선된 이벤트들을 정리한 내용입니다.



[그림 3-8] 온도 맞추기 게임 이벤트 처리 요약

이제 프로그램을 실행하여 개선해야 될 부분을 찾아봅시다.

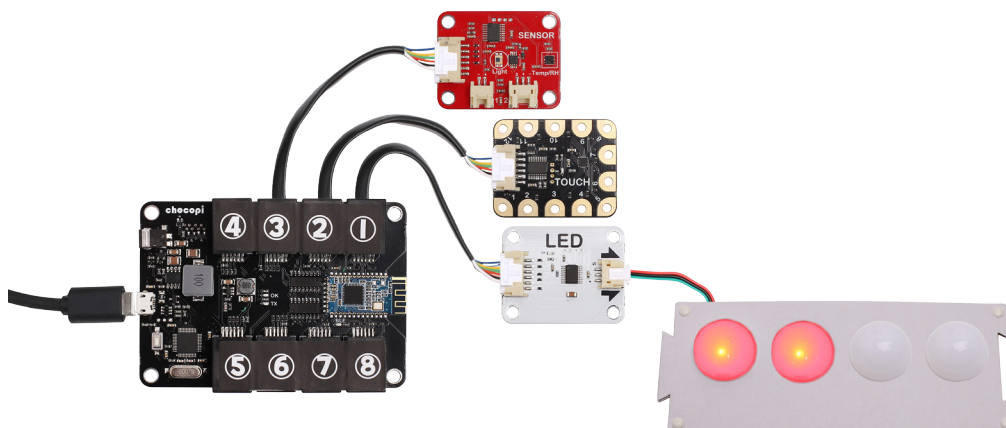
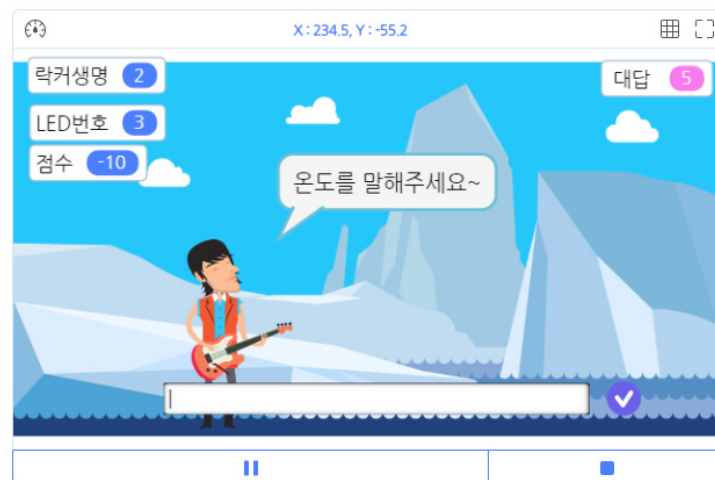




[그림 3-9] 프로그램 시작화면

프로그램을 시작하면 ‘a’키 입력을 기다리고 있습니다. 키를 입력하면 온도를 입력받습니다.

만약 온도를 현재 센서의 온도보다 낮게 답변을 적으면 다음과 같이 락커의 생명이 감소하고 LED 3번째 불빛이 꺼집니다.



그런데 프로그램을 실행해 보면 예상하지 못한 문제들이 나타납니다. 사운드가 재생이 되다가 멈춘다든지, 배경이 바뀌지 않고 그대로 있다든지 하는 문제들은 어떻게 해결하면 될까요?

SUMMARY 써머리~

- 입력되는 정보를 처리하기 위해서는 입력되는 장치에 대한 속성을 확인하고 정보의 종류를 분석해야 합니다.
- 블록과 블록을 연결하여 프로그래밍할 때는 스크립트 상에서 이루어지는 연결 절차를 파악하여 그 순서에 따르는 능력이 중요합니다.
- 게임을 만들기 위해서는 감소, 증가되는 값을 처리하는 변수가 필요합니다.
- 엔트리에서 구현되는 변수의 값을 실제 사물(피지컬컴퓨팅)로 표현하는 능력을 길러야합니다.

이제 좀 더 나은 온도 맞추기 프로그램으로 만들기 위해서 필요한 것들을 오브젝트, 이벤트로 분류해서 적어보시기 바랍니다.

· 오브젝트

· 이벤트



미션을 해결해봅시다!

이전 섹션에서 다루었던 내용을 기초로 하여 자신만의 아이디어를 추가해서 프로그램을 더욱 발전시킬 수 있는 섹션입니다.

MISSION 01 우주를 즐겨

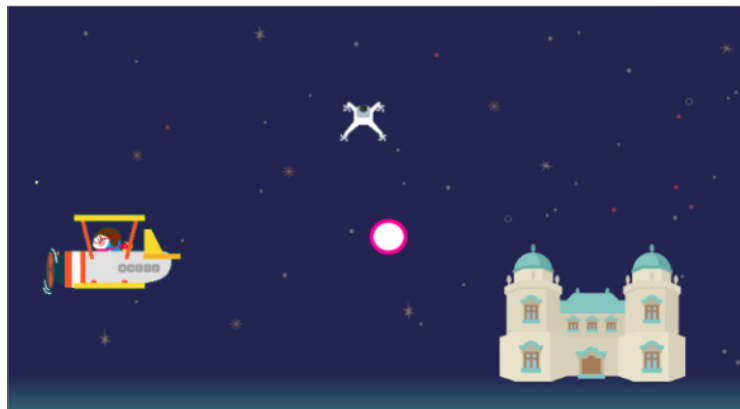
이전 챕터에서 파일럿은 태양을 탐험하기 위해 드론을 띄워서 날리는 경험을 했습니다. 태양의 크기와 질량, 거리에 대해서도 알아보았습니다.

이번엔 수성으로 떠나보도록 하겠습니다.

이제 조금만 가면 수성에 도착하게 됩니다. 500년 전에 지구에서 만들어놓은 건물이 있다고 하는데... 어떻게 변했을까요?



수성에 도착하니 저 멀리 그 건물이 보입니다.



① 이벤트 디자인하기

태양을 탐험할 때 사용하였던 드론을 띄워서 저 건물 속에 들어가 보기로 하였습니다.
이제 여기서 우리가 해결해야 할 문제가 있습니다. 바로 수성에 있는 보물탐험 규칙입니다.

- 파일럿이 수성에 도착하여 드론을 띄워서 수성에 접근하면 다음장면으로 이동한다.
- 보물지역 장면에서 드론의 띄워서 건물에 다가가다
- 건물은 드론이 오게되면 숨어있는 외계인은 빨간, 노랑, 파랑 물약 중 하나를 선택한다.
- 드론은 숨어있는 외계인이 선택한 물약의 번호를 숫자로 입력해서 파일럿에게 신호를 보낸다.
- 파일럿은 드론으로부터 받은 정보를 외계인의 신호와 비교해본다
- 드론이 입력한 숫자(사용자가 입력한 숫자)가 맞으면 점수를 10점 올린다
- 만약 숫자가 3번까지 틀리면 장면이 바뀌면서 'GAME OVER'를 표시하고 끝낸다

② 장면과 오브젝트 추가하기

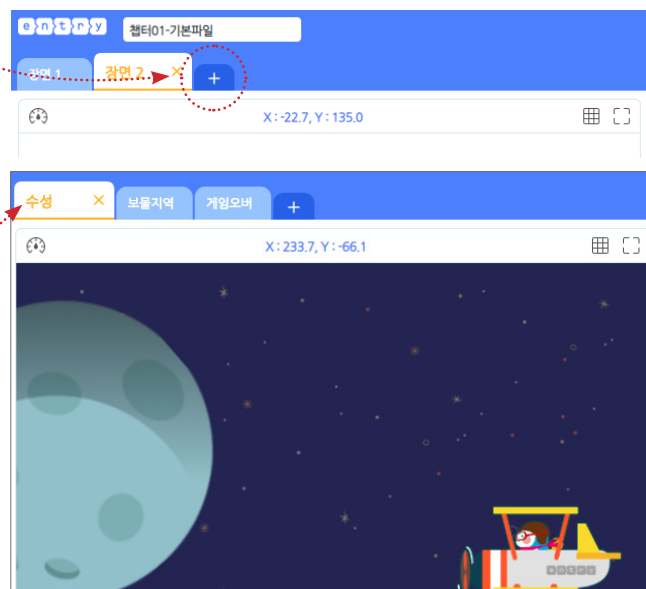
위에서 제시한 문제를 해결할 수 있도록 각각의 장면을 구조화하여 표현해 봅시다.

- | | | |
|------------|------------|-------------|
| · 수성 | · 보물지역 | · 게임오버 |
| - 파일럿 엔트리봇 | - 파일럿 엔트리봇 | - 게임오버 글자 |
| - 드론 | - 건물 | (GAME OVER) |
| - 수성 | - 드론 | |
| | - 전파 | |

장면을 추가하는 방법은 다음과 같습니다.

+ 버튼을 눌러 장면이 추가되면
원하는 이름으로 변경합니다.

수성, 보물지역, 게임오버 장면이
생성되었습니다.



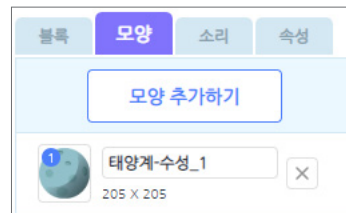


ㄱ. 수성장면


수성장면은 이전 챕터에서 사용하였던 프로그램을 그대로 가져왔습니다.



다만 태양의 모양을 수성으로 바꾸었을 뿐입니다.



그리고 수성과 드론이 닿으면 다음장면으로 이동하기 위해 다음과 같이 블록 프로그래밍합니다.

수성() 오브젝트를 선택하고 코딩합니다.



ㄴ. 보물지역

보물지역은 다음과 같이 오브젝트들을 배치하고 디자인 합니다.



추가해야할 오브젝트는 다음과 같습니다.

‘파일럿 엔트리봇’, ‘건물’을 하나추가하고 ‘외계인2’로 바꿉니다. ‘드론’을 추가합니다.

수성의 배경이 될 ‘별 헤는 밤’을 배경으로 합니다.

다음으로 사용되는 변수들과 신호들은 다음과 같습니다.





ㄷ. 게임오버

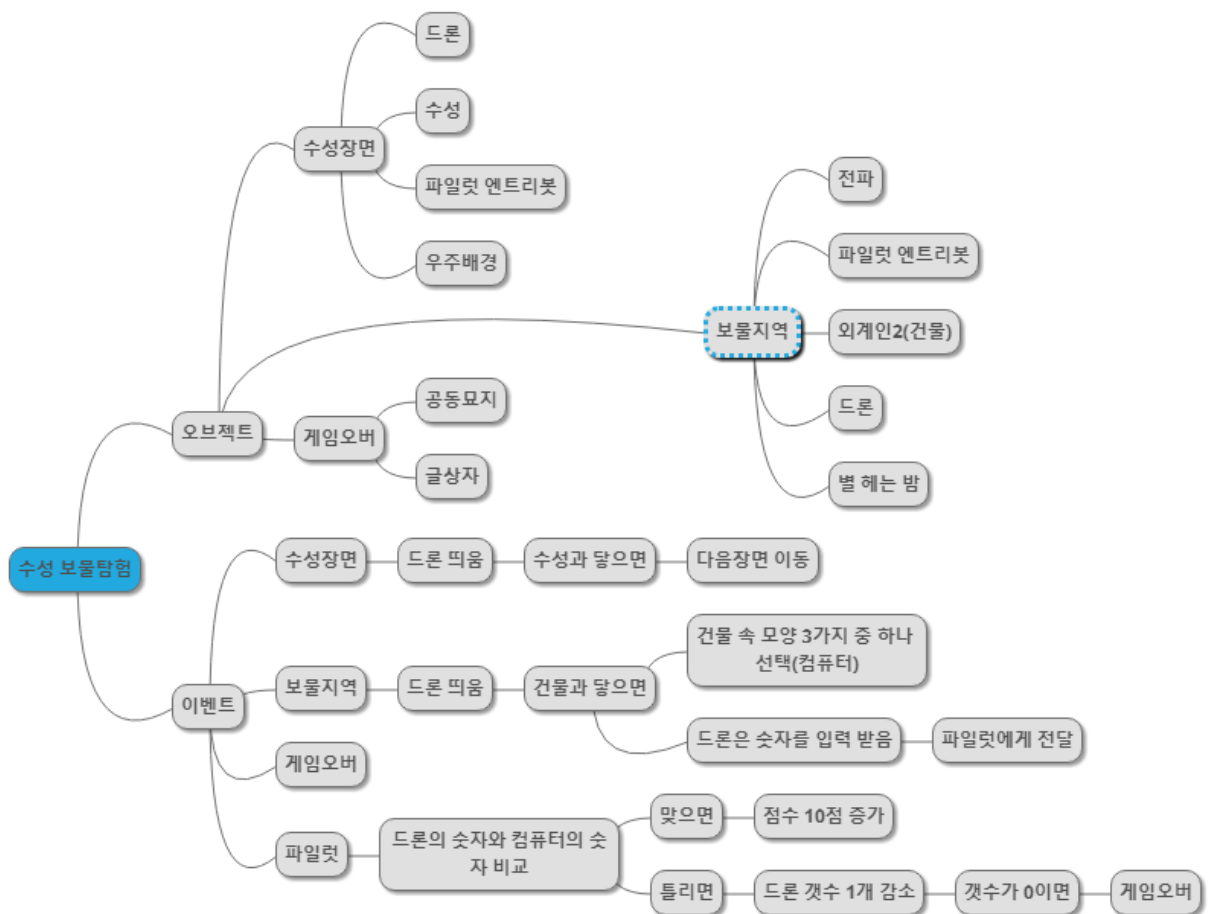
게임오버 장면 디자인과 오브젝트는 다음과 같습니다.



오브젝트는 글상자에서 'GAME OVER'를 입력하고 추가해 줍니다.

③ 오브젝트와 이벤트 연결하기

제시된 문제를 해결하기 위한 알고리즘을 간략하게 정리하면 다음과 같습니다.

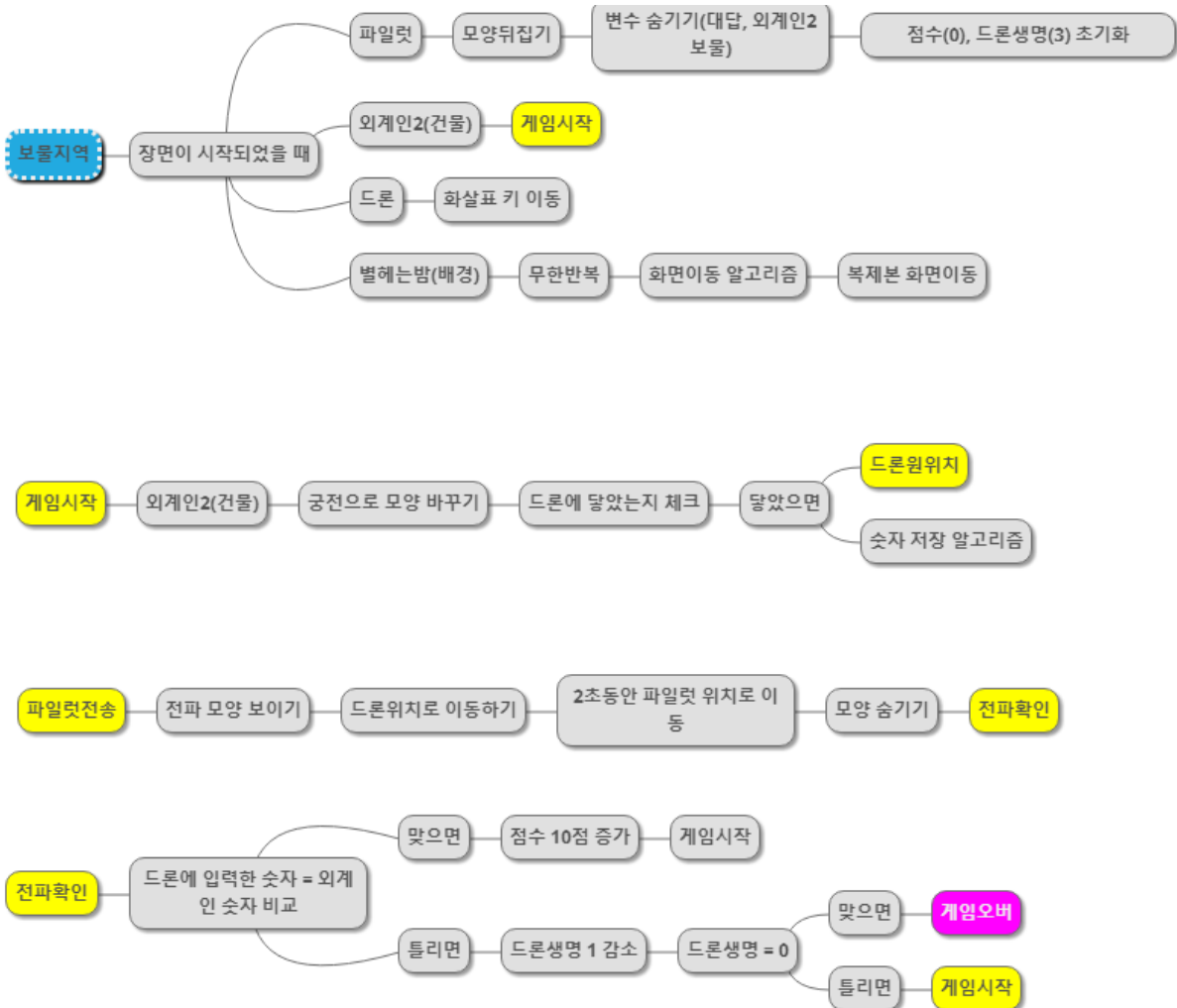


‘수성지역’ 장면에서 진행되는 이벤트 처리는 앞부분에서 설명하였으므로 ‘보물지역’에서 벌어지는 이벤트를 중심으로 설명하겠습니다.

오른쪽 그림과 같이 보물지역에서는 파일럿이 띄운 드론이 건물에 닿으면 외계인(컴퓨터)은 건물 속 모양 3가지 중 하나를 선택하게 됩니다. 드론은 사용자로부터 숫자를 입력받아서 파일럿에게 전달합니다.

파일럿은 드론의 숫자와 컴퓨터의 숫자를 비교해서 맞거나 틀리는 경우에 따라 각각의 이벤트를 처리합니다.

보물지역에서 벌어지는 이벤트를 좀 더 자세하게 풀어서 설명해 보면 다음과 같이 정리할 수 있습니다.





보물지역 장면에서 진행되는 이벤트를 신호(프로시저) 중심으로 설명하면 다음과 같습니다.

- 장면이 시작되었을 때
 - 파일럿 오브젝트는 모양을 뒤집어 표시하고 게임과 관련된 변수를 숨깁니다.
그리고 게임을 초기화합니다.
 - 외계인2(건물)에서는 게임시작 신호를 전달합니다.
 - 드론은 화살표 키 등으로 이동하도록 초기화합니다.
 - 배경의 경우 별 헤는 밤이 왼쪽에서 오른쪽으로 움직이도록 화면이동 처리를 진행합니다.
- 게임시작 신호를 받았을 때
 - 외계인2(건물)은 궁전모양으로 위장합니다. 드론이 닿았는지 확인하여 닿았으면
'드론원위치' 신호를 보내어 드론을 원위치 시키고, 컴퓨터에서 숫자를 발생시키는
알고리즘으로 숫자를 저장합니다.
- 파일럿 전송 신호를 받았을 때
 - 전파 모양을 보이게 하여 드론에서 파일럿으로 전파가 전송되는 것을 표시합니다.
즉 드론위치로 전파 오브젝트를 이동시킨 다음 2초동안 파일럿 위치로 이동합니다.
그리고 전파 오브젝트를 숨기기 합니다.
 - 전파확인 신호를 보냅니다.
- 전파확인 신호를 받았을 때
 - 보물지역의 핵심 알고리즘이라고 할 수 있습니다.
 - 드론에 입력한 숫자와 외계인(컴퓨터)이 생성한 숫자를 비교하여 맞으면 점수를
10 증가시킵니다. 만약 틀리면 드론의 생명을 하나 감소시키고 드론의 생명이 0이면
게임오버 장면으로 이동하고 종료합니다.

이제 오브젝트 중심으로 실제 블록 프로그래밍하는 방법을 설명하겠습니다.

· 파일럿 엔트리봇

파일럿 엔트리봇이 처리하는 신호는 2가지입니다. 장면이 시작되었을 때와 전파확인 신호를 받았을 때입니다.

전파확인 신호를 받았을 때는 드론에서 입력한 숫자를 가지고 외계인(컴퓨터)이 생성한 숫자를 비교하여 다음과 같이 처리합니다.





· 외계인2(건물)

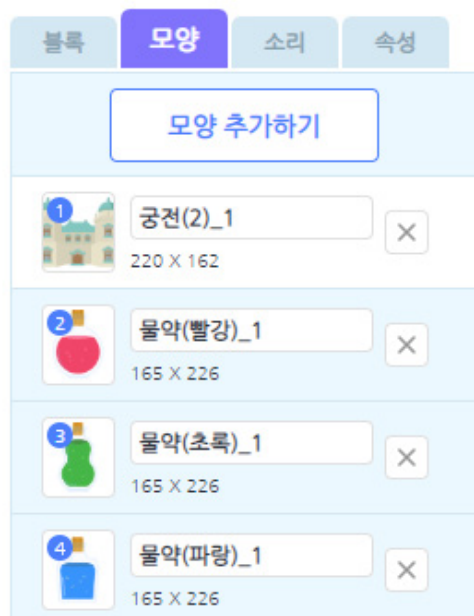


외계인이 숨어있는 건물에서는 먼저 건물의 모양으로 위장하고 드론에 닿았는지를 계속 확인합니다. 외계인2에 드론이 닿으면 ‘드론원위치’ 신호를 보냅니다.



숫자를 계산하고 있다는 것을 보여 주기 위해서 다음과 같이 2와 4사이의 무작위 수를 처리하고 건물의 모양을 계속해서 바꾸어 줍니다.

위의 건물모양을 계속 바꾸기 위해서는 아래 그림과 같이 외계인2(건물) 오브젝트의 모양 탭에 추가되어 있어야합니다.

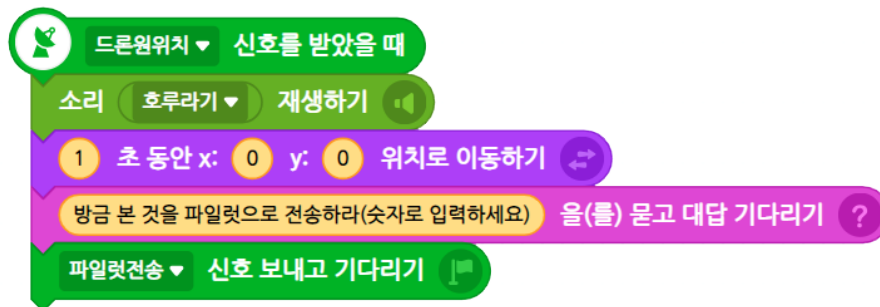


다음으로 '외계인2 보물 변수'에 또다시 1부터 4사이의 난수값을 저장해서 실제 보여지는 모양에 따라 숫자를 추측할 수 없도록 한 번 더 난수를 발생시킵니다.

