

IoT기반 옥외놀이터 모니터링 및 안전관리 시스템

조심또조심하조팀
1615008 김소림
1615079 현재정
1771047 이효정

목차

프로젝트명- 이름선정 이유, 제안 동기

제안 배경, 목적- 필요성, 대상지의 현황, 주요 수혜자, 해결하고자 하는 문제점

구현과정 - 미끄럼틀, 그네, 통신, DB

기대 효과

진행상황

프로젝트명

실내가 아닌 실외 놀이터, 실내 놀이터는 이미
안전사고 관리 시스템이 잘 갖추어져 있음.
반면에 실외놀이터는 안전사고 관리 시스템이
마련되어 있지 않음

놀이기구에
센서 설치

놀이터를 이용하는
어린이가 주 대상

IoT기반 옥외 놀이터

모니터링 및 안전관리 시스템

실시간 모니터링, 데이터 수집을
통한 관리가 이 시스템의
첫번째 목적

모니터링을 통한 안전사고
예방이 이 시스템의 두번째
목적

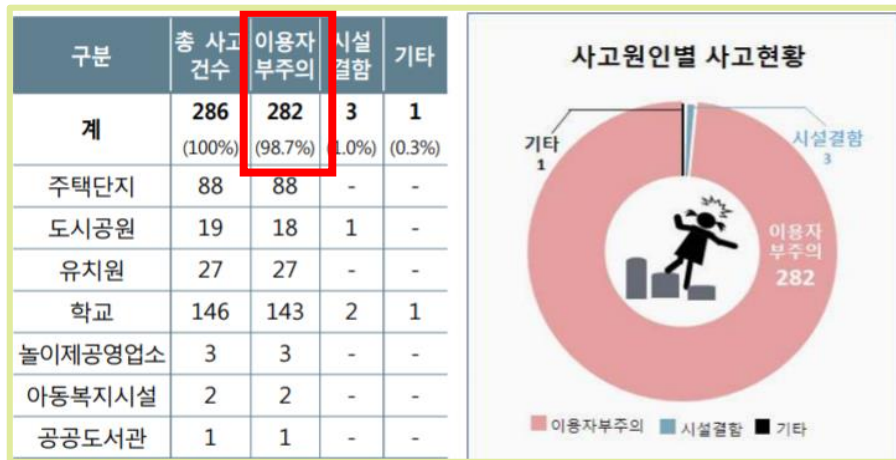
통합 시스템

제안배경-놀이터 안전사고 현황

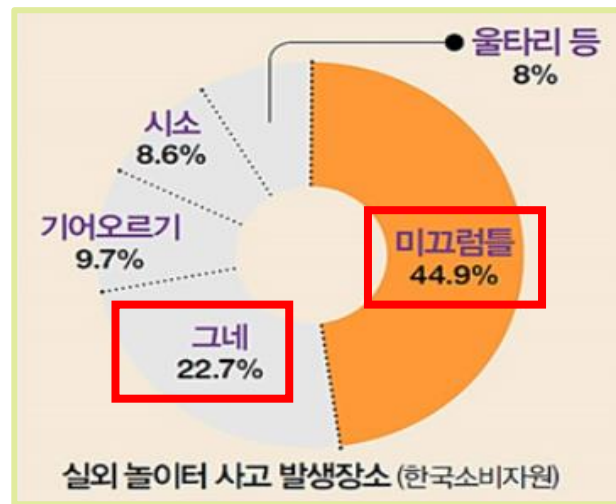
- 사고원인을 살펴보면, **이용자 부주의로 인한 사고가 282건(98.7%)**으로 가장 많았으며, 시설결함 3건(1.0%), 기타 1건(0.3%) 순으로 나타남

《 사고원인별 사고현황 》

(단위 : 건)



실외 놀이터 사고 발생장소는 **미끄럼틀과 그네가 67.6%**를 차지



* 자료 : 행정안전부 - 2017년 어린이놀이시설 안전사고 분석 결과

제안배경 - 주요 수혜자



사고 연령대별로 살펴보면, 활동량이 많은
학령기(7~14세) 어린이가 234명(79.6%) 으로,
취학전(1~6세) 어린이 57명(17.7%)보다 부상자 수가 약
4.1배 많음

출처, 행정안전부

학령기 어린이 (7~14세) 79.6%
+ 취학전 어린이(1~6세) 17.7%
= 97.3%

1세~14세 어린이가 사고의 97.3%를 차지함

세부 전망

KOSIS 국가통계포털
Korean Statistical Information Service

통계표

국내통계 국제·북한통계 쉽게 보는 통계 온라인간행

국내통계

통계목록 현 거주지 성, 연... ※ "관심주제설정"에서 특정 주제를 선택한 경우

1) 현 거주지 성, 연령 및 1년 전 거주지 유형별 인구(1세이상) - 시군구

자료경신일: 2020-08-28 / 수록기간: 년 2016 ~ 2019 / 자료문의처 : 042-481-3756(전수), 042-481-3735(표본)

일괄설정 항목 [7/7] 행정구역별(시군구...) 성별 [3/3] 연령별 [2/17]

(단위: 명) 새창보기 주석정보 주소정보 열람전환 분석 차

| 행정구역별(시군구) | 성별 | 연령별 | 2019 |
|------------|----|--------|------------|
| | | | 1년 전 거주지-계 |
| 전국 | 계 | 5~9세 | 2,251,867 |
| | | 10~14세 | 2,219,445 |
| | 남자 | 5~9세 | 1,156,453 |
| | | 10~14세 | 1,145,070 |
| | 여자 | 5~9세 | 1,095,415 |
| | | 10~14세 | 1,074,375 |

출처, KOSIS 통계자료 (2020)

놀이기구 사용연령대를 전국의 5세~14세로
가정한다면,

$2,251,867 + 2,219,445 = 4,471,312$ 명의 어린이로
잡을 수 있다.

또한 어린이의 부모(어린이 한명 당 2명)를
부수혜자로 잡는다면 대략 어린이의 부모인
 $4,471,312 * 2 = 8,942,624$ 명으로 볼 수 있다.

총 수혜자는
 $4,471,312 + 8,942,624 = 13,413,936$ 명이다.

주요 사고 내용

| | |
|------|---|
| 미끄럼틀 | <ul style="list-style-type: none"> • 미끄럼틀 원통 위로 오르다 실수로 추락하여 다리 골절상 • <u>미끄럼틀을 거꾸로 오르다 내려오던 아이와 충돌 및 추락하여 골절상</u> • 미끄럼틀 측면에 걸터앉아 타다 미끄럼틀 틀에 부딪쳐 치아손상 • 미끄럼틀 출발지점에서 다른 아이가 밀어서 추락하여 골절상 • 앉지 않고 서서 내려오다 바깥쪽으로 떨어지면서 팔에 골절상 • 도착지점 근처 노는 아이와 충돌 후 뒤로 넘어져 다리 인대 파열 • <u>미끄럼틀 타던 중 중간지점에서 뛰어내리다 팔 골절상</u> |
| 그네 | <ul style="list-style-type: none"> • 1인용 그네를 2명이 마주보며 타다가 무릎으로 상대방 아동 얼굴 타격 • 그네 주변 안전 바에서 회전놀이를 하다가 추락해 머리 골절상 • 그네 기둥으로 오르다 추락하여 팔 골절 • 그네를 꼬아 타다가 회전하며 얼굴이 그네 기둥과 충돌하여 골절상 • <u>10세 여아가 그네 옆을 지나가다 그네 타던 아이와 충돌해 골절상</u> |