물론 장면이 시작되면 '게임시작' 신호를 보내어 게임이 시작을 위한 변수들을 초기화 합니다.

· 드론

드론의 경우는 크게 '드론원위치' 신호를 받았을 때와 드론을 이동시키는 키입력 블록들로 구성됩니다.



'드론원위치'는 외계인2(건물)에서 '드론'이 닿았을 때 보내는 신호를 받아 처리하는 것으로 '호루라기'소리를 재생시키고 원래의 위치로 돌아오게 합니다.

그리고 파일럿에게 드론이 수집한 정보를 입력할 수 있도록 '숫자'를 입력받습니다.

'파일럿전송' 신호를 보내고 파일럿에서 처리할 때까지 기다립니다.

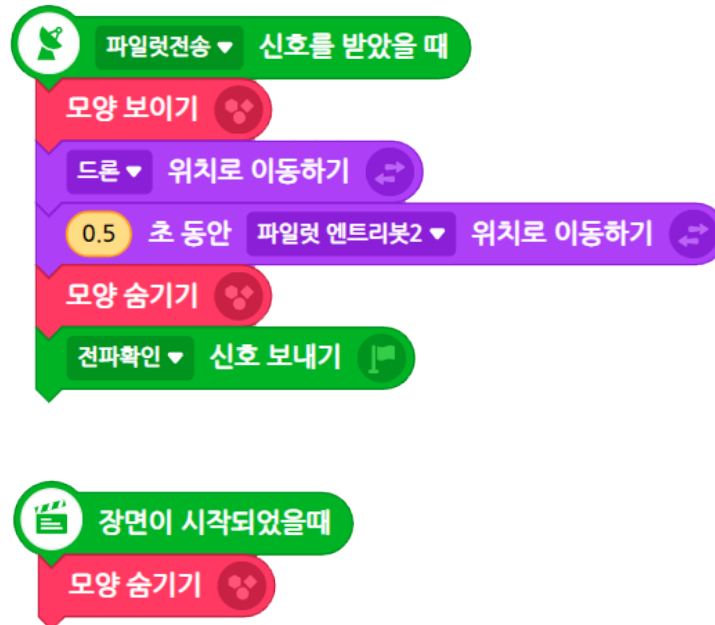
다음의 그림은 드론의 이동을 위해서 키 입력을 처리하는 것입니다.





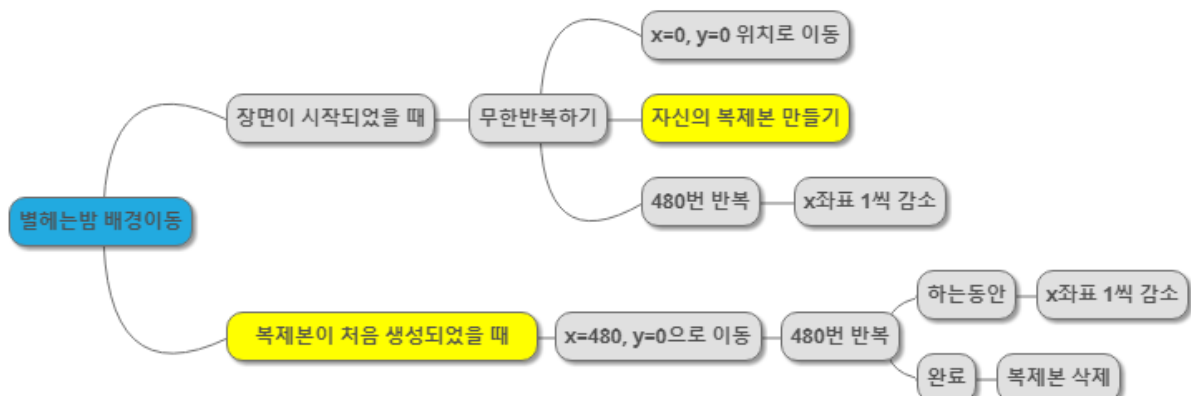
· 전파 ○

전파 오브젝트의 경우 드론에서 파일럿전송 신호를 보낼 때 실제 전파가 전해지는 느낌을 줄 수 있도록 애니메이션 처리하는 부분입니다.

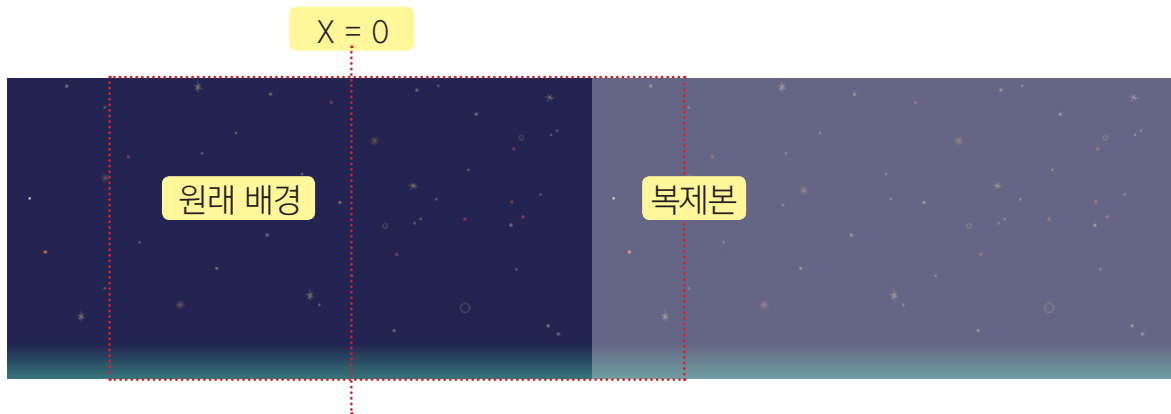


· 별헤는밤(배경)

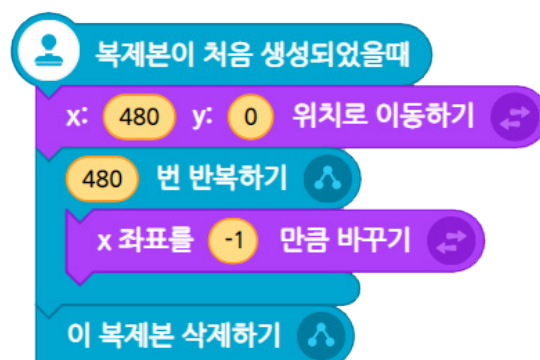
실제 수성이라는 행성이 빠르게 움직이고 있다는 것을 나타내기 위해서 다음과 같은 알고리즘을 적용하였습니다.



실제 배경을 $x=0$, $y=0$ 을 기준으로 왼쪽으로 이동시킵니다. 화면 전체의 가로길이는 480이므로 480번 반복해서 x좌표를 1씩 감소시키면 화면이 왼쪽으로 이동하게 됩니다.

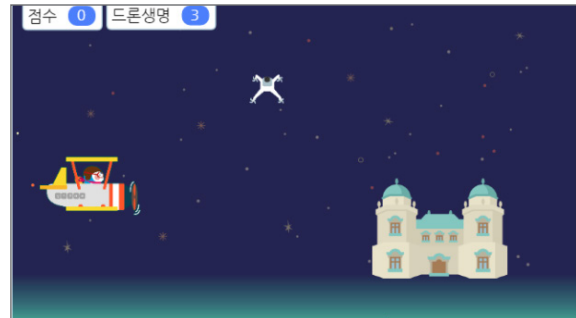
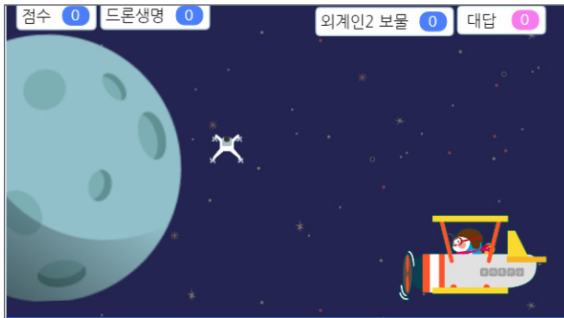


따라서 다음과 같은 블록 프로그래밍으로 처리합니다.





④ 프로그램 테스트와 디버깅



- ▲ 시작 버튼을 클릭하여 프로그램을 실행합니다. 드론을 보이게 하고 수성행성으로 접근합니다. 다음 장면으로 전환됩니다.
- ▲ 건물에 드론이 닿도록 가까이 갑니다.



- ▲ 건물에 드론이 닿으면 건물의 모양이 여러가지로 바뀌면서 숫자를 입력받습니다.
- ▲ 입력받은 숫자가 틀리면 다음과 같은 결과가 나타납니다.



- ◀ 만약 드론생명이 0이 되면 '게임오버' 장면으로 이동하고 게임이 종료됩니다.

직접 코딩한 게임을 해보고 더 수정하거나 개선해야 할 점이 무엇인지 적어 보세요.
입력받은 정보를 처리하는 과정이나 알고리즘을 개선할 필요가 있는 것들을 생각해 봅니다.

MISSION 02 업그레이드 드론

수성의 온도와 습도를 알 수 있는 프로그램을 만들어 봅시다.
이를 통해서 3단계의 색깔로 LED를 표현할 수 있도록 추가해 봅시다.

또한 이것을 게임으로 만들기 위해서는 어떤 것이 필요할까요?
아래의 아이디어를 보고 여러분만의 프로그램을 만들어 주시기 바랍니다.

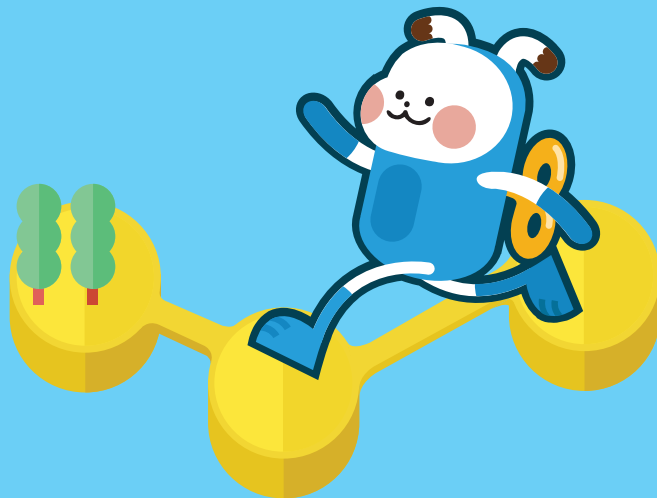
- 온도와 습도의 값에 따라 배경이 움직이는 속도를 정하자
- 건물을 추가하여 2개의 숫자를 비교할 수 있도록 해 보자

이제 상상은 여러분의 몫입니다.

04

잃어버린 기억

조이스틱 다루기





문제가 무엇인가요?

이번 챕터에서는 두 개 이상의 오브젝트가 초코파이보드의 조이스틱과 연결되어 어떻게 상호작용하게 되는지 알아보도록 하겠습니다.

계속해서 학습하고 있지만 문제해결을 위해 생성한 이벤트는 결국 오브젝트의 속성을 활용하여 서로 연결하는 것이 프로그래밍의 핵심과정입니다.

여기 간단한 동전 따먹기 게임을 만들어 봅시다.

- 하늘에서 동전이 떨어진다.
- 아래쪽에서 로켓이 좌우로 움직여서 떨어지는 동전을 따 먹는다.
- 다만 로켓은 초코파이보드의 조이스틱버튼을 활용하여 움직일 수 있다.

그렇다면 엔트리 프로그램에서 초코파이보드의 조이스틱 값을 입력받기 위해서는 어떤 블록을 사용해야 정보를 수집할 수 있을까요?

초코파이보드의 조이스틱 컨트롤은 좌우와 상하 값을 제공합니다.

포트1 ▼ 컨트롤 조이스틱 좌우 ▼ 의 값

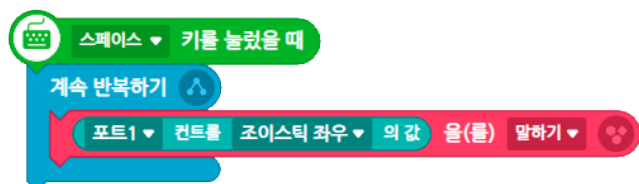
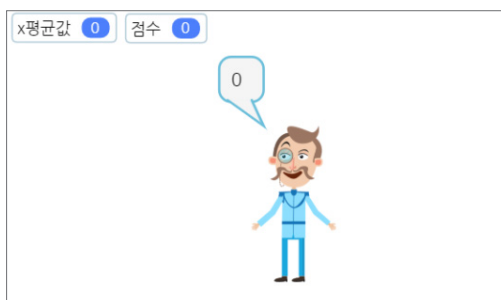
조이스틱 좌우

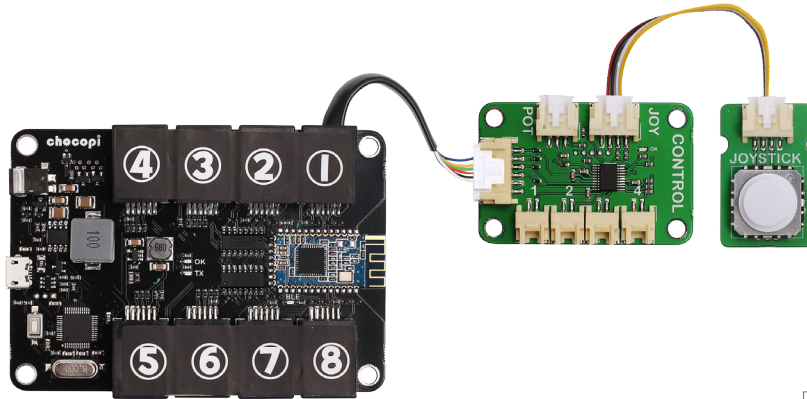
조이스틱 상하

볼륨

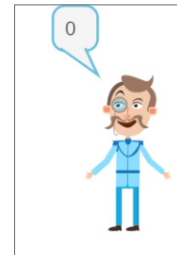
조이스틱에서 제공하는 좌표값의 범위는 다음과 같은 프로그래밍을 통해 확인할 수 있습니다.

먼저 괴짜박사 오브젝트를 하나 추가하고 스페이스 키를 누르면 조이스틱버튼의 x좌표값을 반복해서 말하도록 프로그래밍합니다.

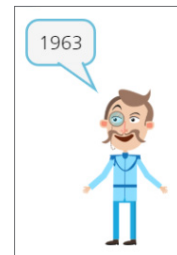




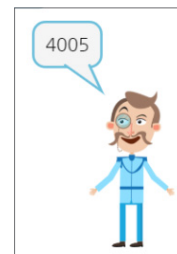
초코파이보드 조이스틱 왼쪽 끝으로 옮겼을 때(x좌표 최소값)
x좌표의 최소값은 '0' 근처에서 조금씩 변합니다.



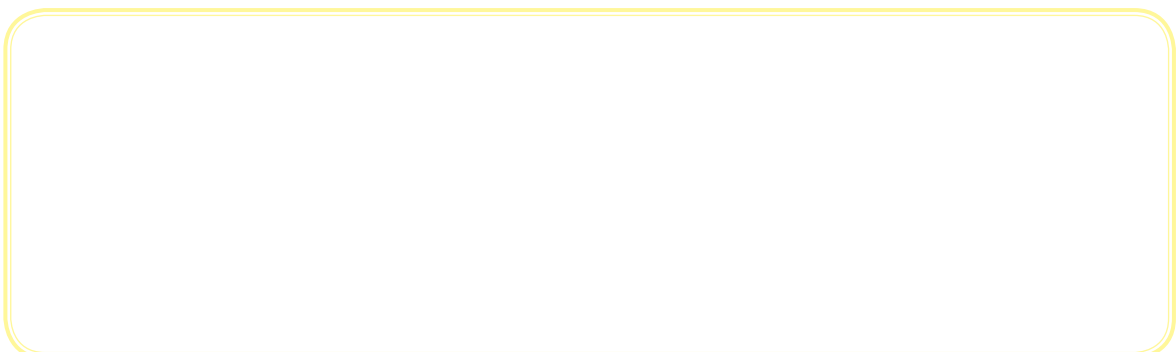
이번에는 조이스틱을 그대로 둔 채 값을 측정해 봅니다.
1962~1964 범위에서 값들이 변합니다.



마지막으로 조이스틱을 오른쪽 끝까지 밀어서
최대값을 측정해 봅시다.
x좌표의 값이 '4005'으로 나타나는데 값들이 조금씩
변하는 것을 알 수 있습니다.
마찬가지로 y좌표의 값들도 같은 방법으로 측정해 봅시다.



여기서, 여러분이 고민해야 할 부분이 있습니다. 어떠한 부분을 중간 값으로 해야 할까요?
중간값을 어떤 식으로 설정하면 좋을지 한 번 생각해 봅시다.





문제를 분해해봅시다!

이제 동전 따먹기 게임을 만들기 위해서 문제에서 제시된 요소들을 가지고 프로그래밍하기 위해서 필요한 것들을 한 번 적어보시기 바랍니다.

제시된 문제를 분석하여 추출한 요소들

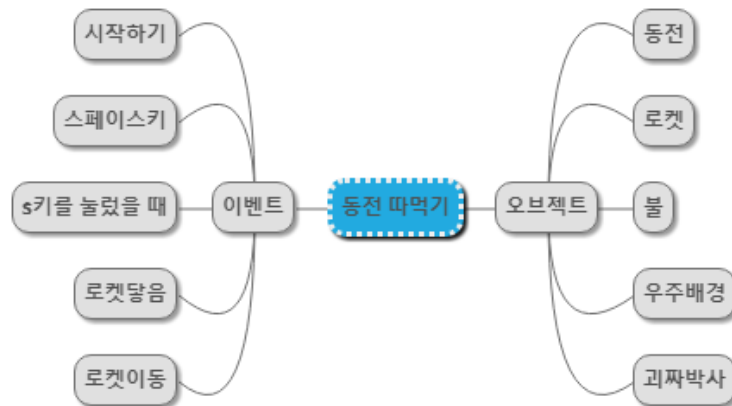
- 하늘에서 동전이 떨어진다.
 - 동전은 처음 어디에 위치할 것인가?
 - 어떻게 떨어지게 할 것인가?
 - 어디까지 떨어지게 할 것인가?
 - 떨어지고 나면 어떻게 할 것인가?
- 아래쪽에서 로켓이 좌우로 움직여서 떨어지는 동전을 따 먹는다.
 - 게임이 시작될 때 로켓은 어디에 위치할 것인가?
 - 어떻게 로켓을 움직일까?
 - 로켓과 동전은 어떤 식으로 상호작용할 것인가?
 - 동전을 따먹게 되면 어떠한 변화를 줄 것인가?
- 다만 로켓은 초코파이보드의 조이스틱버튼을 활용하여 움직일 수 있다.
 - 조이스틱버튼의 어떤 정보를 가지고 로켓을 움직일 것인가?

여러분이 더 생각하고 싶은 부분이 있다면 추가로 적어봅니다.



이벤트 다루기

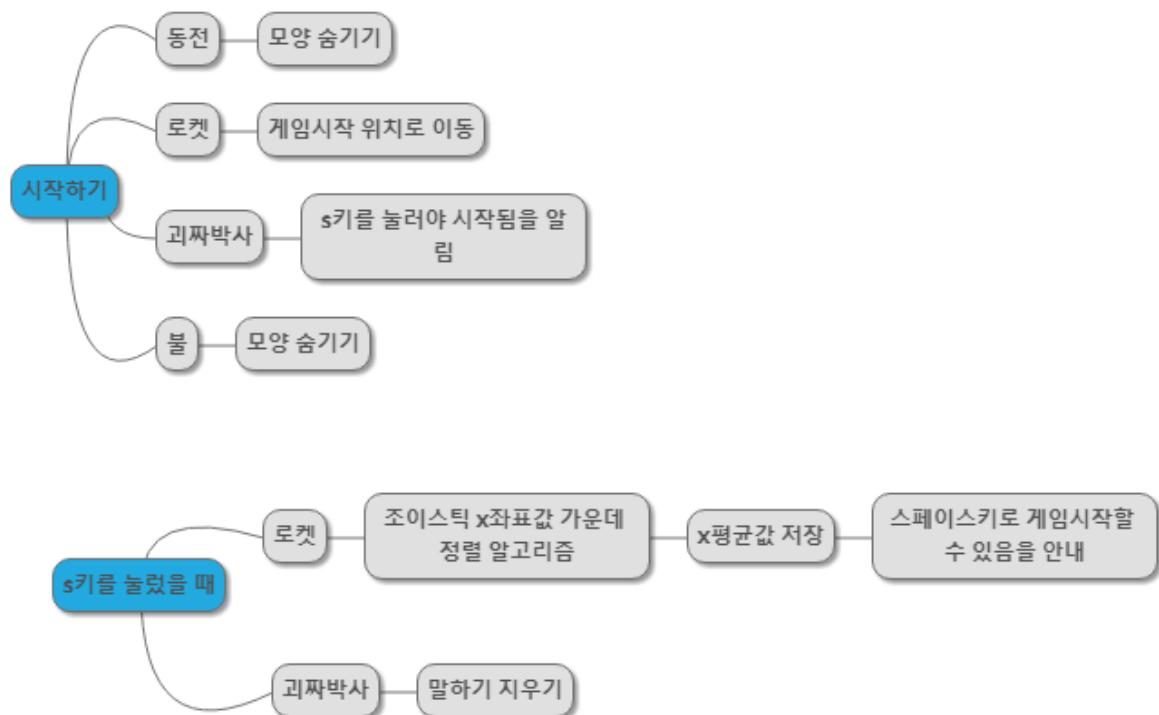
그렇다면 분석한 요소를 기반으로 어떻게 이벤트들을 작성하는지 알아보니다. 이전 챕터들과는 달리 필요한 오브젝트들을 생각해보고 실제 발생할 수 있는 이벤트들을 정리해 보았습니다.

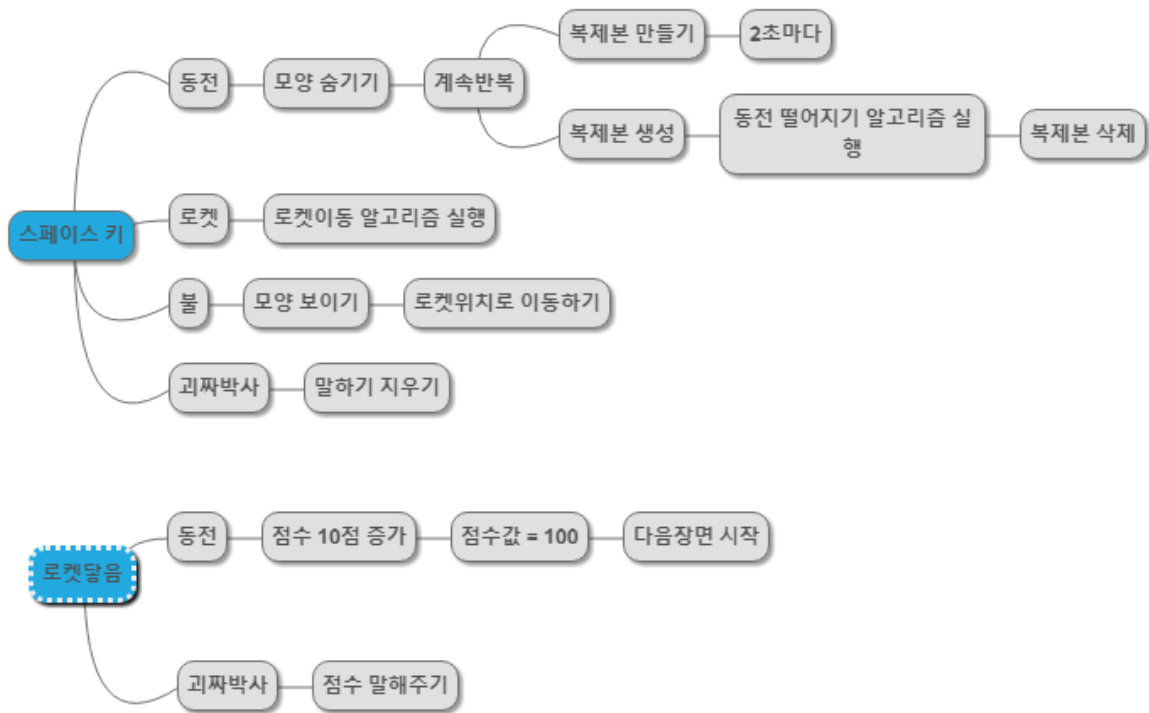


[그림 4-1] 이벤트 요약

스크립트로 표현하기

분석하여 추출된 요소를 시작버튼을 클릭하면, 's'키를 눌렀을 때, 스페이스 키를 눌렀을 때, 로켓달음 신호를 받았을 때의 순서대로 진행되는 스크립트로 변환하여 시각적으로 표시해 보면 다음과 같습니다.





[그림 4-2] 이벤트 스크립트 요약 정리

절차대로 스크립트를 표현하면 다음과 같습니다.

- 시작버튼을 클릭하면
 - 오브젝트에 따라 보일 것과 숨길 것이 있습니다. 그리고 초기화를 위해서 로켓을 특정 위치로 옮겨야 합니다.
 - 점수 변수값을 초기화합니다.
- s키가 입력되면
 - 움직이지 않을 때의 **포트1 ▼ 컨트롤 조이스틱 좌우 ▼ 의 값** 을 10회 정도 반복해서 측정하여 x좌표의 평균값을 만듭니다.
 - 측정된 평균값은 x좌표 평균값 변수에 저장합니다.
- 스페이스키를 눌렀을 때
 - 동전 오브젝트의 경우

게임이 시작되면 먼저 동전 오브젝트는 모양을 숨겼다가 화면 위쪽에서 나타나게 합니다. 이때 동전 오브젝트가 만들어질 때 마다 y좌표는 그대로 두고 x좌표는 바꾸어서 다른 위치에서 오브젝트가 나타나도록 하여 게임의 흥미를 높여 줍니다.



복제본이 생성되면 복제본의 y좌표 값을 감소시켜서 실제로 동전이 떨어지는 효과를 냅니다.

동전이 나타나는 시간은 2초간격으로 설정합니다.

벽에 닿는 순간 동전이 떨어졌다는 것을 의미하므로 생성된 복제본을 삭제합니다.

만약 떨어지는 동안 로켓에 닿게 되면 ‘로켓닿음’ 신호(프로시저)를 호출하여 점수를 처리하는 알고리즘을 실행합니다.

- 로켓 오브젝트의 경우

초코파이보드의 조이스틱으로 x좌표 값을 변화시켜 로켓을 좌우로 이동합니다. 이 때 x좌표의 가운데 평균값을 기준으로 로켓의 좌우 이동방향을 결정합니다.

- 불 오브젝트의 경우

게임에 시각적 효과를 주기 위해서 불 오브젝트는 로켓위치로 이동시켜 항상 로켓 아래에 위치하도록 합니다. 이를 위해서는 중심점을 아래로 이동해야 합니다.

- 괴짜박사 오브젝트의 경우

괴짜박사는 이전에 말한 내용을 지우기 위해서 ‘말하기 지우기’ 블록을 추가합니다.

· 로켓닿음 신호(프로시저)를 받았을 때

- 동전 오브젝트의 경우

점수를 10점씩 증가시키고, 수가 100점이면 다음 장면으로 전환하여 게임을 종료

- 괴짜박사 오브젝트의 경우

획득한 동전 점수를 말합니다.

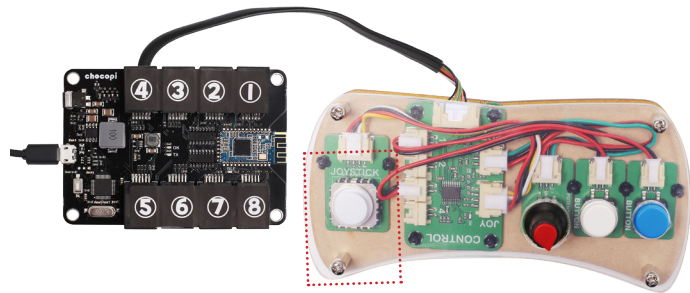
오브젝트 블록 프로그래밍

이제 생각한 스크립트(알고리즘)를 차례대로 블록 프로그래밍해 봅니다.

게임장면의 오브젝트를 오른쪽 그림과 같이 배치합니다.

초코파이보드의 컨트롤 블록을 4번 포트에 꽂고 엔트리와 연결해 줍니다.(조립방법은 초코파이보드 홈페이지 참고)


[그림 4-3]에서 왼쪽에 위치한 흰색 조이스틱이 우리가 사용할 것입니다.



[그림 4-3] 초코파이보드 컨트롤 블록 연결


순서대로 실제 스크립트에 제시된 이벤트 별로 실제 블록 프로그래밍을 실시해 봅시다.

시작하기 버튼을 눌렀을 때

· 동전 오브젝트 


시작하기 버튼을 눌렀을 때 모양을 숨깁니다.

▶ 시작하기 버튼을 클릭했을 때
모양 숨기기

· 로켓 오브젝트 


시작하기 버튼을 눌렀을 때
화면의 중앙(x좌표 0)으로 이동시켜줍니다.

▶ 시작하기 버튼을 클릭했을 때
x: 0 y: -112 위치로 이동하기

· 괴짜박사 오브젝트 

괴짜박사는 오브젝트의 초기화가
완료되었으므로 's'키를 눌러 로켓의
x좌표를 정렬시켜줘'라고 안내해야 합니다.

▶ 시작하기 버튼을 클릭했을 때
s 키를 눌러 로켓의 x좌표를 정렬 시켜줘 **을(를)** 말하기

· 불 오브젝트 

불 오브젝트는 게임이 시작될 때 로켓 아래에서
나타나도록 시작버튼을 클릭하면 모양을 숨겨 줍니다.

▶ 시작하기 버튼을 클릭했을 때
모양 숨기기



s키를 눌렀을 때

· 로켓 오브젝트



s키를 누르면 로켓을 좌우로 이동시키기 위해 조이스틱을 그대로 둔 채로 약간씩 변하는 좌우값을 평균화시킬 알고리즘을 만듭니다. 즉 기준점을 만들어 주는 것입니다.

이를 위해서 x평균값 변수가 필요합니다. 실제 블록 프로그래밍은 다음과 같이 진행 합니다.



사용자에게 조이스틱의 x좌표를 정렬중이라고 안내합니다.

그리고 x평균값을 0으로 정합니다.

다음으로 x평균값에 입력되는 초코파이보드의 조이스틱 좌우 값을 10번 반복하여 더합니다.

마지막으로 앞의 10번 입력하여 더한 값을 10으로 나누어 x평균값을 정합니다.

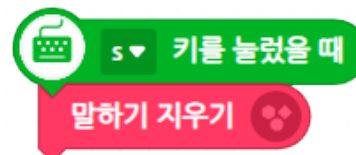
사용자가 알 수 있도록 0.5초 기다리고 x평균값의 정렬이 완료되었음을 안내하고 '스페이스 키'를 누르면 게임을 시작할 수 있다는 것을 안내합니다.

· 괴짜박사 오브젝트



괴짜박사는 점수를 말하는 역할도 있습니다.

따라서 s키를 누르면 말한 내용을 지워줍니다.



이것으로 기초적인 이벤트들인 시작하기, s키를 눌렀을 때 이벤트들에 대한 블록 프로그래밍을 완료하였습니다.



하나씩 연결해봅시다!

이제 스페이스키를 눌렀을 때, 동전이 로켓에 닿으면 오브젝트들이 어떻게 상호작용하는지 자세히 살펴보도록 합시다.

오브젝트간 상호작용

우리는 제시된 문제를 해결하기 위해 이벤트에 어떤 요소가 필요한지 분석하고 이러한 요소들을 스크립트, 즉 알고리즘으로 만드는 과정을 학습하고 있습니다.

실제 프로그래밍 과정에서는 창의적인 아이디어가 떠오르면 그것을 구조화할 수 있는 능력이 중요합니다.

여러분은 어떤 아이디어가 떠오르나요?

떠오른 아이디어를 어떻게 자신만의 정보로 기록하나요?

요즘의 소프트웨어 중심 사회를 많은 사람들은 초연결사회라고 합니다. 모든 것이 연결된 세상 속에서 살고 있다는 의미가 큼니다. 예전에는 나만이 알고 있는 정보(지식)가 중요했다면 지금은 모두가 공유한 정보 속에서 내가 원하는 것을 수집, 분류, 가공하는 능력이 더 중요해 졌습니다. 왜냐하면 구글이나 유튜브 등의 사이트에서 키워드를 가지고 검색해보면 어렵지 않게 수많은 정보를 얻을 수 있기 때문입니다.

그렇다면 이러한 수많은 정보를 내가 원하는 '아이디어'로 전환하기 위해서 필요한 것은 무엇일까요?

필요한 정보는 앞에서 제시한 것처럼 구글, 유튜브의 검색을 통해서 얻는다면 수집된 수많은 정보를 어떻게 하면 효과적으로 분류하고 정리할 수 있을까요?

제가 제시할 수 있는 해답은 '정보를 시각화'하라는 것입니다.

정보를 시각화하는 연습을 통해 체계적이고 구조적으로 사고하는 능력을 키울 수 있습니다. 그 대표적인 방법으로는 마인드맵을 활용하여 수집한 정보를 시각화하는 것입니다.

또한 가지고 있는 스마트기기를 적극적으로 활용하는 습관을 가져보시기 바랍니다. 스마트기기에는 수없이 많은 메모앱들이 있습니다. 가능하면 단순한 텍스트만 저장할 수 있는 앱보다 링크 형태로 저장이 가능한 앱을 사용하시기 바랍니다.



MindMup 2

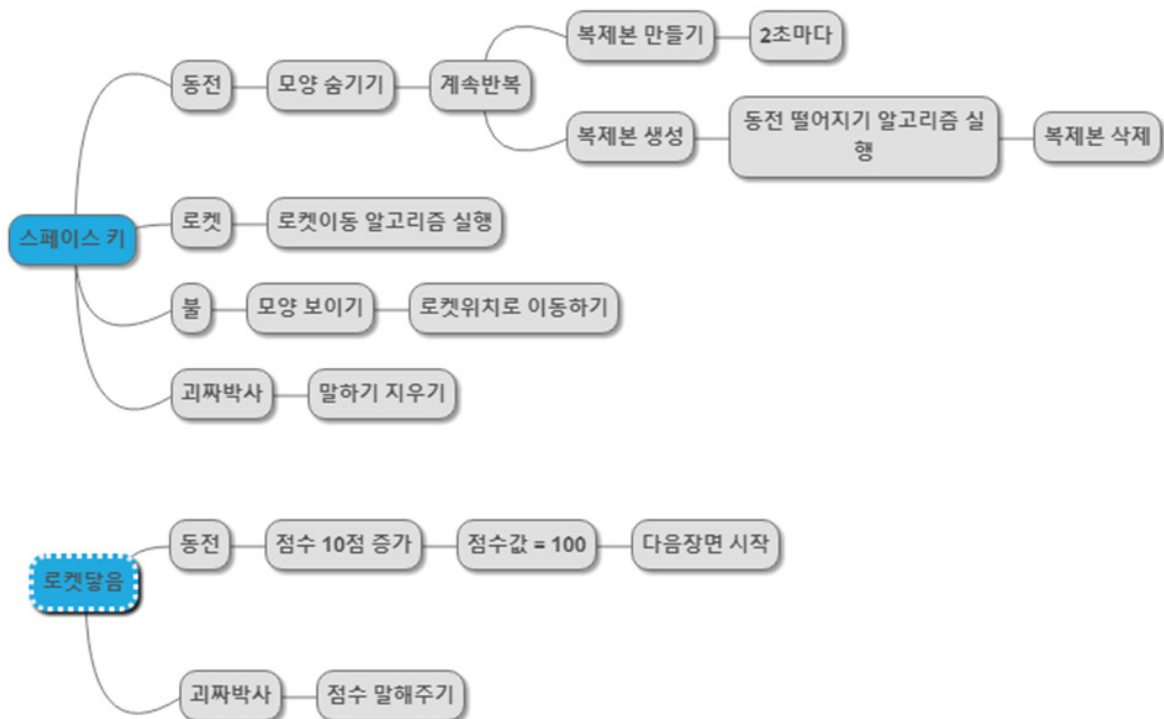
Zero-friction mind mapping, tightly integrated with Google Apps



[그림 4-4] 온라인 마인드맵 MindMup 2(출처: mindmup.com)

이제 본론으로 돌아가서 ‘스페이스 키’를 눌렀을 때 이벤트와 ‘로켓달음’ 이벤트를 처리하는 블록 프로그래밍을 작성하여 오브젝트간에 상호작용이 어떻게 이루어지는지 살펴봅시다.

설정된 문제를 요약한 스크립트를 통해 다시 한 번 살펴보겠습니다.



스페이스 키를 눌렀을 때

① 동전 오브젝트

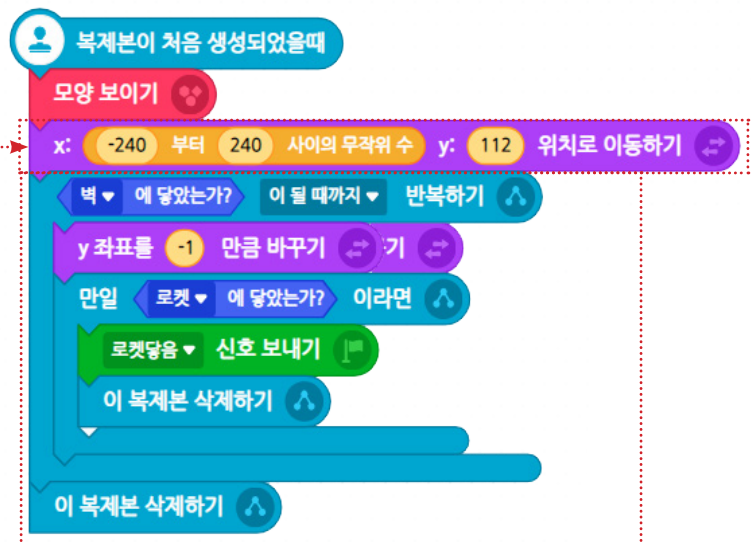
스페이스 키를 누르면 게임을 시작합니다.

동전 오브젝트와 관련된 블록 프로그래밍 소스는 오른쪽 그림과 같습니다.

일단 동전 오브젝트를 숨깁니다. 다음으로 동전이 계속 떨어지기 위해서 '동전' 자신의 복제본을 만듭니다. 2초가 지나면 동전 복제본을 만드는 것을 반복합니다.



동전 복제본이 만들어졌을 때 실제 동전 오브젝트를 보여줍니다. 동전이 가로범위에서 무작위로 나타날 수 있도록 x좌표의 범위내(-240~240)에서 좌표값을 가지도록 처리합니다. y좌표는 특정 위치에 고정했습니다.



실제 동전이 위에서 아래로 떨어지는 것을 구현하는 알고리즘입니다.

동전은 벽에 닿을때까지 아래의 과정을 반복해서 처리합니다.

- y좌표를 -1만큼 바꾸면 동전 오브젝트의 y좌표값이 하나씩 감소하여 동전이 아래로 내려오는 것처럼 보입니다.
- 단, 동전이 내려오면서 '로켓'에 닿을 수 있기 때문에 '로켓에 닿았는가?'를 확인합니다.
- 동전이 로켓에 닿으면 '로켓닿음' 신호(프로시저)를 호출하고 동전이 사라지도록 '이 복제본 삭제하기' 블록을 추가합니다.
- 로켓에 닿지 않더라도 벽에 닿게되면 복제본을 삭제하여 동전이 땅에 떨어졌다는 것을 시각적으로 표현합니다.



여기에 한가지 고민이 있습니다. '벽에 닿는다'와 관련된 것인데 동전 오브젝트가 벽에 닿는 경우를 생각나는대로 적어봅시다.

동전이 벽에 닿는다는 의미는 배경사이즈(x축 -240부터 240, y축 -135부터 135)를 벗어남을 의미합니다. 한 방향의 벽에만 닿는다는 것은 있을 수 없습니다. 따라서 보다 정확한 프로그래밍을 위해서는 다음의 두 가지 경우를 생각해야 합니다.

- 4면의 벽에 닿을 경우 또는 x,y좌표값을 통해 벽에 닿는 경우 확인
- 동전 자체의 x,y좌표값으로 벽에 닿았는지 확인

② 로켓 오브젝트

이번에는 로켓 오브젝트와 관련된 부분을 블록 프로그래밍합니다.

로켓 오브젝트는 실제 초코파이보드 조이스틱의 좌우값을 이용하여 조정합니다.



로켓의 움직임을 계속 확인해야 하기 때문에 다음 사항을 반복실행합니다.

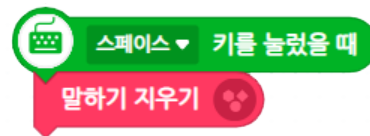
- 초코파이보드 4번 포트에 연결된 조이스틱의 좌우값이 's'키를 눌렀을 때 구했던 x평균값보다 작다면 조이스틱 레버가 왼쪽으로 가있다고 해석할 수 있습니다. 따라서 x좌표를 -2만큼 바꿔서 로켓을 왼쪽으로 이동시킵니다.
- 만약 x평균값보다 크다면 조이스틱 레버가 오른쪽으로 가있다는 뜻이므로 x좌표를 2만큼 바꿔서 로켓을 오른쪽으로 이동시킵니다.

다음으로 괴짜박사, 불 오브젝트와 관련된 블록 프로그래밍을 살펴보겠습니다.

③ 괴짜박사 오브젝트



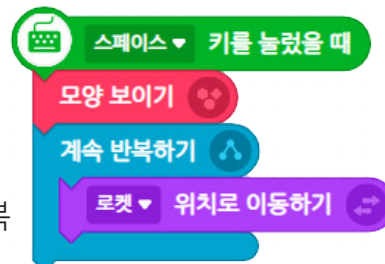
괴짜박사 오브젝트의 프로그래밍은 생각보다 간단한 작업입니다. 스페이스 키를 눌렀을 때(게임이 시작되면) 괴짜박사 오브젝트가 말하고 있다면 그 말을 지우는 작업만 추가하면 됩니다.



④ 불 오브젝트

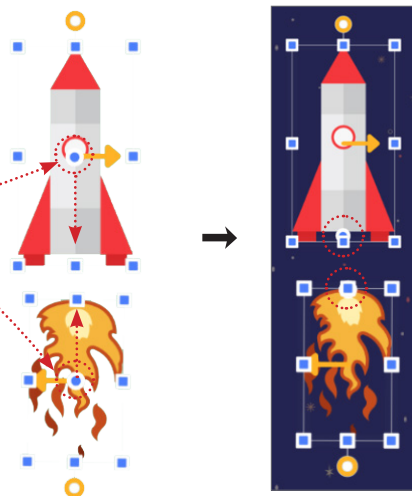


스페이스 키를 누르면 로켓이 점화되는 것처럼 보이도록 '불' 오브젝트를 '로켓'오브젝트로 위치를 이동시켜 줍니다. 다만, 게임이 시작되는 것이므로 로켓이 꺼질 일이 없어야 합니다. 따라서 계속 반복해서 로켓만 따라다니도록 무한반복 흐름을 추가합니다.



여기서 한가지 중요한 점이 있습니다.

로켓 오브젝트와 불 오브젝트간의 연결을 표현할 때 오른쪽 그림과 같이 중심점을 서로 맞춰야 불 오브젝트가 로켓 오브젝트의 위치로 이동할 때 자연스럽게 보입니다.





로켓당음 신호(프로시저)

로켓당음 신호를 처리하기 위한 동전 오브젝트와 괴짜박사 오브젝트의 블록 프로그래밍 방법에 대해서 알아보도록 하겠습니다.

① 동전 오브젝트



동전 오브젝트가 떨어지면서(y좌표가 하나씩 감소하면서) 로켓 오브젝트를 닿았다는 것을 확인하면 로켓이 동전을 먹은 것이 되므로 점수를 10점 올려줍니다.



그리고 올려진 점수가 100점이면 다음장면 시작하기로 게임을 종료 단계로 넘어갑니다.



② 괴짜박사

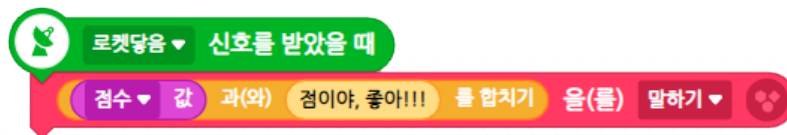
게임이 시작되고 로켓이 동전을 먹으면 점수가 올라갑니다. 하지만 이것은 컴퓨터 내부적으로 일어나는 것일뿐 사용자는 이것을 알 수 없습니다.

앞에서 설명한 것과 같이 진정한 컴퓨팅이 되려면 '입력-처리-출력'의 과정이 있어야 합니다. 다시 말해서,

- 로켓이 동전을 먹는 '입력'
- 동전을 먹음으로서 점수를 올리는 '처리'
- 점수가 올라가는 것을 보여주는 '출력'

이 과정이 필요한 것입니다.

따라서 괴짜박사 오브젝트가 점수를 표현함으로써 출력과정을 구현하는 것입니다.



실제 설계된 이벤트를 스크립트로 만들어서 블록 프로그래밍으로 하나씩 구현해 보았습니다.