위험한 행동 패턴에 대해

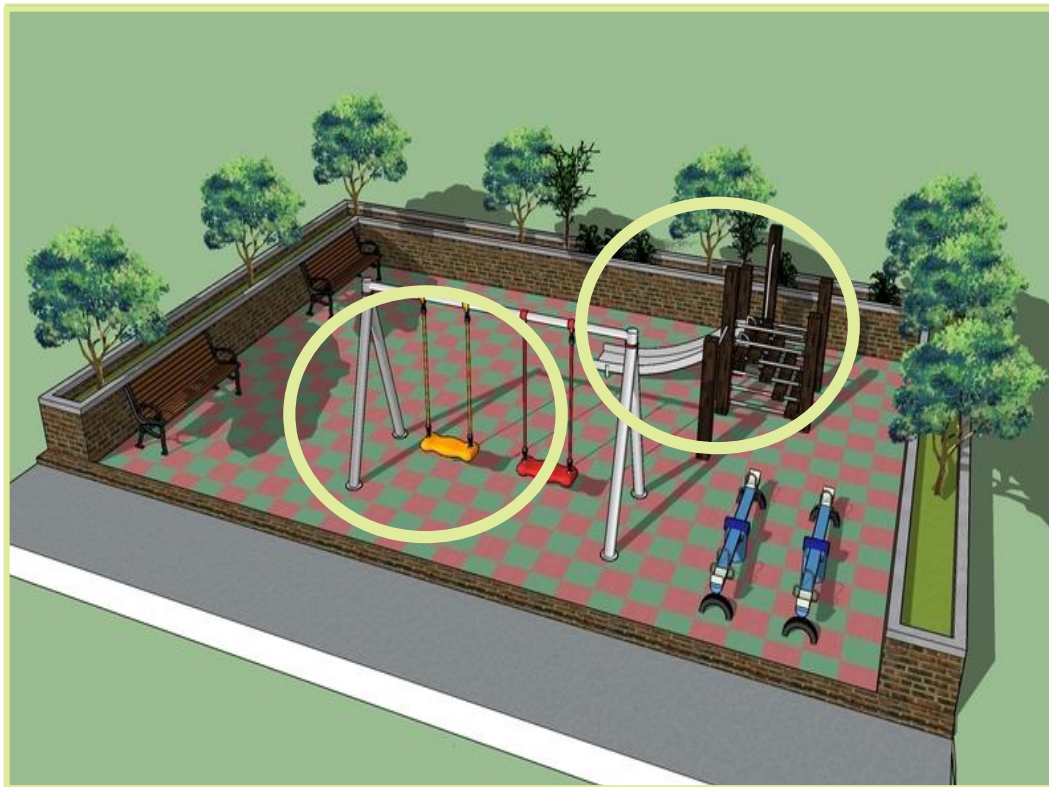
센서 측정으로

알람/경고를 통하여

위험을 인지하게 하여 사고예방

* 자료 : 행정안전부 - 2017년 어린이놀이시설 안전사고
분석 결과

시스템 구성도



Target

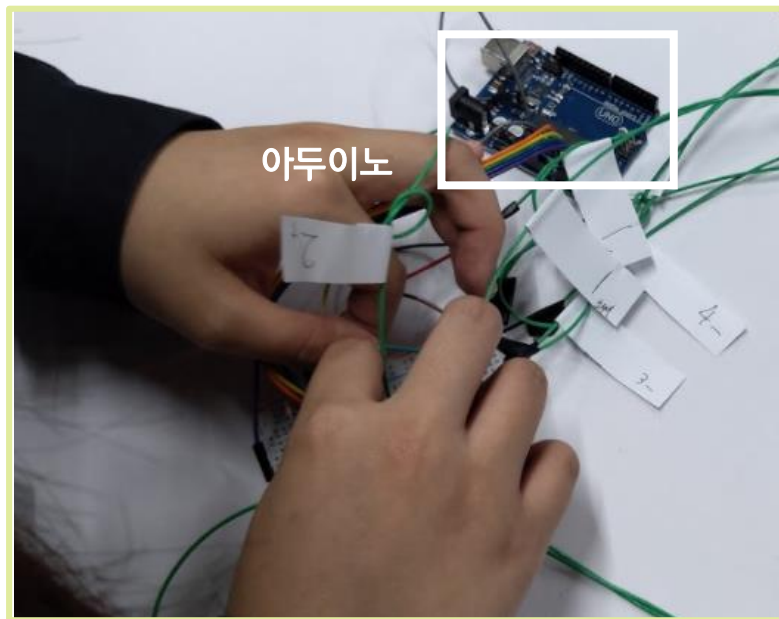
놀이기구: 미끄럼틀/그네

선정사유 : 대부분 놀이터에
있는 기본적인 놀이기구

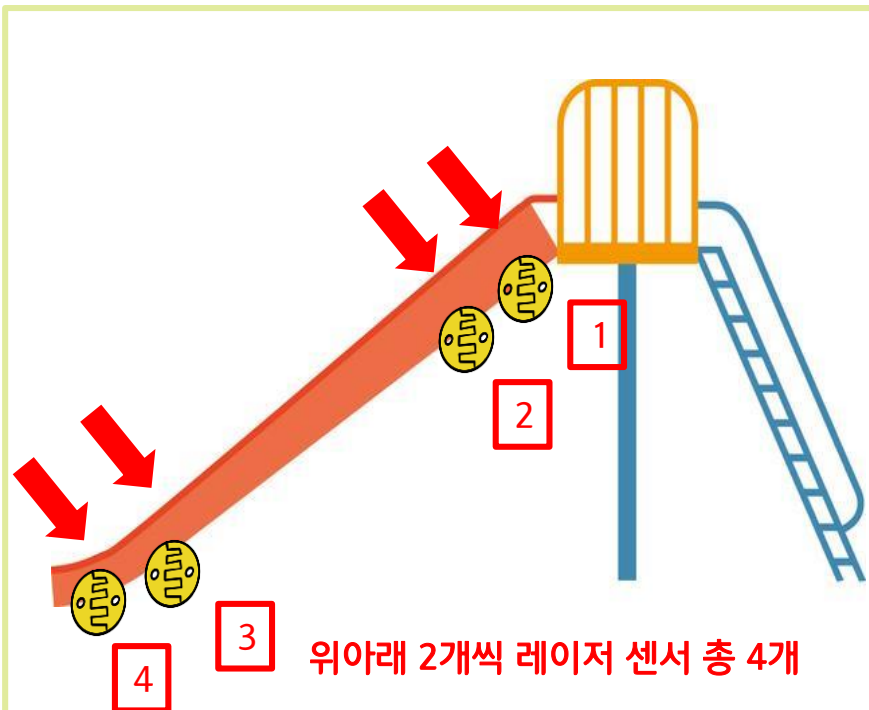
사용 센서 : 조도센서,
레이저(광)센서/트래킹 센서
통신 방식 : wifi(무선)
프로토콜 : UDP

언어: 아두이노 - C++
프로세싱 - JAVA
DB: SQLite

구현 과정 - 미끄럼틀



구현 과정 - 미끄럼틀



| 경우 | 상황 | 센서인식 순서 |
|----|--------------------|---------|
| 1 | 순서대로 내려옴 | 1234 |
| 2 | 위에서, 내려갔다 올라옴 | 1221 |
| 3 | 거꾸로 올라옴 | 4321 |
| 4 | 밑에서, 위로 올라갔다 내려옴 | 4334 |
| 5 | 순서대로 내려옴 & 중간에 머무름 | 12-34 |
| 6 | 거꾸로 올라옴 & 중간에 머무름 | 43-21 |
| 7 | 위, 아래 모두 사람이 있음 | - |