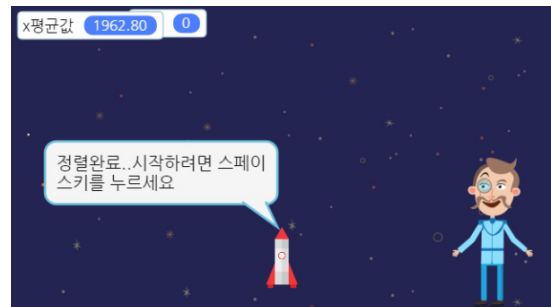
이제 프로그램을 실행해 보고 개선할 부분을 찾아봅시다.

프로그램 테스트와 디버깅

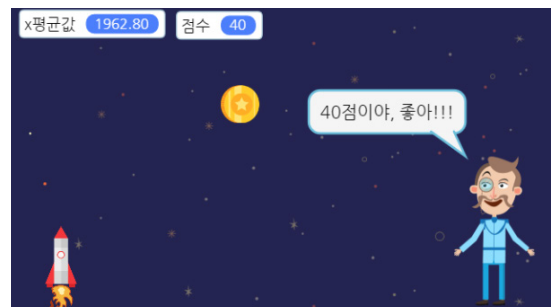
프로그램을 실행하면 다음과 같이 's'키를 눌러서 조이스틱 좌표를 정렬하라고 안내합니다.



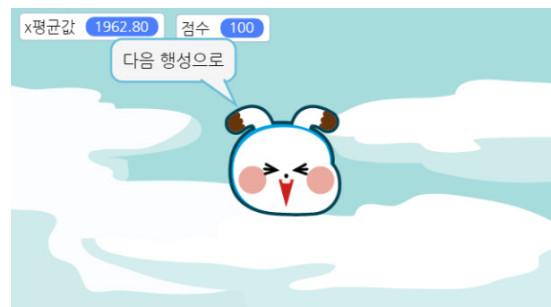
's'키를 눌러서 조이스틱의 x좌표를 정렬하면 다음과 같이 x평균값이 표시됩니다. 그리고 정렬이 완료됐다는 메시지가 나타납니다. 그리고 '스페이스 키'를 누르면 게임을 시작할 수 있다는 것을 안내합니다.



'스페이스 키를 누르면' 동전들이 떨어지기 시작합니다. 초코파이보드의 조이스틱을 좌우로 움직여 떨어지는 동전을 먹으면 점수가 올라가고 괴짜박사가 해당 점수를 말해 줍니다.



점수가 100점이 되면 다음장면으로 넘어가서 게임이 종료되는 것을 볼 수 있습니다.





SUMMARY 써머리~

- 초코파이보드에 연결되는 센서들은 정보를 수집하기 위한 고유의 속성을 가지고 있습니다. 조이스틱은 좌우, 상하값, 버튼 1,2,3,4의 속성을 가지고 있습니다.
- 제시된 문제를 해결하기 위해서는 이벤트의 순서에 따라 스크립트를 짜는 것도 중요하지만 오브젝트에서 만들어지는 신호(프로시저)들을 어떻게 연결하는지 분석하는 것이 더 중요합니다.
- 생각을 시각화하는 것은 수집된 정보를 보다 구조화하고 논리적으로 연결하기 위한 좋은 방법입니다. 스마트기기의 메모앱, 마인드맵 도구등을 사용하여 수집된 정보를 시각화하는 연습이 필요합니다.

이제 좀 더 게임적인 요소를 추가하려면 무엇이 필요할까요?

여러분이 생각하는 것들을 적어보기 바랍니다.

· 오브젝트

· 이벤트

게임은 제한적인 요소들(생명, 점수)을 잘 활용하면 훨씬 더 재미있게 만들 수 있습니다.



미션을 해결해봅시다!

이전 섹션에서 다루었던 내용을 기초로 하여 자신만의 아이디어를 추가해서 프로그램을 더욱 발전시킬 수 있는 섹션입니다.

MISSION 01 동전 따먹기 2

이번 챕터에서는 ‘동전 따먹기’ 게임을 업그레이드하는 내용으로 만들어 보겠습니다.

점수가 100점이 되기 어렵게 만들려면 어떻게 해야 할까요?

처음에는 쉽게 점수를 얻다가 40점이 넘어가면 새로운 규칙을 넣어 점수따기가 어렵게 만들어 보겠습니다. 그 방법은 다음과 같습니다.

- 동전이 떨어지는 속도가 빨라진다.
- 로켓이 점수가 40점 아래면 이동속도가 3이지만 40점이 넘어가면 2로 바뀐다.

이 밖에도 더 재미있는 것들이 많겠지만 여기에서는 40점이 넘어가면 ‘동전이 떨어지는 속도가 빨라진다’를 처리할 수 있는 프로그램을 만들어 봅시다.

① 이벤트 디자인하기

수정해야 할 이벤트는 무엇일까요?

속도와 관련된 규칙을 추가하려면 아래 내용을 확인합니다.

- 스페이스 키를 눌렀을 때 동전 오브젝트의 떨어지는 알고리즘
- 로켓닿음 신호를 받았을 때 점수를 계산하고 처리하는 부분

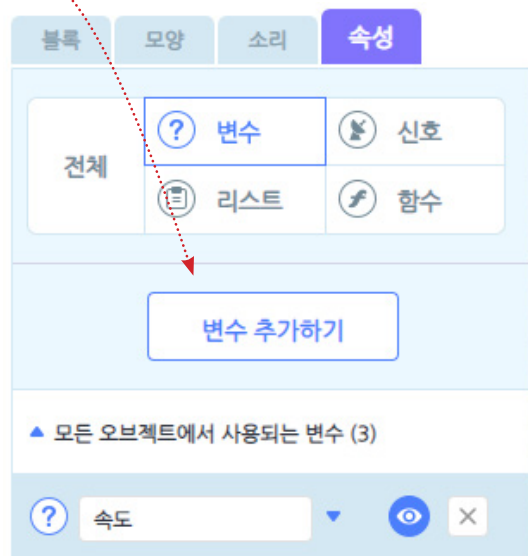



앞에서 제시한 문제를 해결할 수 있도록 구조화하여 표현해 봅시다.

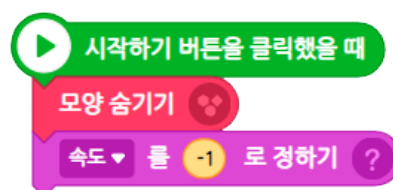
- 스페이스 키
 - 동전
- 로켓닿음
 - 동전

② 이벤트에 따른 스크립트 수정하기

먼저 속도를 제어할 수 있는 변수를 만듭니다.



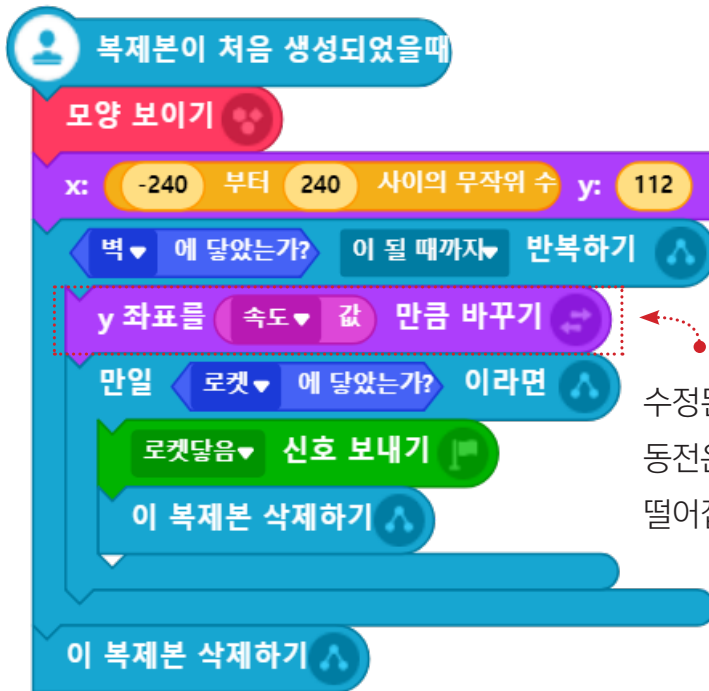
- 시작하기 버튼을 클릭했을 때 동전오브젝트()를 클릭하고 동전의 떨어지는 속도를 기본으로 -1로 프로그래밍합니다.



③ 오브젝트와 이벤트 연결하기

· 동전오브젝트

먼저 동전이 떨어질 때 설정되어 있는 속도로 떨어질 수 있도록 동전 오브젝트의 '복제본이 처음 생성되었을 때' 블록을 수정합니다.



복제본이 처음 생성되었을 때

- 모양 보이기
- x: -240 부터 240 사이의 무작위 수 y: 112 위치로 이동하기
- 벽에 닿았는가? 이 될 때까지 반복하기
 - y 좌표를 속도 값 만큼 바꾸기 (수정된 부분)
 - 만일 로켓에 닿았는가? 이라면
 - 로켓닿음 신호 보내기
 - 이 복제본 삭제하기
- 이 복제본 삭제하기

수정된 부분
동전은 설정된 속도(시작할 때는 -1)로 떨어집니다.

다음으로 '로켓닿음' 신호를 받을 때 점수가 올라가고 점수가 '40'이 넘으면 다음과 같이 속도를 바꿔줍니다. 여기서는 5배 빠르게 떨어지도록 하였습니다.



로켓닿음 신호를 받았을 때

- 점수에 10 만큼 더하기
- 만일 점수 값 ≥ 40 이라면
 - 속도를 -5 로 정하기
- 만일 점수 값 = 100 이라면
 - 다음 장면 시작하기



④ 프로그램 테스트와 디버깅

시작 버튼을 클릭하여 프로그램을 실행하여 봅시다.

40점이 넘어가니까 속도가 엄청나게 빨라졌다는 것을 느낄 수 있습니다.



더 수정하거나 개선해야 할 점이 무엇인지 적어 보세요.

입력받은 정보를 처리하기 위한 것이나 알고리즘을 개선할 필요가 있는 것들을 생각해 보시기 바랍니다. 예를 들어 동전이 나타나는 간격을 더 짧게 할 수도 있을 것입니다.

MISSION 02 구구단

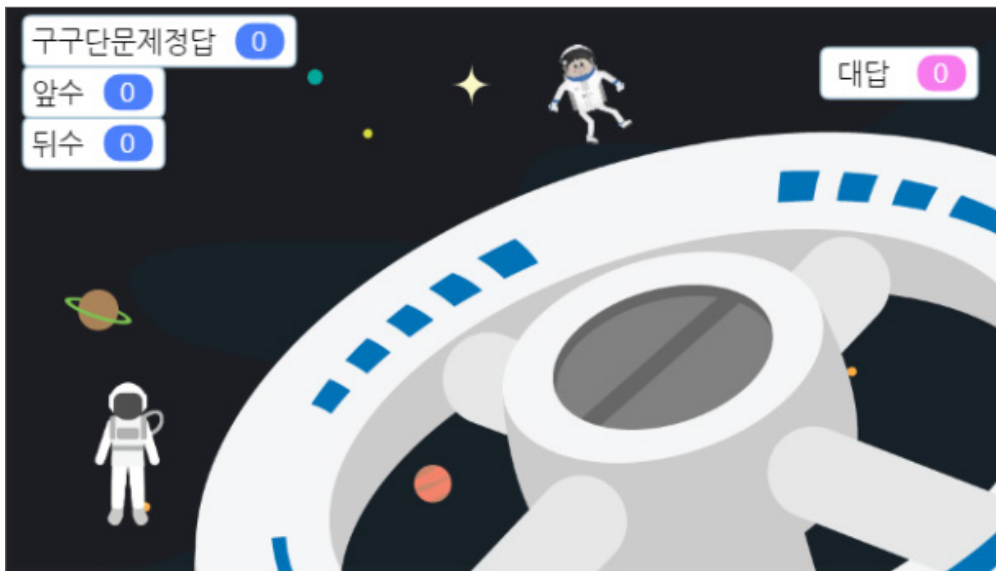
이번에는 금성으로 가보겠습니다.

금성에 있는 우주정거장에서 지구의 연구소에게로 계속 수신되는 전파가 있었습니다.

‘치치치지지지직...’

‘I’m Comming Home’ 집으로 간다는 오랜 팝송인데.. 아니 이것은 코드명 ‘잃어버린기억’의 목소리가 아닌가? 탐험대는 지구에서 파일럿을 띄워서 금성에 있는 우주 정거장으로 갔습니다.

다음과 같은 장면이 펼쳐져 있었습니다.



영혼없이 우주를 떠돌고 있는 ‘잃어버린기억’ 우주인의 노래 소리였습니다.

이제 우주인을 상상하며 게임을 만들어 봅니다.

- 조이스틱 볼륨버튼을 사용하여 ‘잃어버린기억’에게로 간다.
- ‘잃어버린기억’과 닿으면 ‘잃어버린기억’은 구구단 문제를 낸다.
- 구구단 정답을 입력하면 게임이 종료된다.



앞장의 게임을 어떻게 프로그래밍해야 할까요? 다양한 생각을 적어봅니다.

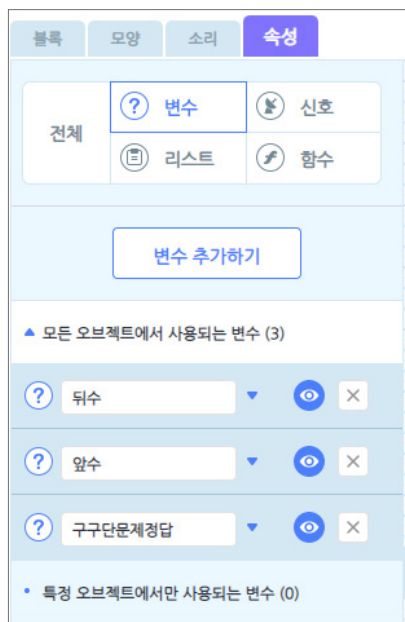
① 오브젝트별 샘플 블록 프로그래밍 살펴보기

· 오브젝트 배치

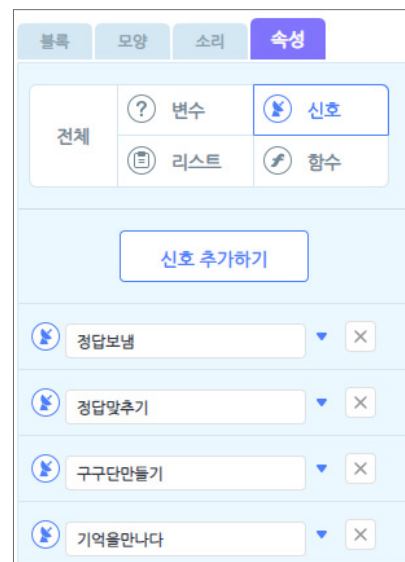
잃어버린 기억, 우주인, 우주정거장 오브젝트를 추가하고 오른쪽 그림과 같이 배치 합니다.



· 변수 만들기



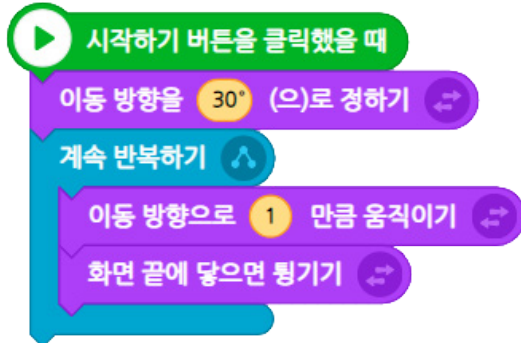
· 신호 만들기






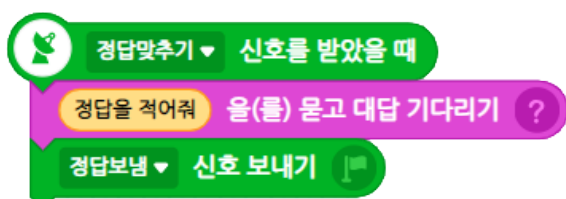
· 프로그래밍 하기

- '잃어버린 기억' 오브젝트()를 클릭하고 프로그래밍합니다.





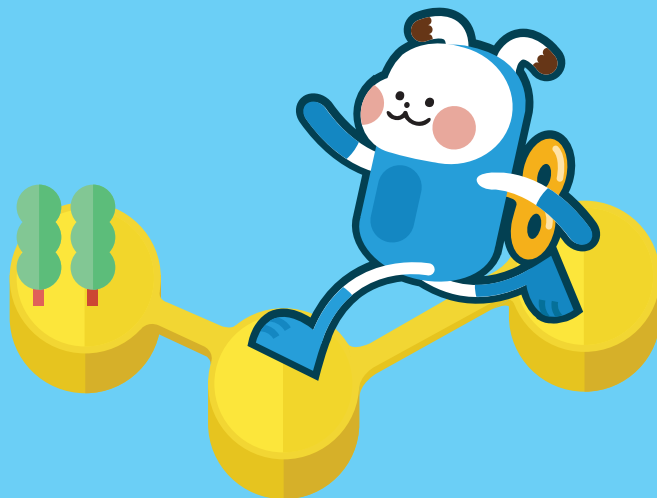
- '우주인' 오브젝트()를 클릭하고 프로그래밍합니다.



부록 1

기초 배우기

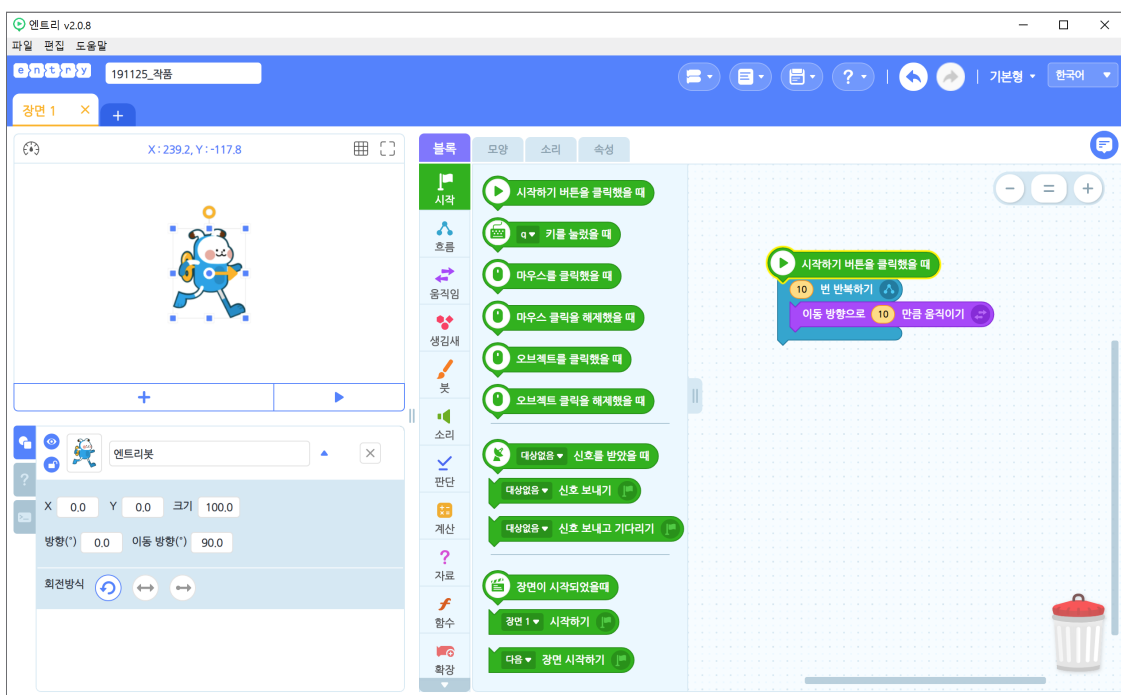
엔트리와 초코파이보드 알아보기





엔트리 사용하기

컴퓨터를 이용하여 재미있는 이야기를 꾸며봅시다. 컴퓨터 프로그램을 이용하면 이야기 순서에 따라 등장인물들의 동작이나 말을 다양하게 표현할 수 있습니다. 프로그램은 컴퓨터에게 시킬 명령어들을 모아 둔 것입니다. 컴퓨터 프로그램을 사용하면 복잡한 일도 쉽게 처리할 수 있도록 도와줍니다.



여러분들이 알고 있는 이야기를 프로그램으로 만들 수 있는 도구가 '엔트리'입니다. 엔트리는 커넥트 재단에서 제공하는 컴퓨터 프로그래밍 도구입니다. 프로그래밍이란 프로그램을 만드는 과정으로서, 엔트리를 이용하면 다양한 블록 명령어를 조립하면서 컴퓨터 프로그램을 쉽게 만들 수 있습니다. 이러한 엔트리를 사용하려면 어떻게 해야 하는지 알아봅시다.



엔트리 홈페이지 방문하기

여러분들이 사용하는 검색 사이트에 방문해 보세요. 그리고 검색 창에 '엔트리'라고 입력한 후 키보드에 있는 엔터키를 눌러 주세요. '다음' 검색 사이트에 접속한 후 '엔트리'를 입력하면 아래와 같이 검색 결과가 나타납니다.

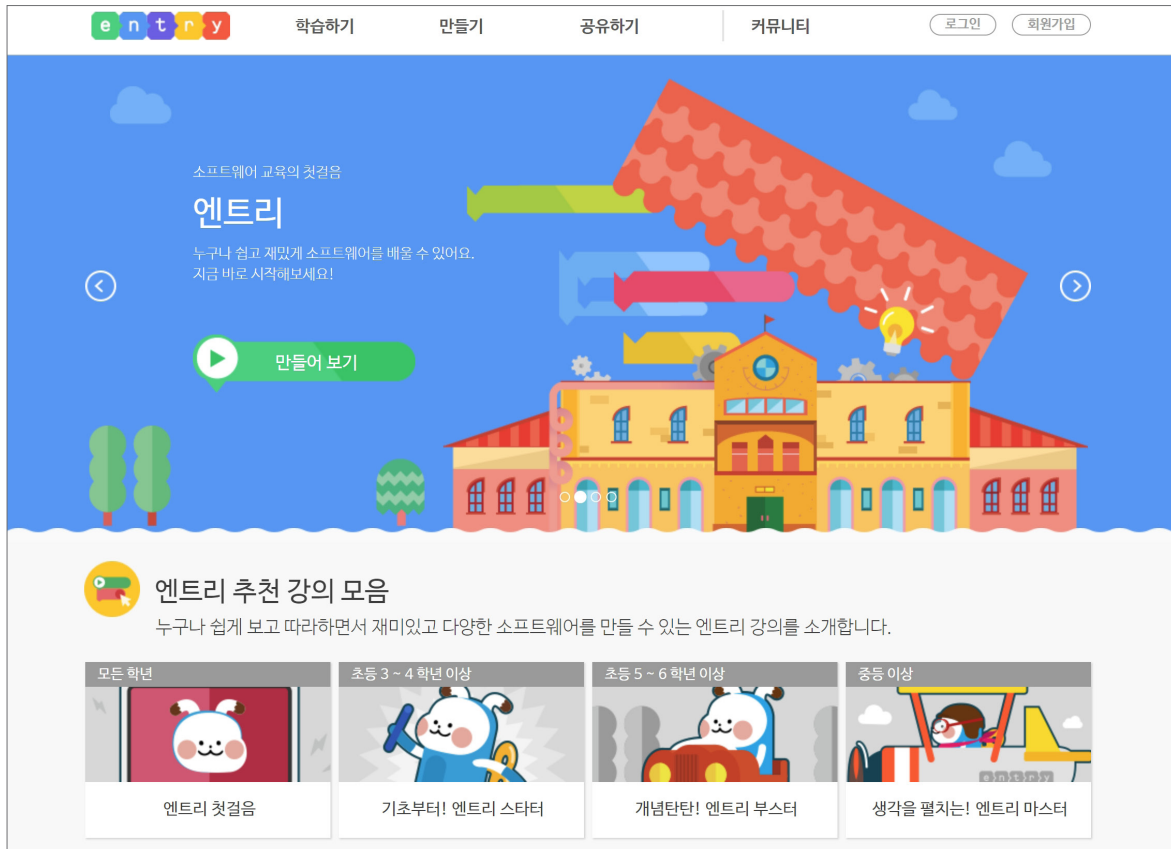
여러분들이 사용하고 있는 검색 사이트에 방문해 보세요. 그리고 검색 창에 '엔트리'라고 입력한 후 키보드에 있는 엔터키를 눌러 주세요. '다음' 검색 사이트에 접속한 후 '엔트리'를 입력하면 다음과 같이 검색 결과가 나타납니다.