구현 과정 - 미끄럼틀

sketch_sep22a | 아두이노 1.8.13

파일 편집 스케치 툴 도움말

sketch_sep22a.s

```
// ButtonA
EasyButton buttonA(BUTTON_A_PIN);
// ButtonB
EasyButton buttonB(BUTTON_B_PIN);
// ButtonC
EasyButton buttonC(BUTTON_C_PIN);
// ButtonD
EasyButton buttonD(BUTTON_D_PIN);
```

```
long t1;
long t2;
long t3;
long t4;
```

```
// Callback function to be called when button1 is pressed
void onButton1Pressed()
{
  Serial.println("Button1 pressed");
}
```

```
// Callback function to be called when button is pressed
void onButtonAPressed()
{
  t1 = micros();
  //Serial.println("ButtonA pressed");
  //Serial.println(t1);
}
```

```
void onButtonBPressed()
{
  t2 = micros();
  //Serial.println("ButtonB pressed");
  //Serial.println(t2);
}
```

```
void onButtonCPressed()
{
  t3 = micros();
  //Serial.println("ButtonC pressed");
  //Serial.println(t3);
  //Serial.println(t2);
  float timeinterval1 = (float)(t3-t2)/(float)1000000;
  double timeinterval2 = (double)(t3-t2)/(double)1000000;
  Serial.println(timeinterval1,7);
  Serial.println(timeinterval2,7);
}
```

```
void setup()
{
  // Initialize Serial for debugging purposes
  Serial.begin(BAUDRATE);
}
```

```
Serial.println();
```

A
B
C
D
D
C
B
A

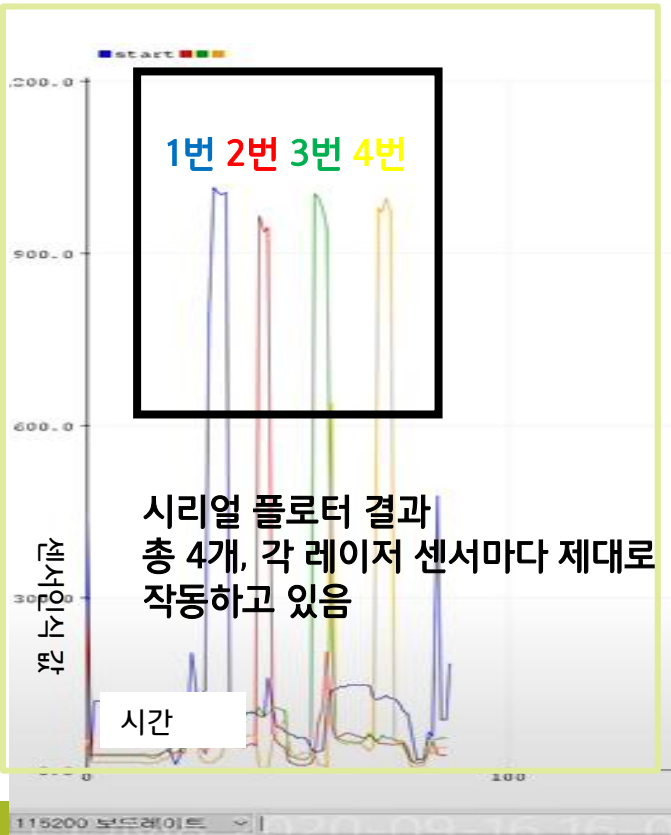
아두이노 코딩:
4개의 레이저 센서에 대한 선언

시리얼 모니터 결과)

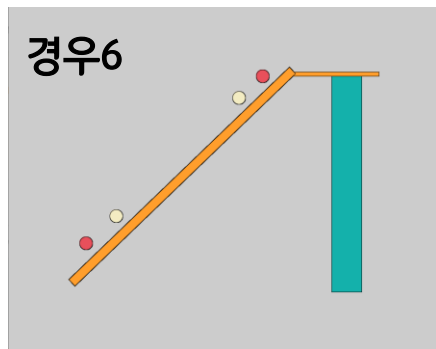
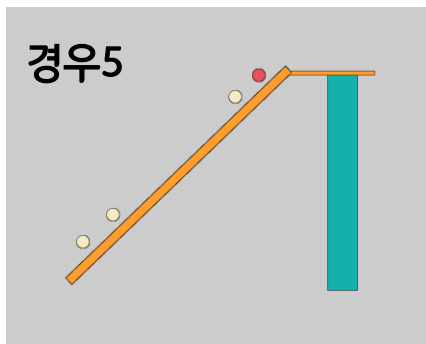
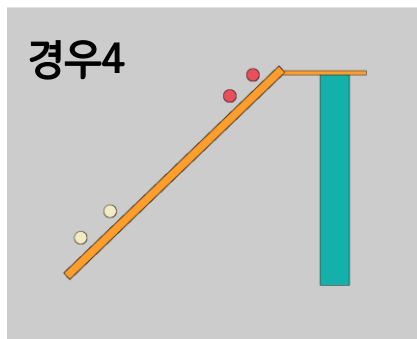
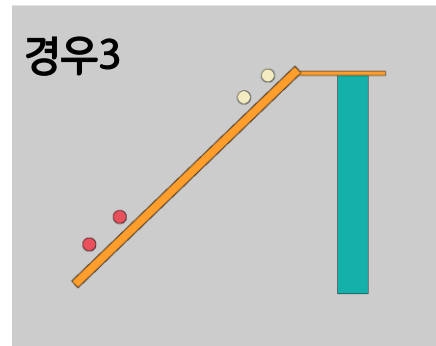
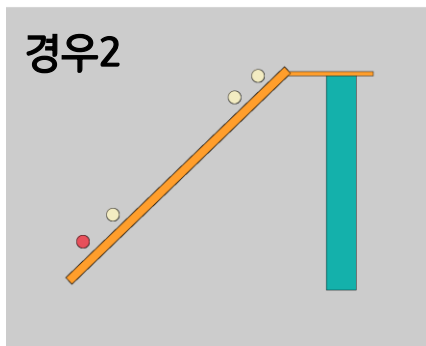
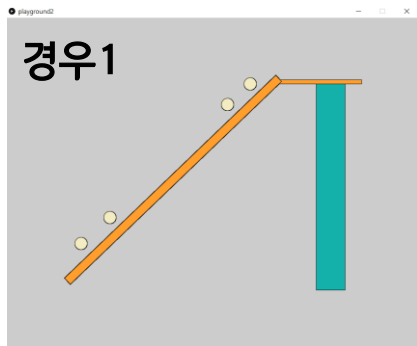
순서대로 내려오는 경우