여러분이 찾은 엔트리 홈페이지의 주소는 무엇인가요? 그 주소를 아래의 빈칸에 써보세요.

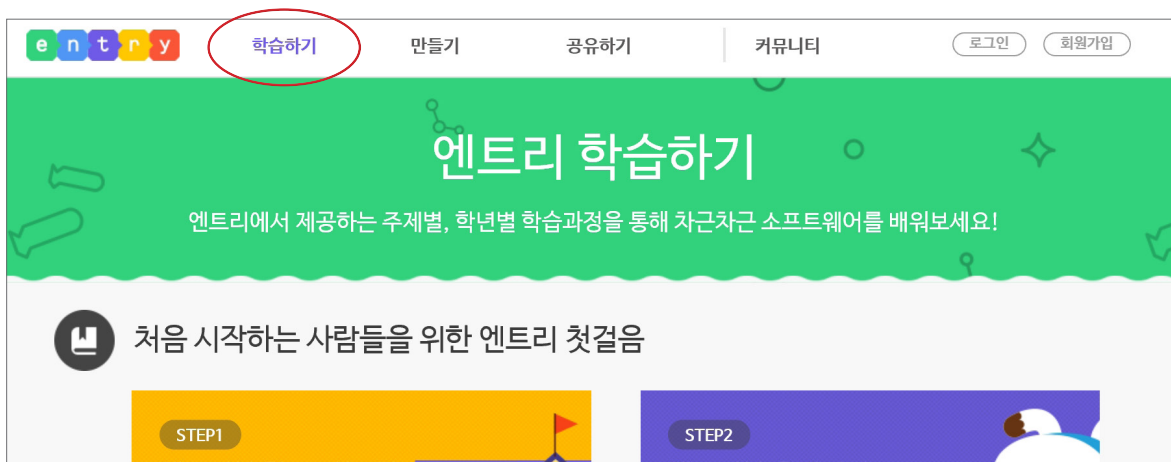
엔트리 홈페이지 주소

엔트리를 잘 사용하려면 먼저 엔트리에 어떤 기능이 있는지 살펴봐야 합니다.



엔트리 홈페이지의 기능은 크게 학습하기, 만들기, 공유하기, 커뮤니티 등 네 가지 메뉴가 있습니다. 각각의 기능에 대해 구체적으로 살펴봅시다.

① 학습하기



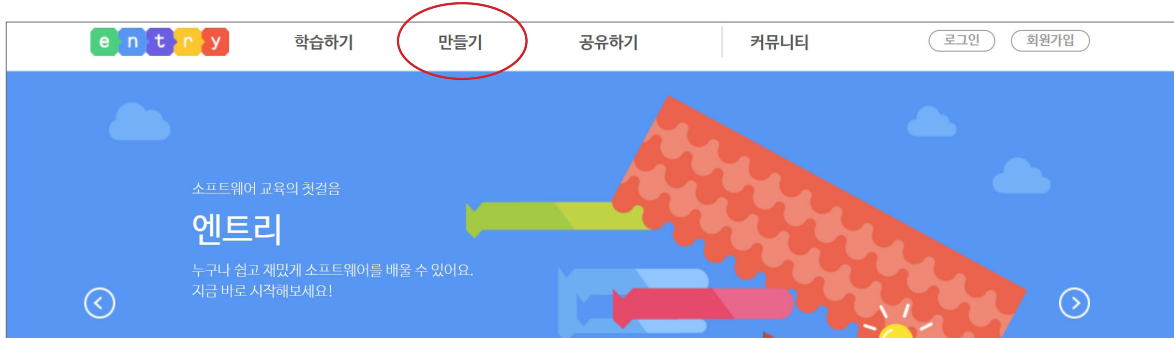


‘학습하기’는 엔트리를 스스로 배울 수 있는 공간입니다. 학습하기에서는 엔트리를 이용하여 여러 가지 문제를 해결할 수 있는 다양한 자료가 준비되어 있습니다. 주어진 문제를 해결하기 위해 게임하듯이 프로그래밍을 배울 수 있습니다. 또한, 재미있고 유익한 동영상을 통해 프로그래밍의 원리를 이해할 수 있습니다.

초등학교 학생들을 위해 학년별 학습 과정도 다음 표와 같이 제공합니다. 따라하기, 도전하기, 퀴즈 풀기, 자유롭게 만들기 등 4단계로 구분되어 있어 자신의 학년이나 수준에 따라 선택하여 프로그래밍을 배울 수 있습니다.

학년	학습과정 이름	학습 요소
초등 3~4학년	엔트리봇의 학교 가는 길	순차, 반복
	동물 친구 이야기 만들기	순차, 병렬, 반복
	꽃송이 만들기	반복
	파리잡기 게임 만들기	반복, 이벤트
	어항 만들기	반복, 이벤트
	도장으로 미술작품 만들기	반복, 이벤트
	애완동물 만들기	반복, 이벤트
초등 5~6학년	엔트리봇의 전기 자동차	순차, 반복, 선택
	생일축하 카드 만들기	순차, 반복, 이벤트
	달리기 게임 만들기	반복, 선택, 이벤트
	로봇 청소기 만들기	반복, 선택
	세계의 국가 수도 맞추기	입출력, 선택, 반복
	계산 로봇 만들기	변수, 입출력, 문자열
	두더지 잡기 게임 만들기	변수, 반복, 선택

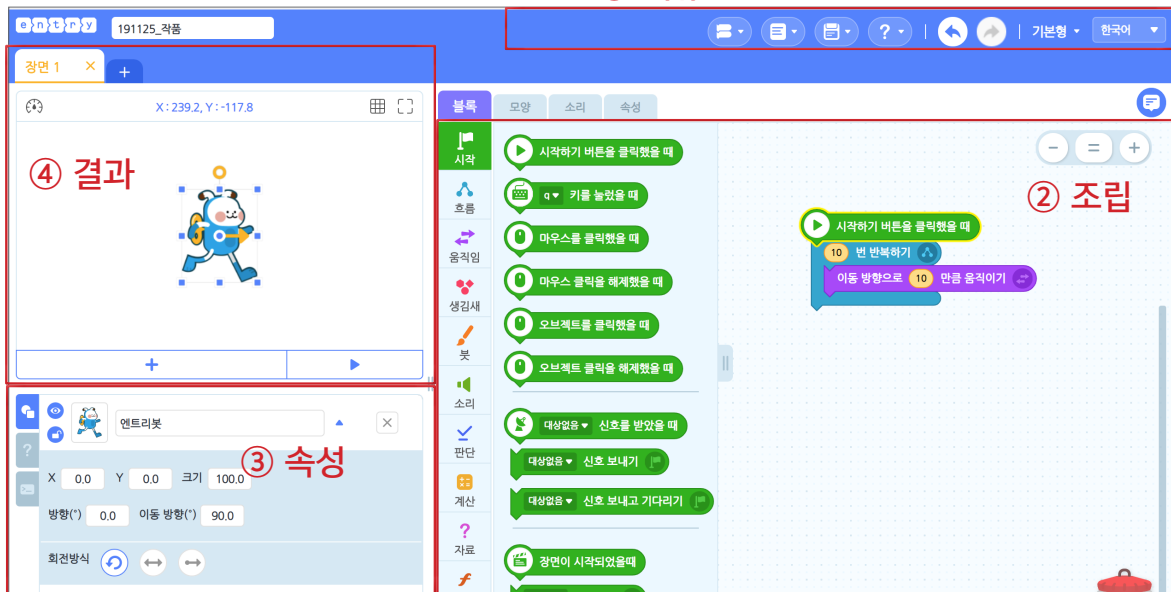
② 만들기



‘만들기’는 엔트리를 이용해 프로그램을 만들 수 있는 공작소입니다. 블록형 프로그래밍 언어를 처음 접하는 학생들도 여러 가지 블록들을 조립하면서 자신만의 프로그램을 만들 수 있습니다. 오브젝트를 움직이거나 모양, 색깔, 크기 등을 자유롭게 바꿀 수 있으며, 붓을 이용하여 그림도 그릴 수 있습니다.

만들기 화면의 구성은 다음과 같이 메뉴, 조립, 속성, 결과 등 크게 4가지로 구분할 수 있습니다.

① 메뉴



엔트리 작품을 엔트리 홈페이지에 저장하여 어디서든지 수정하려면 회원에 가입하는 것이 편리합니다. 만약 작품을 엔트리 홈페이지에 저장하지 않고 내 컴퓨터에만 저장한다면 굳이 회원에 가입할 필요는 없습니다. 세부적인 엔트리 메뉴별 주요 기능을 살펴보면 다음과 같습니다.



메뉴		주요 기능
	모드	블록코딩이나 엔트리파이선 모드로 바꿀 수 있습니다.
	파일	프로그램을 새로 만들거나, 저장된 프로그램을 불러올 수 있습니다.
	저장하기	프로그램을 내 컴퓨터나 엔트리 홈페이지에 저장할 수 있습니다.
	도움말	프로그램 만드는 방법이나 블록들의 사용법을 안내합니다.
	장면	장면을 이전이나 이후로 이동할 수 있습니다. 단, 1개의 장면만 있을 경우에는 사용할 수 없습니다.
	유형	엔트리의 전체적인 기능을 기본형이나 교과형으로 바꿀 수 있습니다.
	언어	엔트리에서 사용하는 기본적인 언어를 한국어, 영어, 베트남 언어로 바꿀 수 있습니다.

조립은 크게 블록, 모양, 소리, 속성 등 4가지로 구분됩니다.

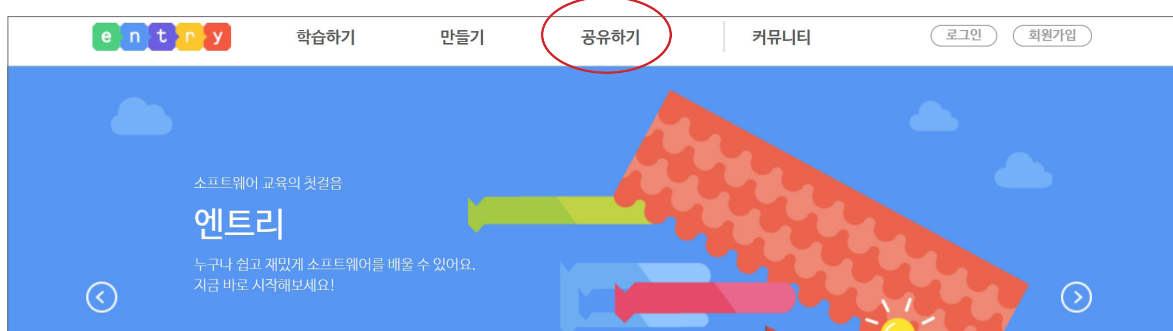
첫째, 블록은 오브젝트를 움직이게 하거나 모양 등을 변경할 때 사용할 수 있는 블록 명령어들을 제공합니다. 엔트리에서 제공하는 블록에는 시작, 흐름, 움직임, 생김새, 빛, 소리, 판단, 계산, 자료, 함수, 확장 하드웨어 등이 있습니다.

둘째, 모양은 오브젝트의 모양을 변경할 수 있습니다. 오브젝트의 크기나 색 등을 자유롭게 수정할 수 있고, 새로운 모양을 추가할 수 있습니다.

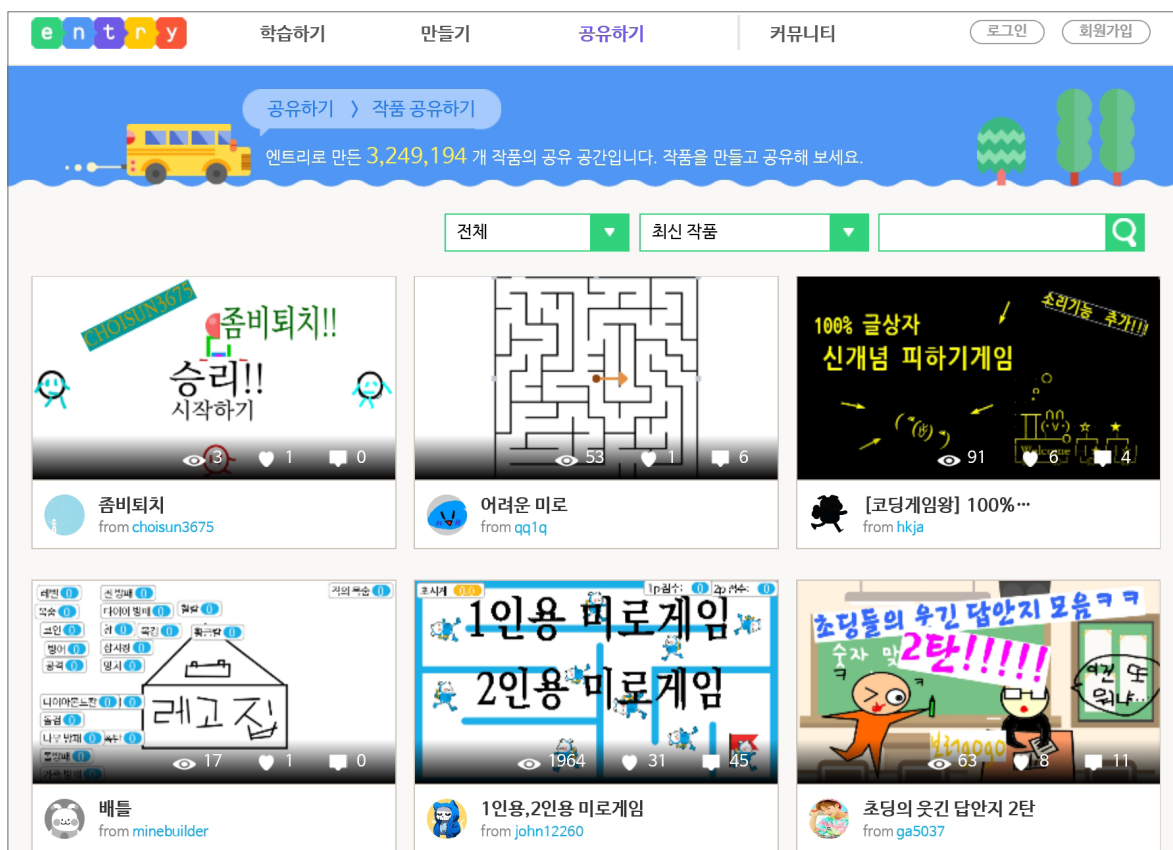
셋째, 소리는 오브젝트에 대한 효과음이나 배경음악을 추가할 수 있습니다. 엔트리는 사람, 자연, 사물, 판타지, 악기 등 다양한 효과음을 제공하고 있습니다.

넷째, 속성은 변수, 신호, 리스트, 함수 등 다양한 기능을 제공합니다. 변수를 추가하거나 신호를 추가할 수 있으며, 리스트나 함수를 새롭게 만들 수 있습니다.

③ 공유하기



‘공유하기’는 다른 친구들이 만든 작품을 살펴보거나, 자신이 만든 작품을 다른 친구들이 볼 수 있도록 엔트리 홈페이지에 저장할 수 있습니다. 엔트리는 누구든지 자신이 만든 작품을 엔트리 홈페이지에 공유하여 다른 친구들에게 도움을 주거나, 자신의 작품을 자랑할 수 있습니다.

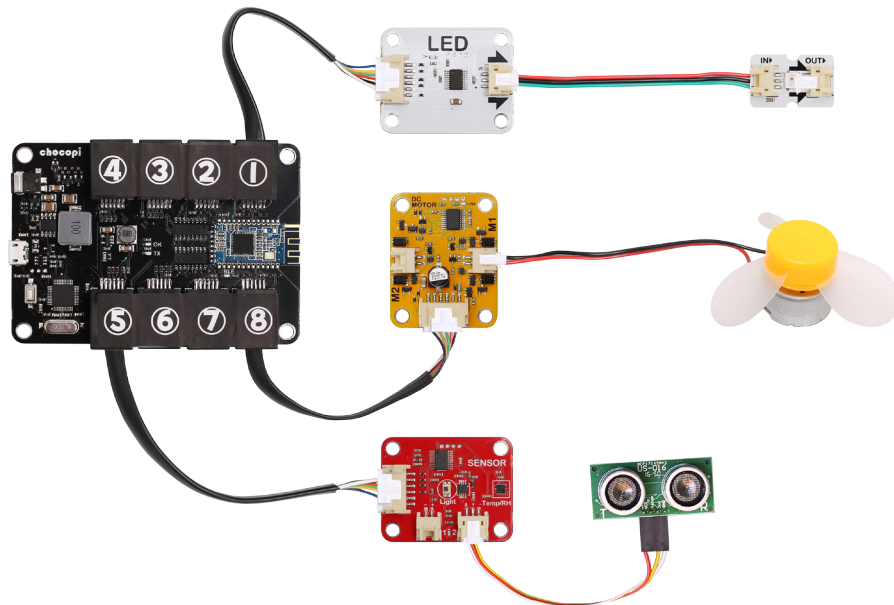




초코파이 보드 사용하기

초코파이 보드는 과학기기를 개발하고 SW 교육을 지원하는 ‘한국과학’이라는 회사에서 2016년 8월에 개발한 교육용 프로그래밍 도구입니다. 프로그래밍은 컴퓨터가 이해할 수 있는 언어를 이용하여 프로그램을 만드는 활동을 의미하며, 프로그램은 컴퓨터에게 내리는 명령어들을 모아 둔 것입니다.

사실, 초코파이 보드만으로는 프로그램을 만들 수는 없습니다. 프로그래밍 언어를 통해 초코파이 보드에 명령을 내려야 하는데 이때 사용되는 언어가 ‘엔트리’입니다. 엔트리를 이용하면 초코파이 보드와 연결된 여러 가지 센서와 버튼, LED, 모터 등을 이용하여 간단한 생활용품을 만들 수 있습니다.



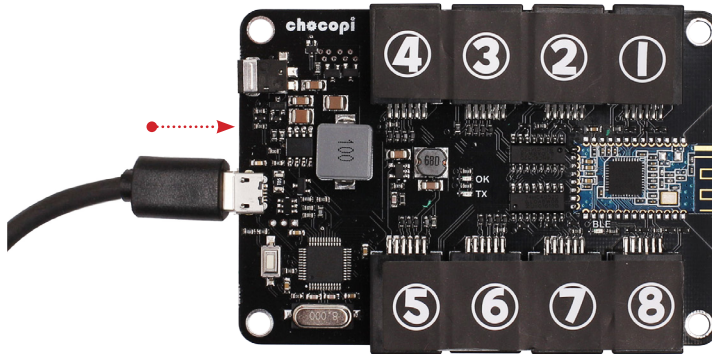
[그림 1] 초코파이 보드의 연결

이번 장에서는 엔트리에서 초코파이 보드를 연결하는 방법을 배우고, 초코파이 보드에서 제공하는 여러 가지 센서를 살펴봅니다.

초코파이 보드와 컴퓨터 연결하기

초코파이 보드의 센서 블록이나 터치 블록, LED 등을 이용한 프로그램을 만들려면 어떻게 해야 할까요? 우선, 초코파이 메인보드를 컴퓨터와 연결해야 합니다. 그리고 초코파이 보드를 활용하는 데 필요한 명령 블록들이 엔트리에 나타나도록 초코파이 보드와 엔트리를 연결해야 합니다.

초코파이 메인보드를 컴퓨터에 연결하기 위해서 아래 그림과 같이 초코파이 메인보드에 USB 케이블을 연결합니다.



이때 USB 케이블의 연결 부분의 모양과 초코파이 메인보드의 연결 부분이 일치하는지 잘 살펴보고 연결합니다. USB 케이블을 초코파이 메인보드에 반대로 꽂을 경우에는 잘 들어가지 않습니다. USB 케이블의 둥근 부분과 평평한 부분을 확인한 후에 USB 케이블을 초코파이 메인보드에 꽂습니다.



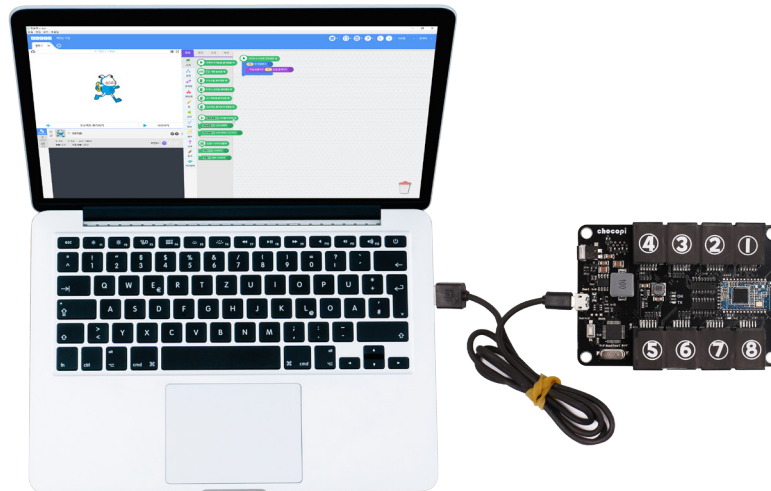
[초코파이 메인보드]



[USB 케이블]



이제, USB 케이블을 엔트리가 설치된 컴퓨터와 연결합니다. 초코파이 메인보드를 연결한 USB 케이블을 아래 사진과 같이 컴퓨터의 USB 단자에 꽂습니다. 컴퓨터와 초코파이 메인보드가 제대로 연결되면, 초코파이 메인보드에서 노란 불이 깜빡거립니다.



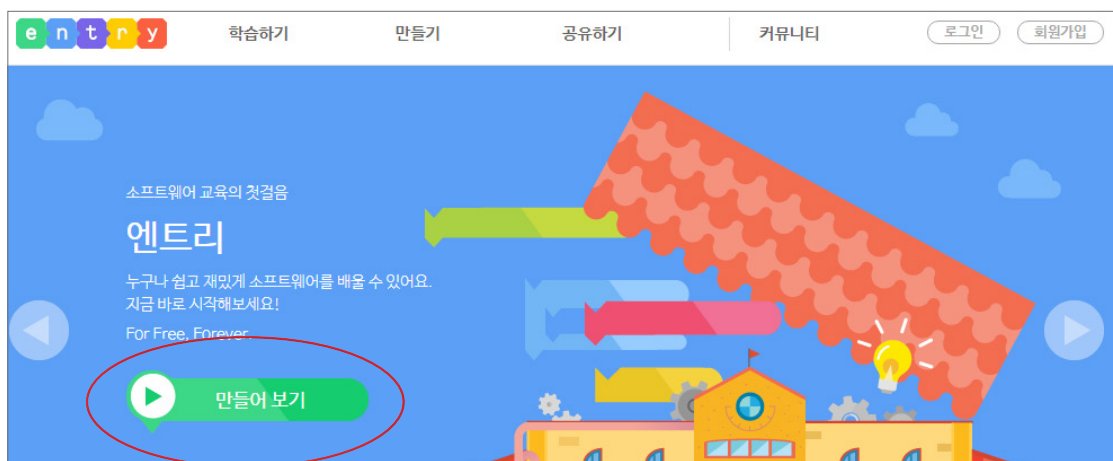
초코파이 보드와 엔트리 연결하기

초코파이 보드를 엔트리에 연결하는 방법은 두 가지가 있습니다.

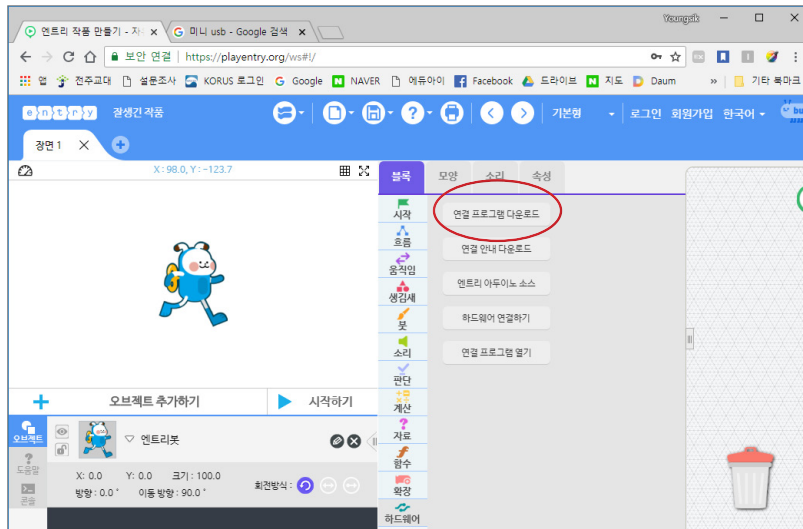
첫 번째 인터넷이 연결된 상태에서 사용하는 온라인 실행 방법과 두 번째 윈도우에 프로그램을 설치하고 사용하는 오프라인 실행 방법이 있습니다. 두 가지 설치 및 연결 방법에 대해 알아보시다.

① 온라인 연결 방법

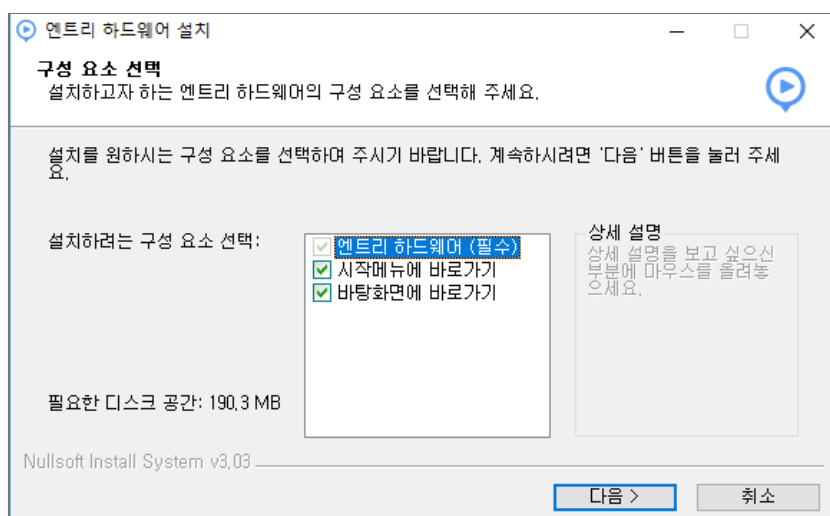
웹 브라우저를 실행해서 엔트리 홈페이지(<https://playentry.org>)에 접속합니다. 홈페이지에서 [만들어 보기]를 클릭하면 온라인 엔트리 프로그램이 실행됩니다.



초코파이 메인보드와 연결된 LED 블록이나 센서 블록, 터치 블록을 활용하려면, 엔트리에서 초코파이 보드와 관련된 명령 블록을 설치해야 합니다. 엔트리를 실행한 후에 [블록] 창의 맨 아래에 있는 '하드웨어'를 클릭합니다. 아래 그림과 같이 [연결프로그램 다운로드]를 클릭하여 초코파이 보드를 연결하는 데 필요한 프로그램을 다운받아서 설치합니다.

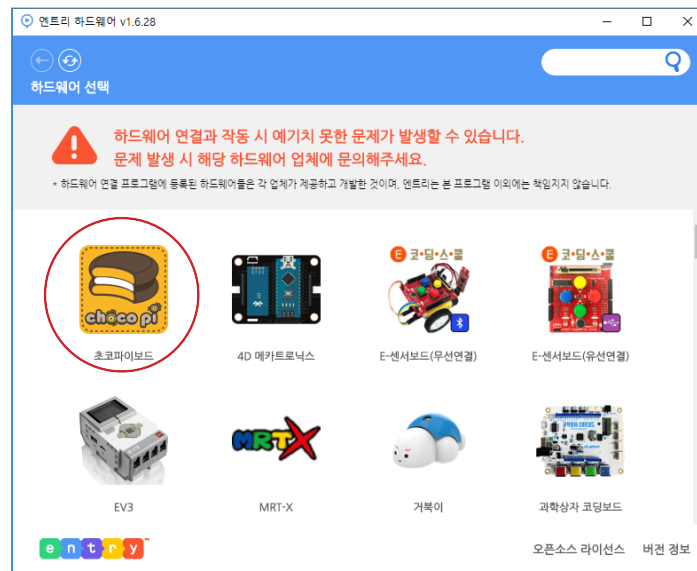


[연결프로그램 다운로드]를 클릭하면 연결 프로그램(예: Entry_HW_1.6.28_Setup.exe)이 다운로드됩니다. 모두 다운로드가 되면 그 파일을 클릭하여 프로그램을 설치합니다. 아래 그림과 같이 엔트리 하드웨어 설치 프로그램이 실행되면서 초코파이 보드와 관련된 프로그램들이 함께 설치됩니다.





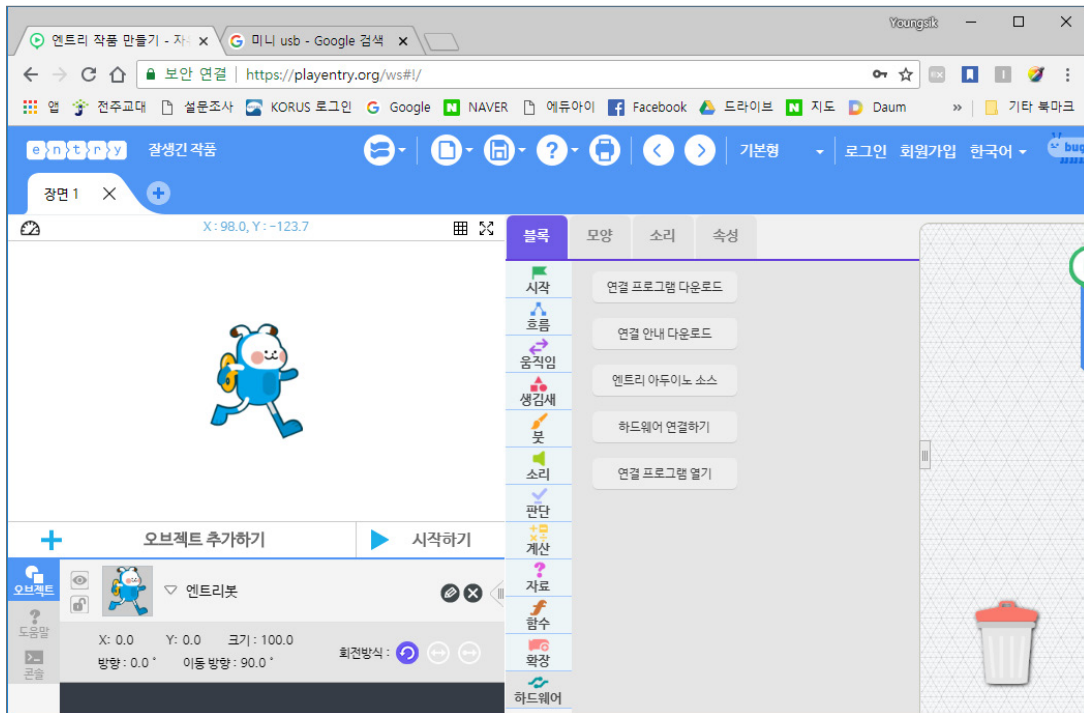
엔트리 하드웨어 프로그램이 설치되면 자동으로 아래 그림과 같이 ‘엔트리 하드웨어’ 연결 프로그램이 실행됩니다. 만약 자동으로 실행되지 않으면, 엔트리의 블록 창에서 [하드웨어]를 클릭한 후에 [연결 프로그램 열기]를 클릭하면 실행됩니다. 이때 초코파이 보드 그림을 찾아 클릭합니다.



초코파이 보드를 선택하면 아래 그림과 같이 초코파이가 연결된 포트를 선택하는 메뉴가 나타납니다. 우선 COM1을 선택한 후에 [연결] 버튼을 클릭합니다. 초코파이 보드가 성공적으로 연결되면 오른쪽 그림과 같이 ‘하드웨어가 연결되었습니다.’라고 표시됩니다. 만약 포트를 잘못 선택한 경우에는 ‘하드웨어와 연결 중입니다’라고 표시되므로, 이때는 성공적으로 연결될 때까지 다른 포트를 차례대로 선택합니다.



성공적으로 연결되면, 초코파이 보드와 관련된 명령 블록을 볼 수 있습니다. 초코파이 보드를 사용하는 동안에는 반드시 '엔트리 하드웨어' 연결 프로그램이 실행되고 있어야 하므로, 초코파이 보드와 관련된 명령 블록이 나타나지 않으면, '엔트리 하드웨어' 창이 닫혀 있는지를 확인합니다.



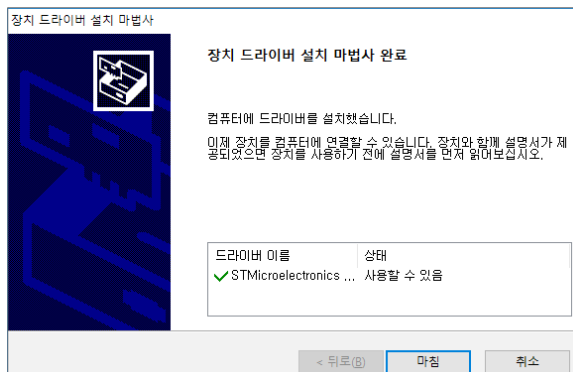
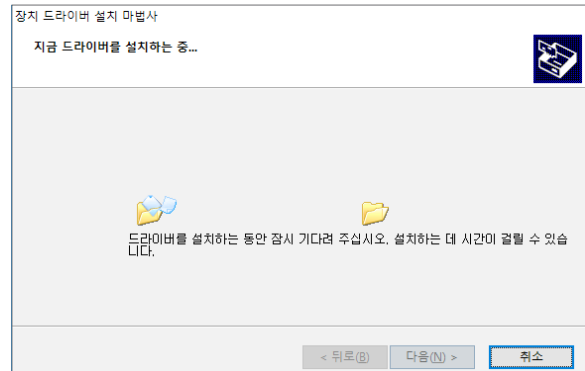
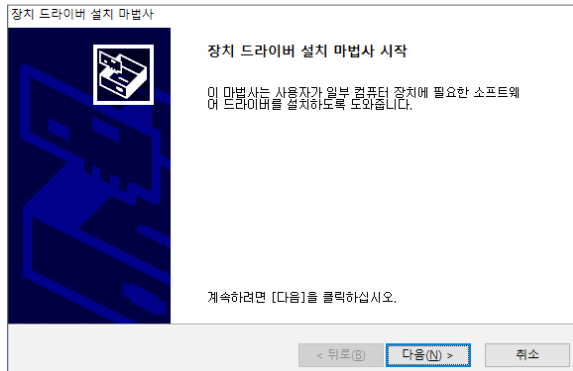
'엔트리 하드웨어' 창이 실행 중인데도 하드웨어 탭에 초코파이 명령 블록이 나타나지 않으면 드라이버를 설치해야 합니다.

드라이버 설치를 해야 할 경우 COM 포트 리스트를 닫으면 오른쪽 창이 나타납니다. 창에서 [드라이버 설치]를 클릭합니다.





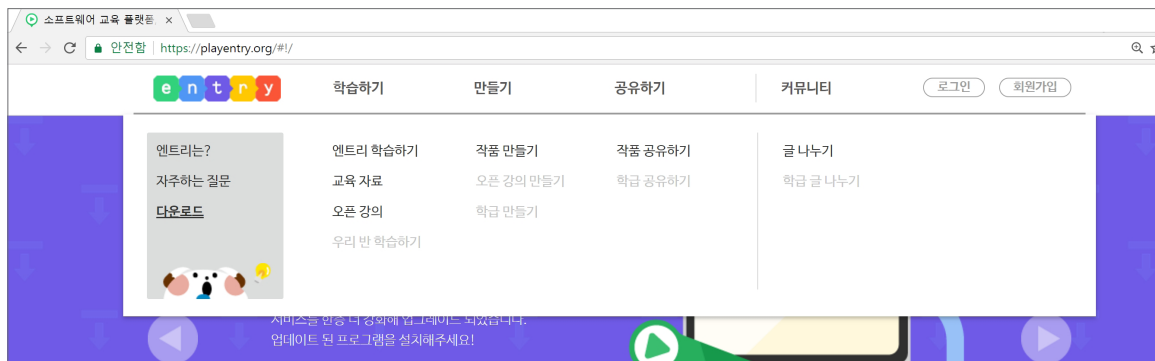
초코파이 보드를 연결하고 드라이버 설치를 진행합니다.



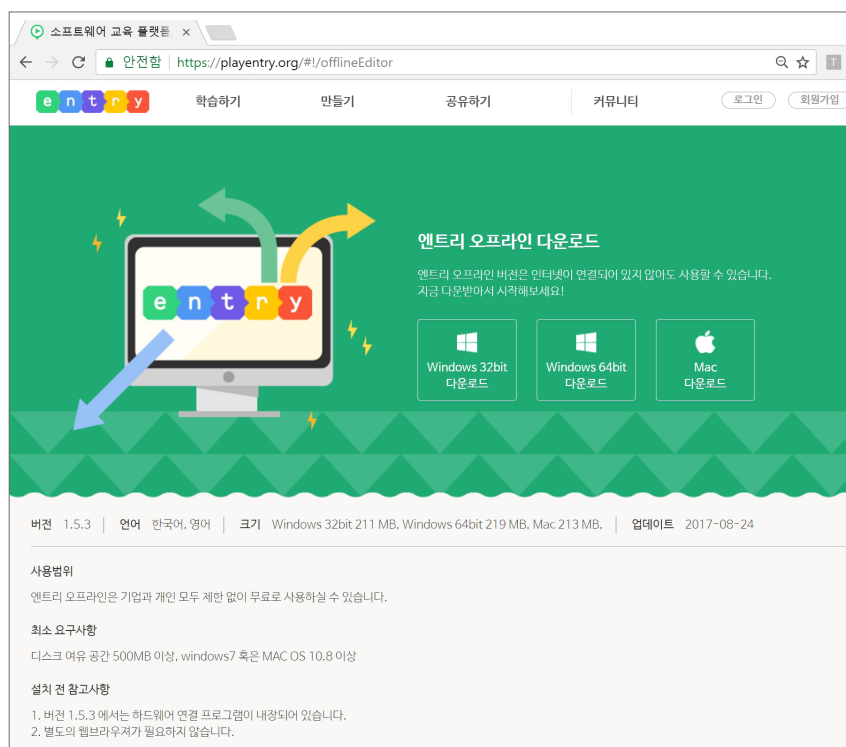
드라이버 설치 완료 후 재연결 버튼을 클릭합니다.

② 오프라인 설치

웹 브라우저를 실행해서 엔트리 홈페이지(<https://playentry.org>)에 접속합니다.
엔트리로고(**e n t r y**)에 마우스를 올리면 아래쪽에 엔트리 다운로드 메뉴가 나타나게 됩니다. 클릭하여 다운로드 창으로 이동합니다.

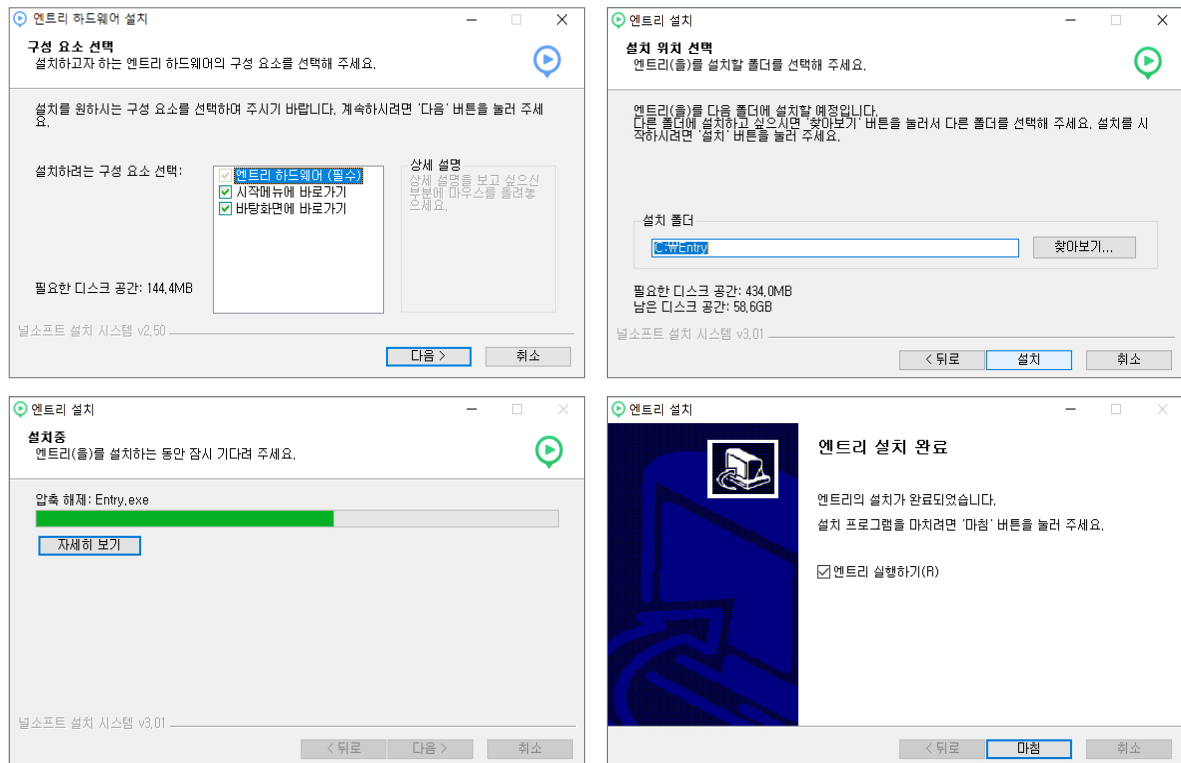


엔트리 오프라인 다운로드 창에 컴퓨터에 설치된 운영체제의 종류를 확인하고 프로그램을 다운로드합니다. 윈도의 바탕화면에서 [내 컴퓨터] 아이콘에 오른쪽 마우스를 클릭하고 속성에 들어가면 윈도 운영체제의 종류를 확인할 수 있습니다.



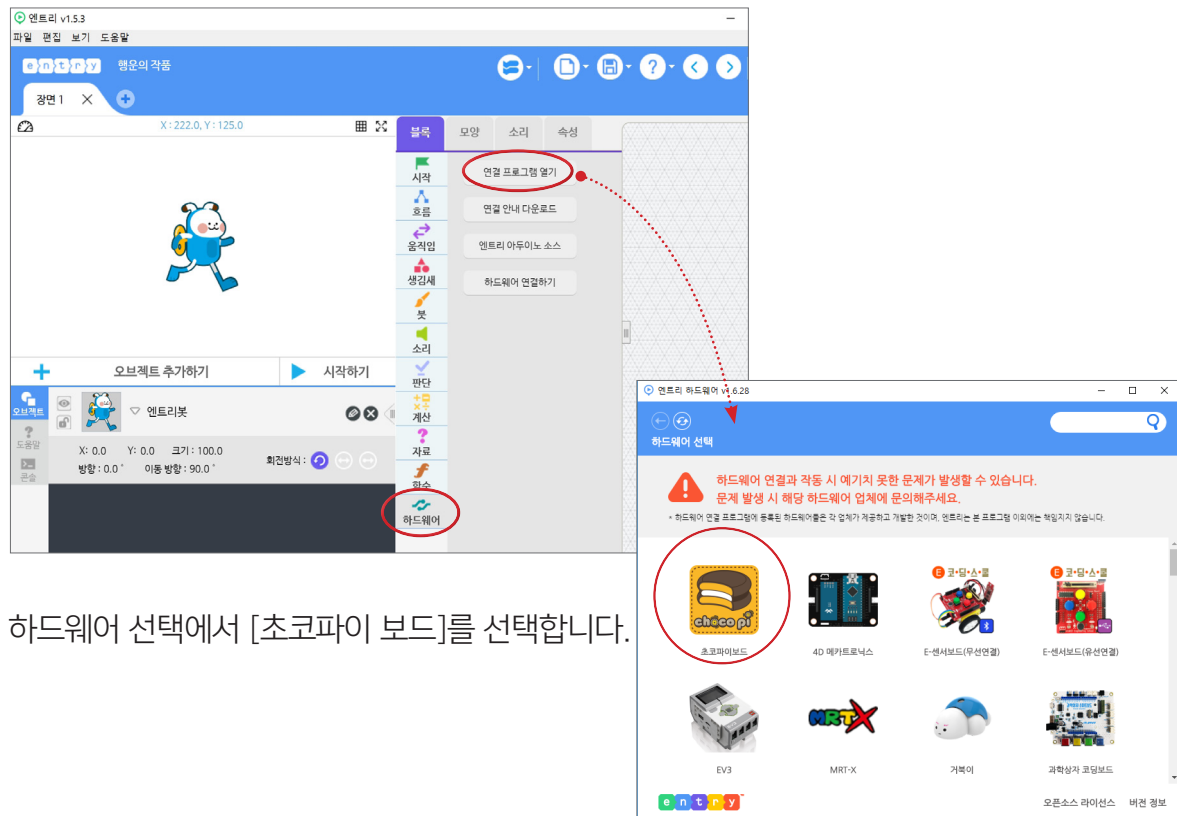


다운로드 받은 엔트리 프로그램을 실행하여 설치합니다.