거꾸로 올라오는 경우

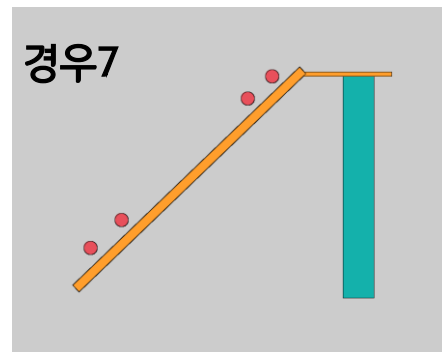
```
buttonA.read();
buttonB.read();
buttonC.read();
buttonD.read();
```



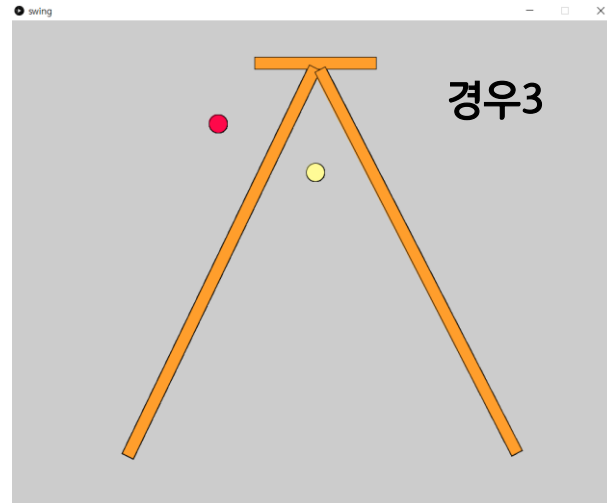
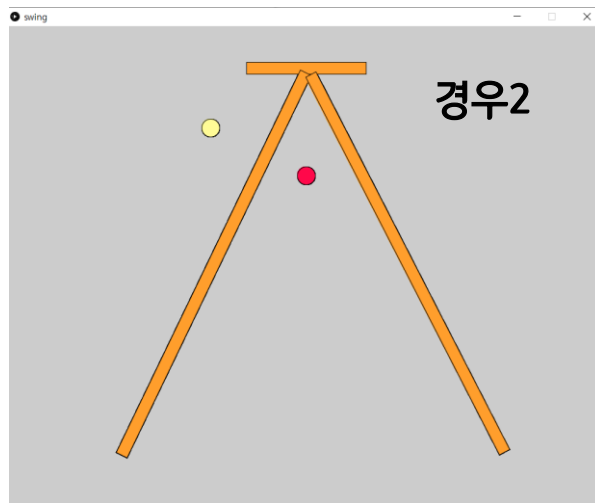
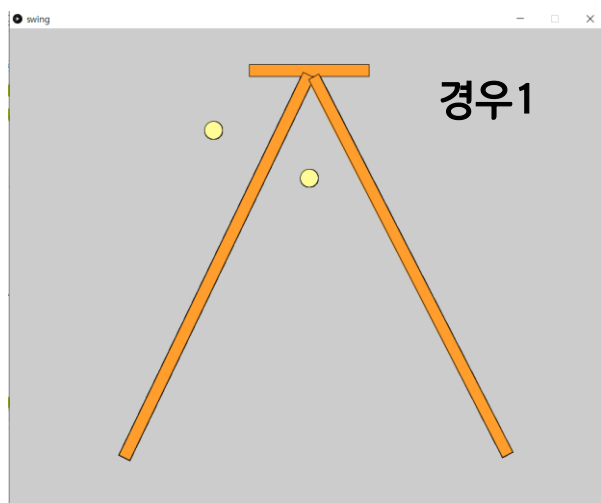
구현 과정 - 미끄럼틀



프로세싱에서
미끄럼틀 타는
경우의 수에 따른
관리자 화면 결과들



구현 과정 - 그네

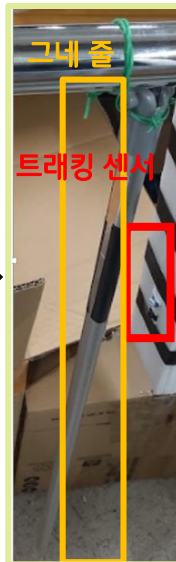


프로세싱에서 그네타는 경우의 수에 따른 관리자 화면 결과들

구현 과정 - 그네



구현 과정-그네



그네실험 모형
옆 모습

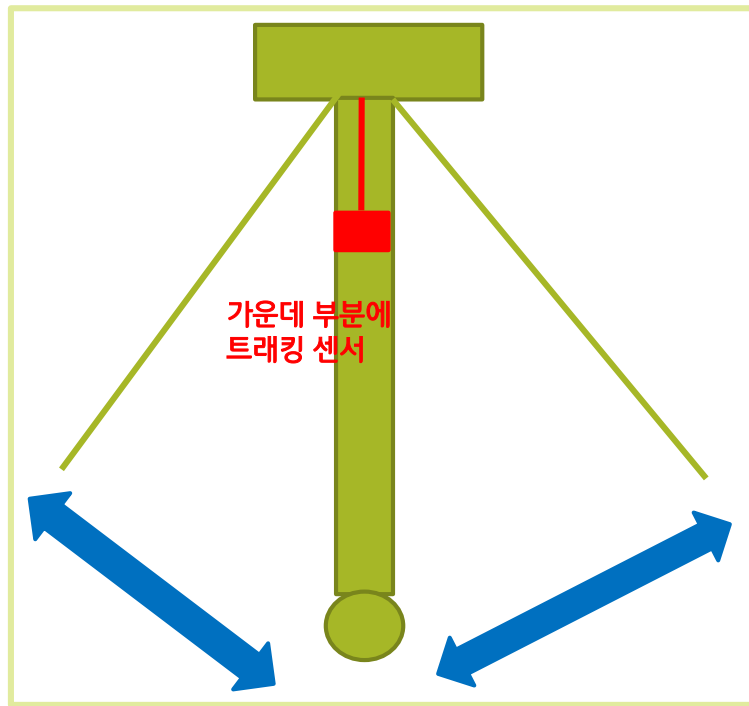


그네 옆 모습