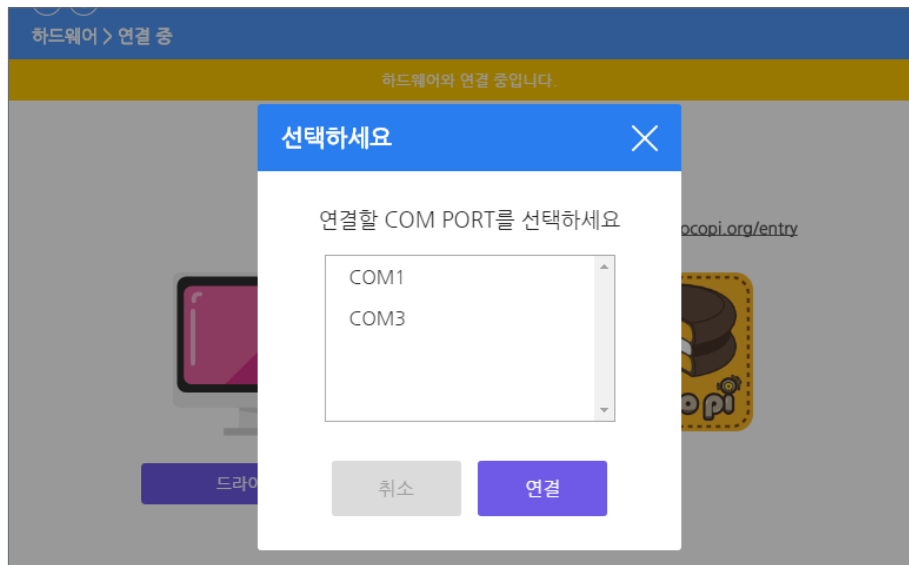
엔트리 프로그램 실행 후 하드웨어를 클릭합니다.

[엔트리 프로그램 실행] → [블록에서 하드웨어 클릭] → [연결프로그램 열기]



하드웨어 선택에서 [초코파이 보드]를 선택합니다.

초코파이 보드와 엔트리프로그램 통신역할을 하는 COM 포트를 선택합니다.



연결 성공 확인 메시지를 확인합니다.



연결 완료 후 초코파이 명령 블록이 화면에 나타나면 성공적으로 연결되었습니다.





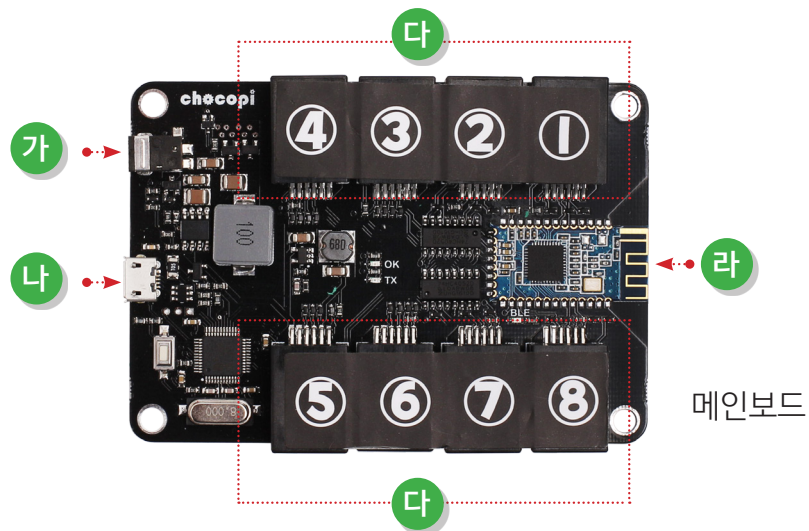
초코파이 보드 살펴보기

초코파이 보드의 주요 부품들의 구조와 연결 방법을 살펴보고, 해당 부품들을 제어하는 데 필요한 엔트리 명령 블록을 알아봅시다.



① 메인보드

초코파이 메인보드는 아래 그림과 같이 크게 4가지 부분으로 구분됩니다. 우선 ㉠은 초코파이 메인보드에 건전지나 충전지와 같은 외부 전원을 연결할 때 사용합니다. ㉡는 컴퓨터와 연결할 때 사용하는 USB 케이블을 꽂을 수 있습니다. ㉢는 초코파이 보드에서 제공하는 센서 블록이나 LED 블록 등 다양한 블록을 연결할 때 사용합니다. ㉣는 블루투스로 통신할 때 사용합니다.

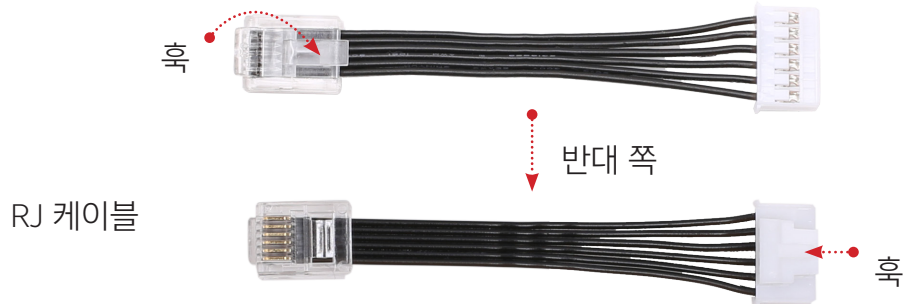


초코파이 메인보드는 아래 그림과 같이 건전지 홀더를 이용하여 1.5V 건전지 6개를 연결하여 작동시킬 수 있으며, 최대 12V까지 사용할 수 있습니다. 이때 건전지 홀더의 잭을 초코파이 메인보드의 ㉠에 연결하여 사용할 수 있습니다.





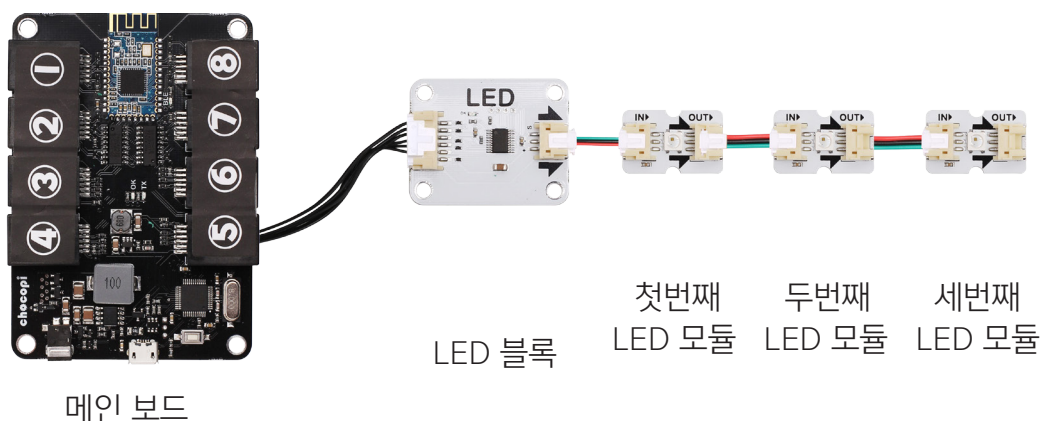
초코파이 메인보드와 센서 블록, LED 블록과 연결하려면 아래 그림과 같은 RJ 케이블을 활용해야 합니다. RJ 케이블은 전화기를 연결할 때 사용하는 잭과 비슷하게 생겼습니다. 초코파이 메인보드는 1~8번까지 총 8개의 숫자로 표기된 포트를 사용할 수 있습니다. 포트는 다른 장치와 연결하는 곳으로서 RJ 케이블을 연결하여 센서 블록과 연결할 수 있습니다.



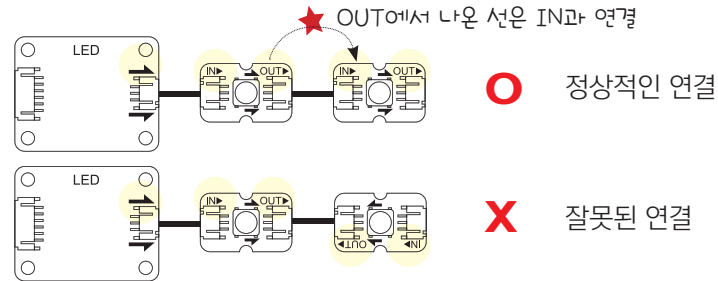
초코파이 보드에서 사용되는 케이블을 메인보드나 센서 블록, LED 모듈에서 뺄 때에는 흑을 눌러줘야 합니다. 흑을 누르지 않을 경우에는 잘 빠지지 않습니다. 무리하게 힘을 주면 선이 끊어질 수 있으므로, 반드시 흑을 누른 후에 빼야 합니다.

② LED 블록과 LED 모듈

LED 블록은 아래 그림과 같이 여러 가지 색의 불빛을 만들 수 있는 LED 모듈을 초코파이 메인보드와 연결할 때 사용합니다. LED 블록은 초코파이 메인보드의 1번부터 8번까지의 포트 중에서 어디든지 연결할 수 있습니다. LED 블록은 1개의 LED 모듈만을 연결할 수 있지만, 아래 그림과 같이 LED 모듈에 또 다른 LED 모듈을 연결하여 최대 256개의 LED 모듈을 연결할 수 있습니다. LED 블록에 LED 모듈을 연결할 때에는 반드시 IN과 OUT 표시를 확인해야 하며 LED 블록과 연결되는 LED 모듈은 IN과 연결되어야 합니다.



또한, LED 모듈과 LED 모듈이 연결될 때에도 IN과 OUT 표시를 확인해야 하는데, 아래 그림과 같이 반드시 LED 모듈의 OUT과 LED 모듈의 IN을 연결해야 합니다. 그렇지 않으면 정상적으로 작동되지 않습니다.



초코파이 보드에 포함된 LED 모듈은 빨간색(Red), 녹색(Green), 파란색(Blue) 등 3가지 색을 조합하여 160만 가지 이상의 다양한 색을 만들 수 있습니다. LED 블록과 LED 모듈에 실제로 불이 들어오게 하려면 엔트리에서 명령 블록을 조립하고 [시작하기]를 클릭해야 합니다.

LED 명령 블록은 다음과 같이 포트 번호, LED 모듈 번호, RGB 색상 값으로 구성됩니다.

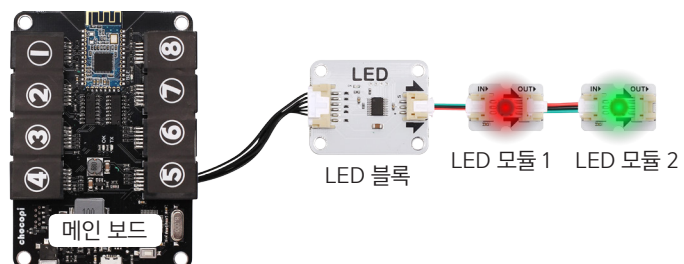
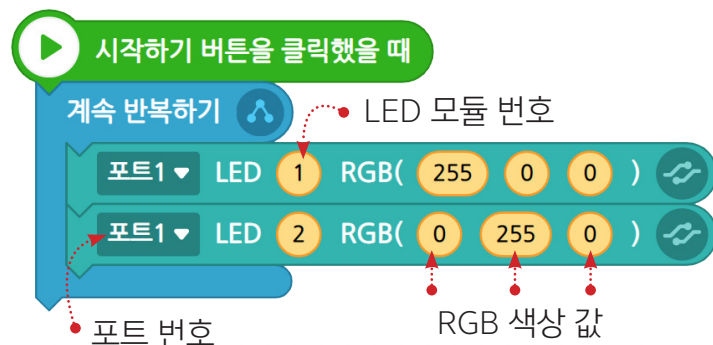
첫째, 앞 페이지의 그림에서 LED 블록이 초코파이 메인보드의 5번 포트에 연결되었기 때문에 포트 번호는 '포트 5'를 선택 합니다.

둘째, LED 모듈 번호는 LED 블록에 가장 가까이에 연결된 LED 모듈을 1번으로, 그 다음에 연결된 것을 2번으로 설정합니다.

셋째, RGB 색상 값은 3가지 값으로 구성되는데, 첫 번째 값은 빨간색, 두 번째 값은 녹색, 세 번째 값은 파란색을 의미합니다.

그리고 각각의 값은 0부터 255까지의 값을 가질 수 있습니다.

따라서 위에 제시된 명령 블록이 실행된다면, 첫 번째 LED 모듈은 빨간색으로, 두 번째 LED 모듈은 녹색으로 빛을 내게 됩니다.

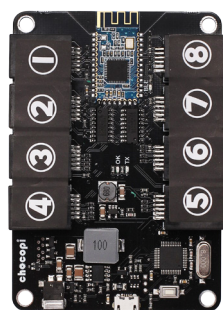




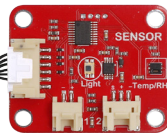
③ 센서 블록

센서 블록은 온도, 습도, 빛 센서가 포함된 블록입니다. 온도 센서는 -40도에서 125도까지 측정이 가능하고, 습도 센서는 0%에서 100%까지 측정이 가능하며, 빛 센서는 0부터 4095까지의 밝기를 측정할 수 있습니다. 빛센서는 가장 어두울 때 0, 밝을수록 4095에 가까워집니다.

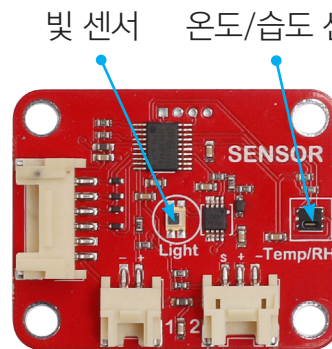
센서 블록을 이용하면 환경 변화에 따라 다양한 생활 용품을 만들 수 있습니다. 예를 들면, 온도 센서의 값이 30도보다 높아지면 선풍기를 돌린다든지, 빛 센서의 값이 100보다 작으면 LED 모듈에 불이 켜지게 한다든지, 온도와 밝기, 습도와 같은 환경 변화에 따라 다양한 프로그램이 가능합니다.



메인 보드



센서 블록



빛 센서 온도/습도 센서

센서 블록

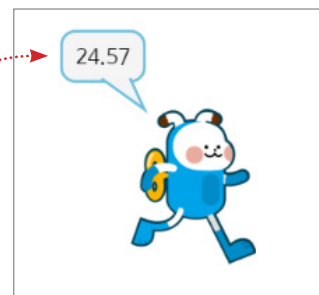
센서 블록의 값을 읽어오려면 엔트리를 이용하여 아래 그림과 같이 센서 명령 블록을 사용해야 합니다. 센서 명령 블록은 포트 번호와 센서 종류로 구분됩니다.

첫째, 포트 번호는 위 그림과 같이 초코파이 메인보드의 5번 포트에 센서 블록이 연결될 경우 '포트5'를 선택합니다.

둘째, 온도 센서의 값을 읽을 때에는 '온도'를 선택하고, 습도 센서의 값을 읽을 때에는 '습도'를 선택하며, 빛 센서의 값을 읽을 때에는 '빛'을 선택합니다.

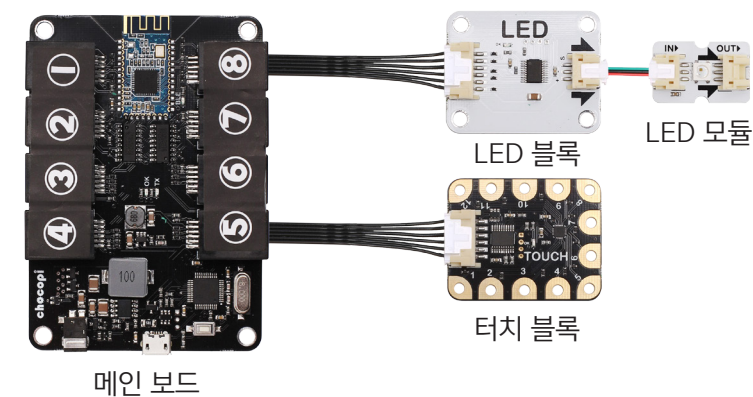


센서 블록을 연결한 후에 위에 제시한 명령 블록을 실행하면 엔트리봇이 현재의 온도를 4초 동안 말하게 됩니다.

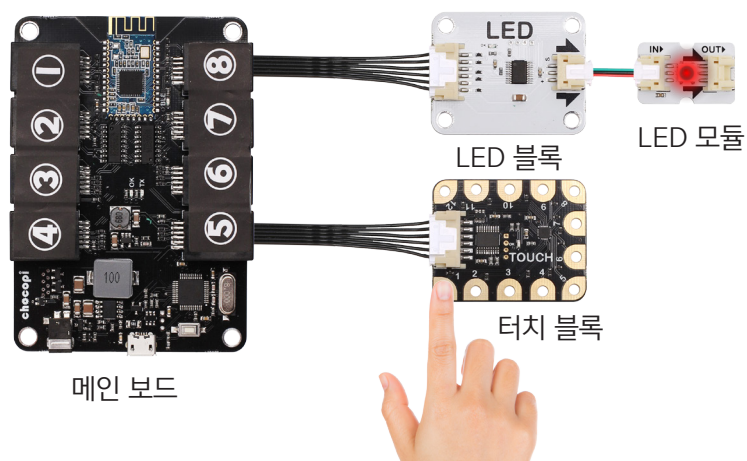


④ 터치 블록

터치 블록은 아래 그림과 같이 1번부터 12번까지의 12개 터치 센서로 구성되어 있습니다. 터치 센서에 물건이 닿으면 전류의 변화가 미세하게 발생하게 되는 데 이것을 감지할 수 있습니다. 따라서 각각의 터치 센서를 손으로 직접 터치하거나, 악어 집게나 전선 등을 연결하여 터치할 경우 그것을 감지하여 전자 피아노와 같은 여러 가지 생활 용품을 만들 수 있습니다.



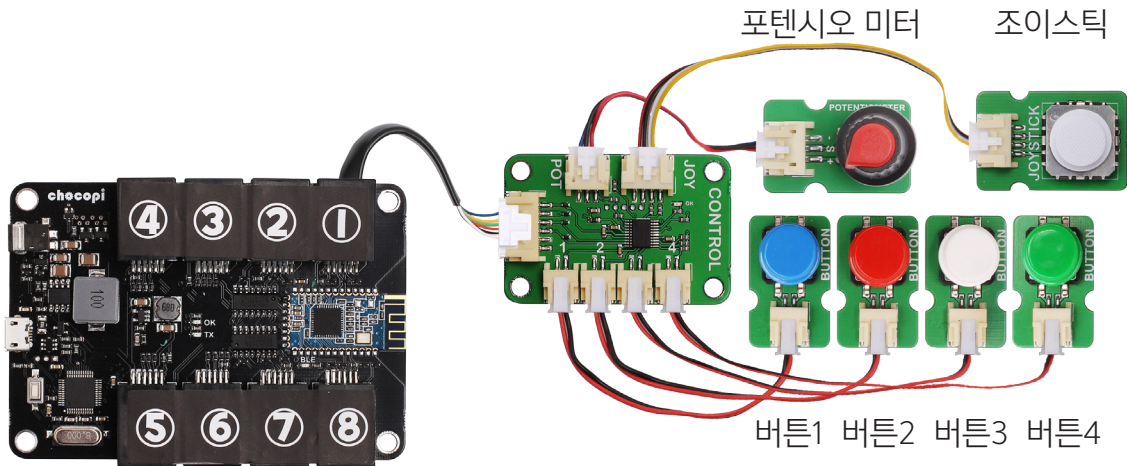
초코파이 보드를 연결하고 위와 같이 명령 블록을 조립한 후 실행하면, 초코파이 메인보드의 8번에 연결된 LED 블록의 첫 번째 LED 모듈에서 빨간색 불빛이켜집니다.



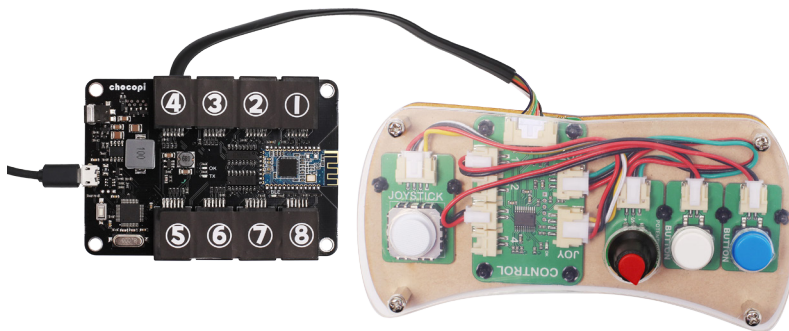


⑤ 컨트롤 블록

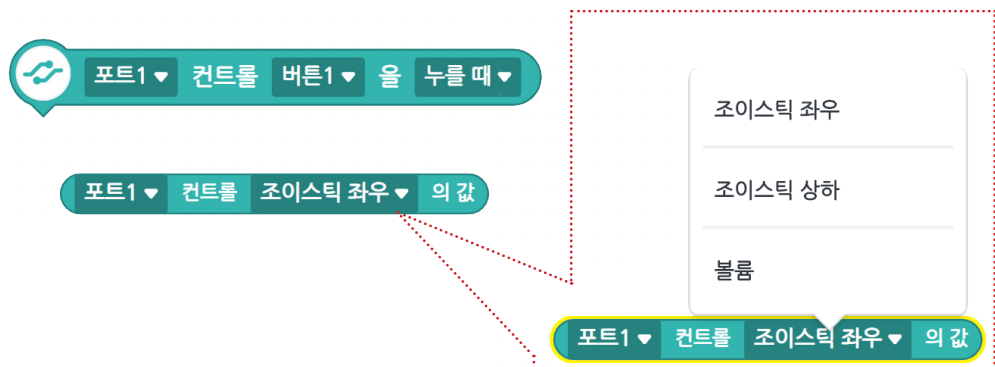
컨트롤 블록은 총 3가지 종류의 장치를 연결할 수 있는 블록입니다. 연결할 수 있는 장치는 조이스틱 1개, 포텐시오미터 1개, 버튼 4개 입니다.