구현 과정-그네

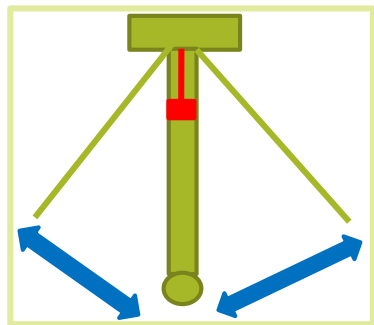


그네실험 모형 옆 모습



그네 옆 모습

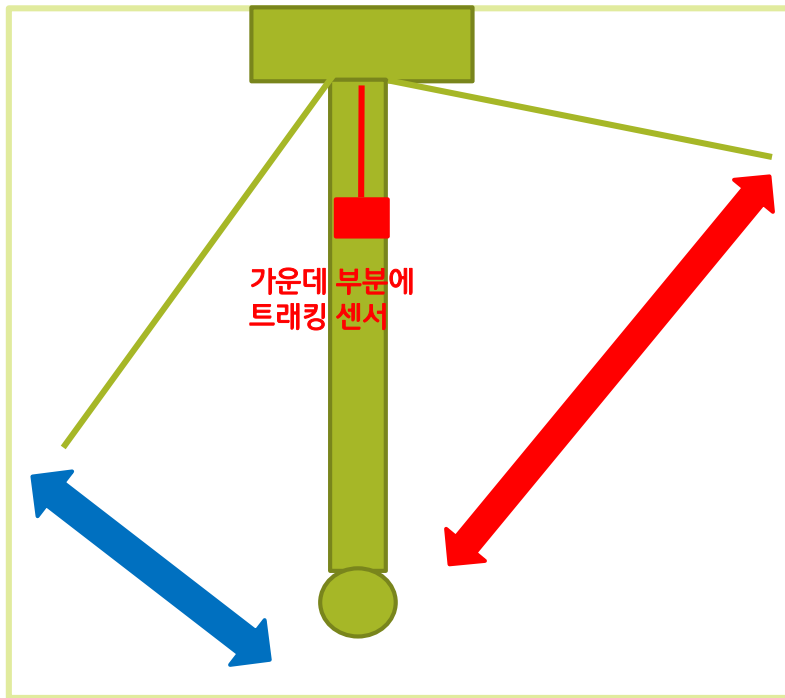
구현 과정-그네



구현 과정-그네

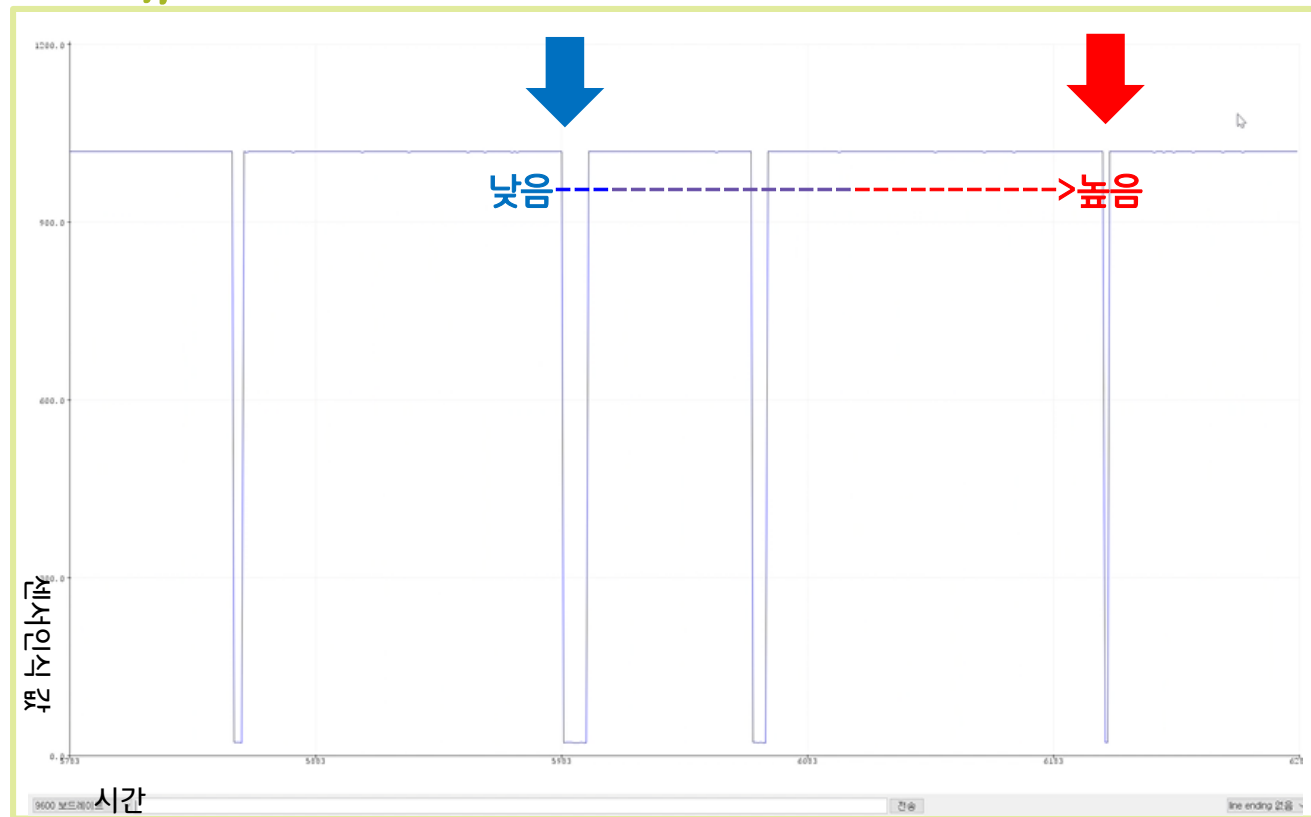
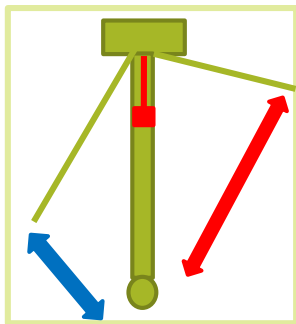