컨트롤 블록으로 컨트롤러를 만들어 사용할 수 있습니다. 자동차, 로봇팔 등을 조정하고 게임할 때 사용합니다.

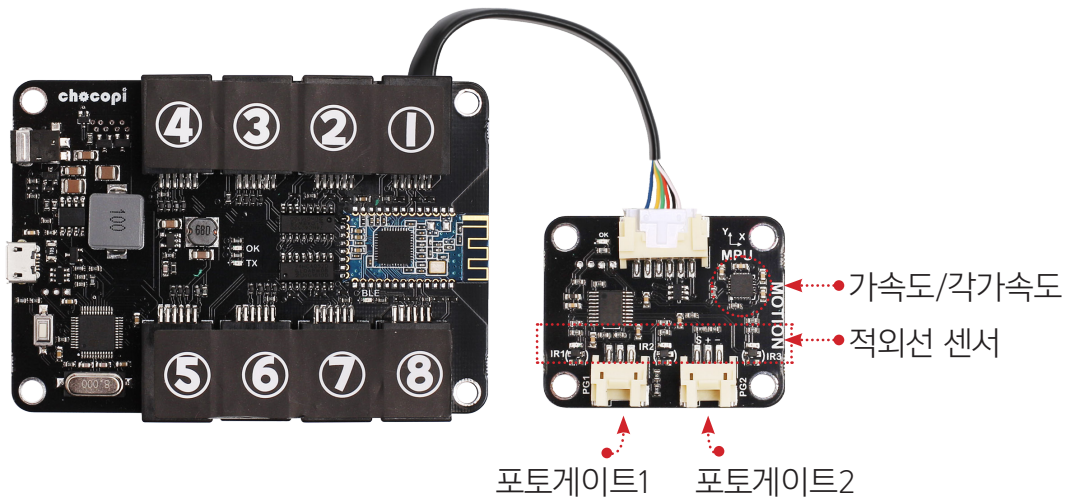


컨트롤 블록을 실행하는 명령 블록을 살펴보겠습니다.

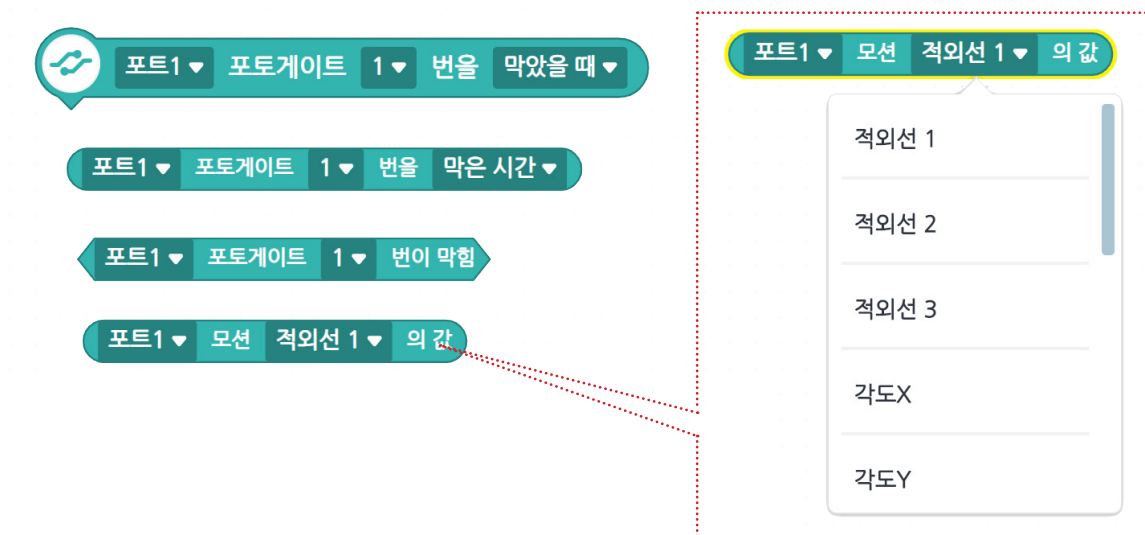


⑥ 모션 블록

- 모션 블록은 물체와의 거리를 감지하는 적외선 센서 3개, X,Y,Z 3축의 가속도 측정센서, U,V,W각가속 측정 센서가 내장되어있습니다.
- PG1, PG2 채널에 포토게이트 센서(별도판매)를 연결하면 게이트 사이를 지나는 물체를 감지할 수 있습니다.
- 라인트레이서 자동차, 자세 제어 로봇 등을 만드는데 사용합니다.



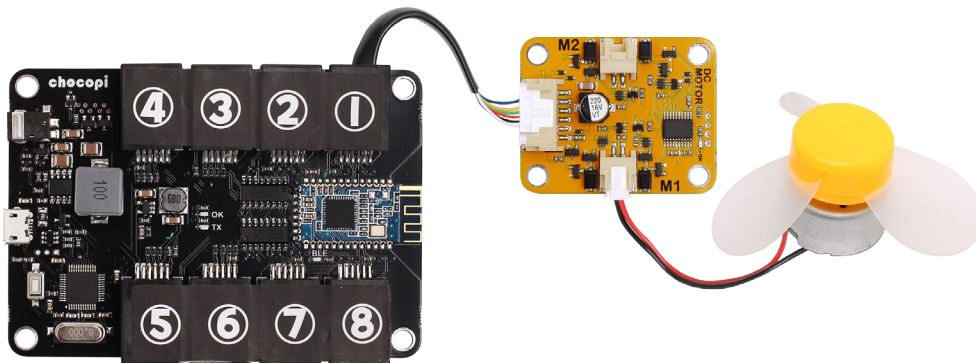
모션 블록을 실행하는 명령 블록을 살펴 보겠습니다.



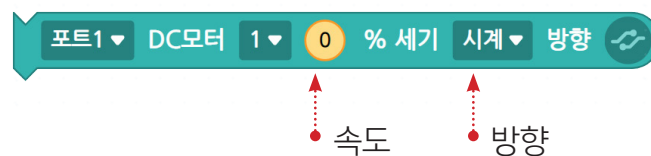


⑦ DC 모터 블록

- DC 모터 블록은 DC 모터 두 개의 속도, 방향을 제어합니다.
- 속도는 -100~100까지 조절할 수 있습니다.
- 자동차, 프로펠러, 팬 회전에 사용할 수 있습니다.



DC 모터 블록을 실행하는 명령 블록을 살펴 보겠습니다.



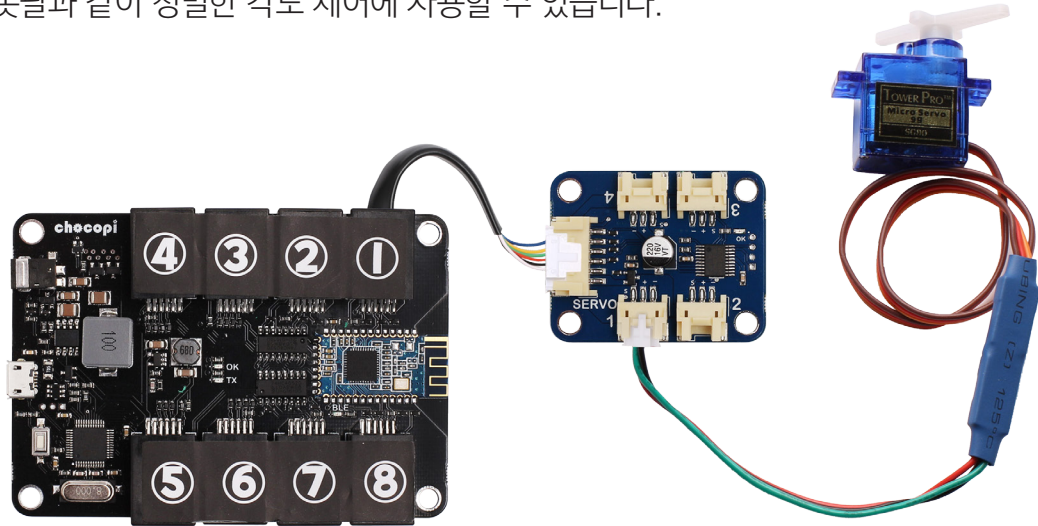
DC 모터 블록 사용 예시

초코파이 카는 두 개의 DC 모터로 작동됩니다.



⑧ 서보 모터 블록

- 서보 모터 블록은 서보 모터 네 개의 각도를 제어합니다.
- 로봇팔과 같이 정밀한 각도 제어에 사용할 수 있습니다.

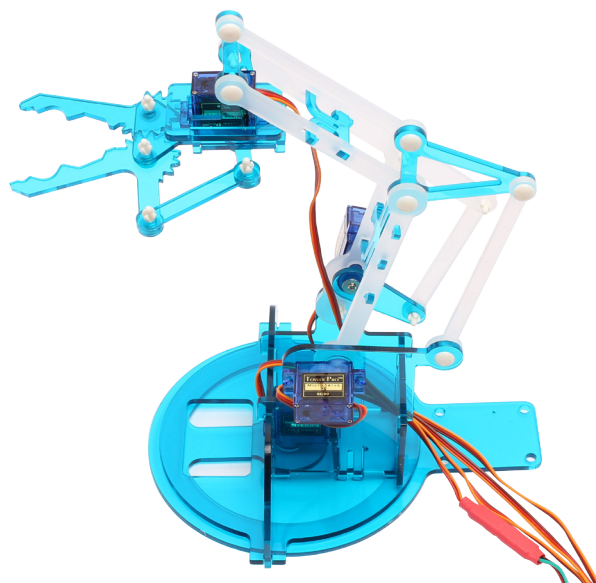


서보 모터 블록을 실행하는 명령 블록을 살펴보겠습니다.



서보 모터 블록 사용 예시

로봇팔은 네 개의 서보 모터로 작동됩니다.



부록 2

심심풀이 확장블록



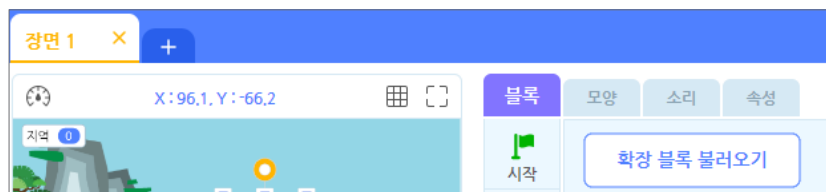


문제가 무엇인가요?

이번 챕터에서는 엔트리에서 제공하는 확장블록을 활용하여 화살표 키를 클릭하면 간단하게 최저기온, 최고기온, 습도 및 비가 올 확률을 말해주는 프로그램을 만들어 보겠습니다.
먼저 오브젝트 추가하기를 눌러서 여러분이 원하는 배경을 추가합니다.
다음으로 확장블록 불러오기를 클릭합니다.



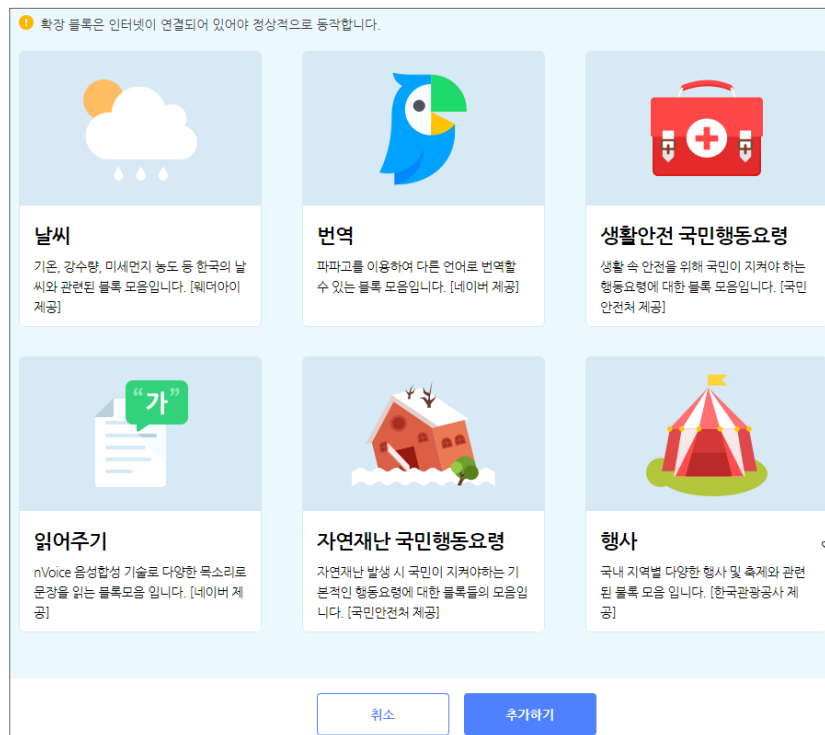
[그림] 배경을 추가하는 장면



[그림] 확장 블록 불러오기



확장블록 불러오기를 클릭하면 다음과 같은 다양한 블록들이 나타납니다.



[그림] 다양한 확장 블록들

확장블록 중에서 날씨 블록을 선택하여 불러오면 다음과 같이 지역별 온도, 습도, 강수확률 등을 알려 주는 다양한 블록들이 있는 것을 볼 수 있습니다.



[그림] 날씨 확장블록의 다양한 기능들

그렇다면 불러온 블록들을 활용하여 화살표 아래쪽, 위쪽, 왼쪽, 오른쪽을 클릭하였을 때 각각 최저기온(화살표 아래쪽), 최고기온(화살표 위쪽), 습도(화살표 왼쪽), 강수확률(화살표 오른쪽)을 나타낼 수 있도록 다음과 같이 프로그래밍하여 봅시다.

```

    when green flag clicked
    say 서울 지역의 왼쪽: 비가 올 확률, 아래쪽: 최저기온, 위쪽: 최고기온 for 2 sec

    when down arrow key pressed
    say 서울의 최저기온(도): 과(와) 오늘 서울 전체 의 최저기온(c) for 2 sec

    when up arrow key pressed
    say 서울의 최고기온(도): 과(와) 오늘 서울 전체 의 최고기온(c) for 2 sec

    when left arrow key pressed
    say 서울의 습도(%) 과(와) 오늘 서울 전체 의 습도(%) for 2 sec

    when right arrow key pressed
    say 서울의 비올확률(%) 과(와) 오늘 서울 전체 의 강수확률(%) for 2 sec
  
```

[그림] 화살표로 날씨를 알려주는 프로그램

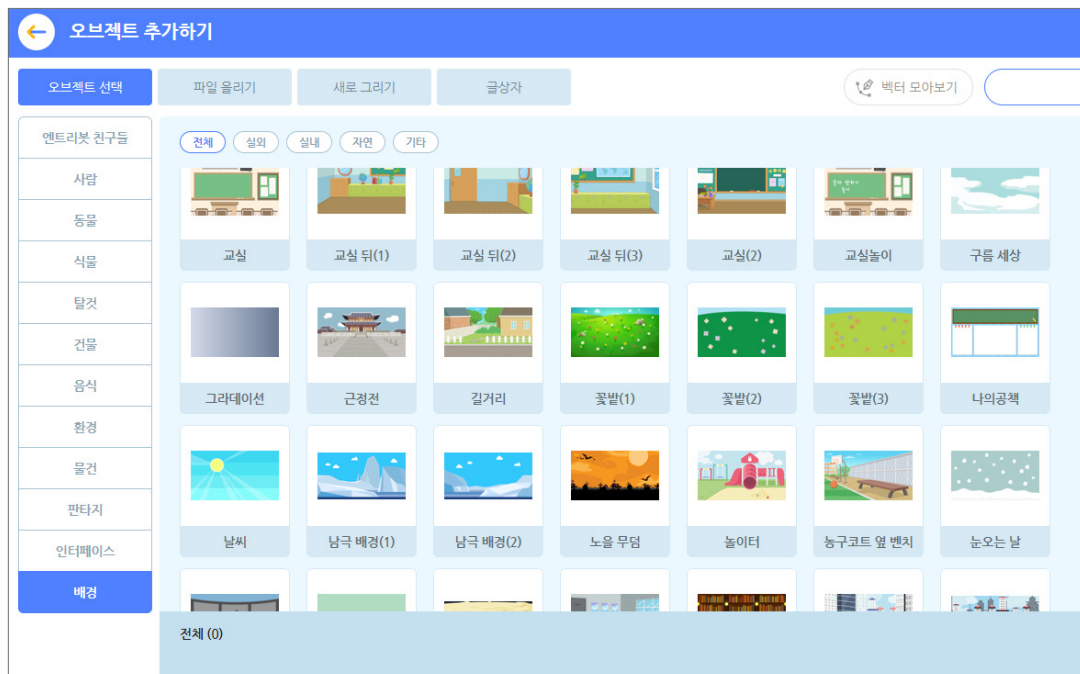


문제를 분해해봅시다!

여기에서 더 나아가서 온도에 따라서 배경이 바뀌는 프로그램을 만들어 봅시다.

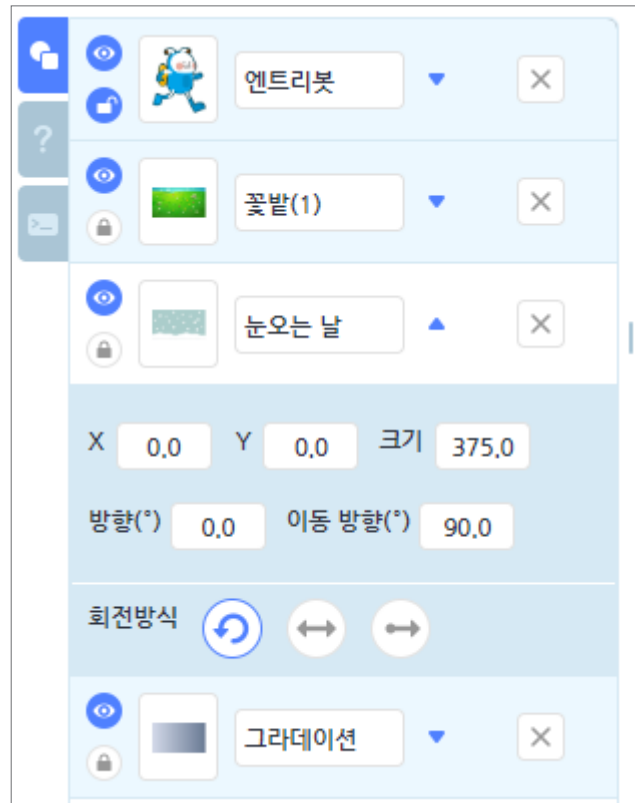
- 온도가 영상(0도 이상)이면 따뜻한 배경을 보여줍니다.
- 온도가 영하(0도 미만)이면 차가운 배경을 보여줍니다.

꽃밭 배경, 눈오는 날 배경 및 그라데이션 배경을 추가합니다.



[그림] 배경 추가하기

추가된 배경은 다음과 같은 순서대로 정리합니다.



[그림] 배경 정리

엔트리봇 오브젝트에 다음과 같은 블록을 프로그래밍합니다.



[그림] 엔트리봇 블록 프로그래밍



서울의 전체 기온이 10도 이상이라면 따뜻한 배경 바꾸기 신호를 보내고, 10도 미만이라면 차가운 배경바꾸기 신호를 보냅니다.

다음으로 꽃밭 배경에서는 다음과 같이 프로그래밍 합니다.



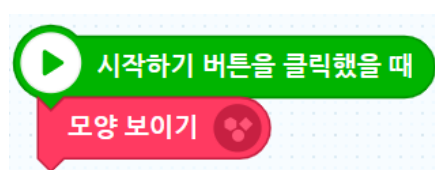
[그림] 꽃밭 배경 블록 프로그래밍

또한 눈오는 날 배경 블록 프로그래밍은 다음과 같이 진행합니다.

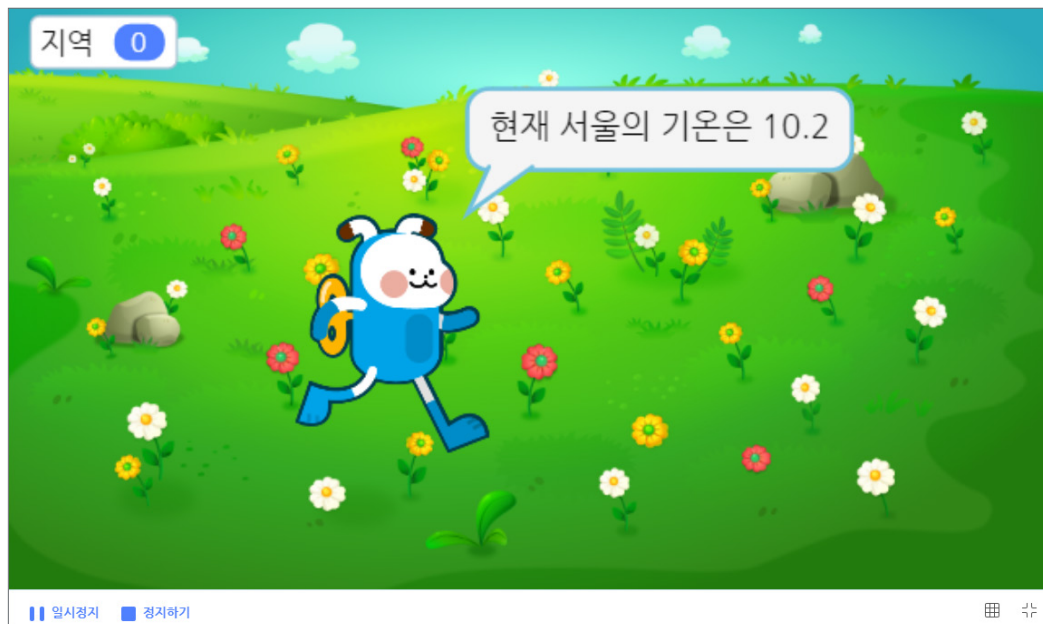


[그림] 눈오는 날 배경 블록 프로그래밍

그라데이션 배경 블록은 다음과 같이 프로그래밍 합니다.



프로그램을 실행하면 다음과 같이 나타납니다. 서울 전체 기온이 10도보다 높을 경우 배경이 꽃밭으로 변경됩니다. 만약 10도 미만일 경우는 눈오는 날 배경으로 바뀌겠지요.



[그림] 프로그램 실행 화면

여러분도 보다 다양한 엔트리 확장블록들을 활용하여 프로그래밍해 보시기 바랍니다.

소스 코드 다운 받는 곳

(주)한국과학 초코파이보드 홈페이지

<http://chocopi.org/download>

초코파이보드로 만드는 엔트리 코딩비법

발행일 : 2019년 12월 27일

저 자 : 성영훈

발행인 : 조원득

발행처 : 주식회사 한국과학

주소 : 서울특별시 양천구 국회대로 56 (신월동)

전화 : 02-929-1110

팩스 : 02-929-0966

웹사이트 : www.chocopi.org

ISBN : 979-11-967462-1-6



초코파이 보드로 만드는 엔트리코딩



9 791196 746216

ISBN 979-11-967462-1-6

(EPUB2)