그네실험 모형 옆 모습

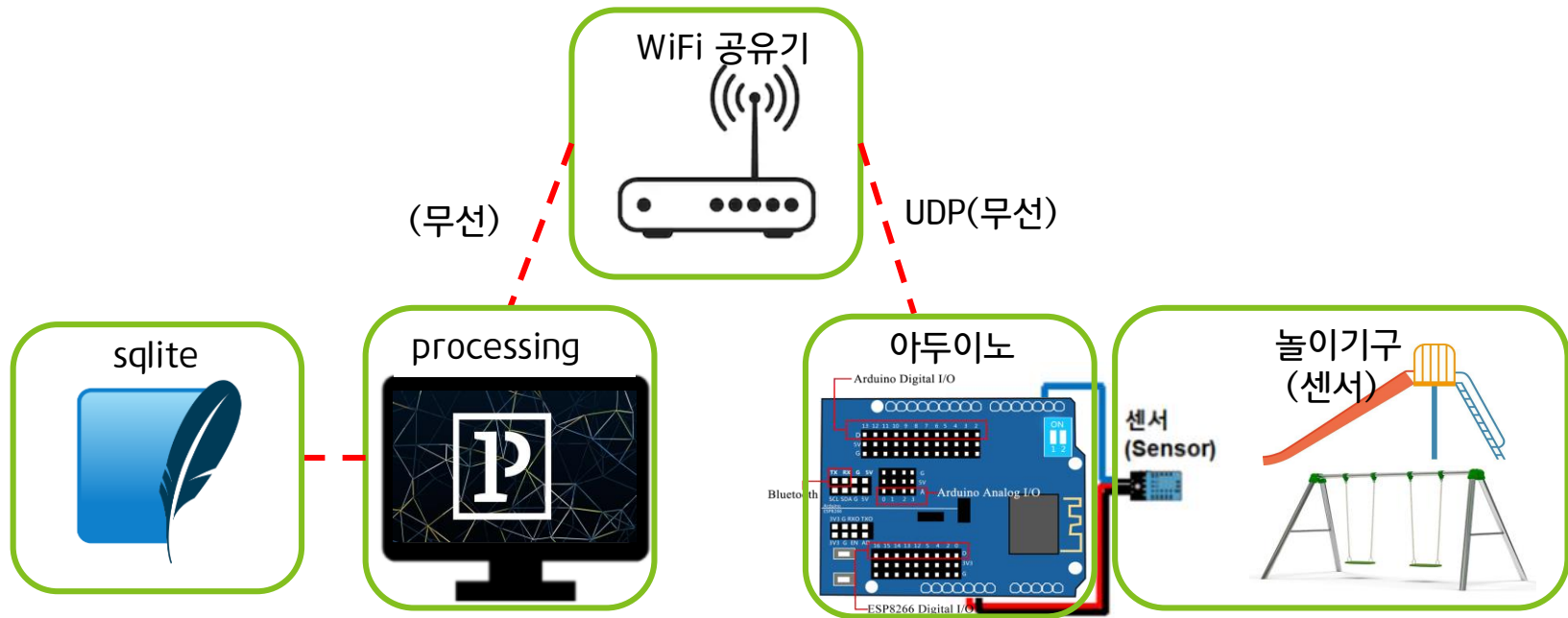


그네 옆 모습

구현 과정-그네



시스템 구성도



구현 과정-통신

```
receive: "testUDP" from 10.200.20.167 on port 8100
[20-09-22 12:57:33.734 +0900] receive packet <- from /10.200.20.167, port:8100, length: 113
receive: "[{"name":"slide0","value":0},{"name":"slide1","value":1},{"name":"slide2","value":2},{"name":"slide3","value":3}]" from
10.200.20.167 on port 8100
[20-09-22 12:57:38.740 +0900] receive packet <- from /10.200.20.167, port:8100, length: 113
receive: "[{"name":"slide0","value":0},{"name":"slide1","value":1},{"name":"slide2","value":2},{"name":"slide3","value":3}]" from
10.200.20.167 on port 8100
[20-09-22 12:57:43.739 +0900] receive packet <- from /10.200.20.167, port:8100, length: 113
receive: "[{"name":"slide0","value":0},{"name":"slide1","value":1},{"name":"slide2","value":2},{"name":"slide3","value":3}]" from
10.200.20.167 on port 8100
[20-09-22 12:57:48.739 +0900] receive packet <- from /10.200.20.167, port:8100, length: 113
receive: "[{"name":"slide0","value":0},{"name":"slide1","value":1},{"name":"slide2","value":2},{"name":"slide3","value":3}]" from
10.200.20.167 on port 8100
[20-09-22 12:57:53.744 +0900] receive packet <- from /10.200.20.167, port:8100, length: 113
receive: "[{"name":"slide0","value":0},{"name":"slide1","value":1},{"name":"slide2","value":2},{"name":"slide3","value":3}]" from
10.200.20.167 on port 8100
```

UDP connection되고 있는 화면

구현 과정-DB

playground_slide

| | |
|---------------|---------|
| playground_id | Integer |
| slide_id | Integer |

playground

| | |
|---------------|---------|
| playground_id | Integer |
| name | varchar |

playground_swing

| | |
|---------------|---------|
| playground_id | Integer |
| swing_id | Integer |

slide

| | |
|-----------|----------|
| id | Integer |
| slides_id | Integer |
| case | Integer |
| startTime | DateTime |
| duration | double |
| day | Integer |

swing

| | |
|-----------|----------|
| id | Integer |
| swing_id | Integer |
| case | Integer |
| startTime | DateTime |
| duration | double |
| day | Integer |

구현 과정-DB

데이터베이스 구조 데이터 보기 Pragma 수정 SQL 실행

데이터베이스(T): slides

| | slides_id | case | startTime | duration | id | day |
|----|-----------|------|----------------------------|----------|-----|-----|
| | 필터 | 필터 | 필터 | 필터 | ... | ... |
| 1 | 0 | 0 | 2020-09-29-14-33-57.646492 | 0 | 1 | 2 |
| 2 | 0 | 0 | 2020-09-29-14-33-57.646492 | 0 | 2 | 2 |
| 3 | 0 | 0 | 2020-09-29-14-34-02.661425 | 0 | 3 | 2 |
| 4 | 0 | 0 | 2020-09-29-14-34-02.661425 | 0 | 4 | 2 |
| 5 | 0 | 0 | 2020-09-29-14-34-07.683313 | 0 | 5 | 2 |
| 6 | 0 | 0 | 2020-09-29-14-34-07.683313 | 0 | 6 | 2 |
| 7 | 0 | 0 | 2020-09-29-14-34-12.700174 | 0 | 7 | 2 |
| 8 | 0 | 0 | 2020-09-29-14-34-12.700174 | 0 | 8 | 2 |
| 9 | 0 | 0 | 2020-09-29-14-34-17.724894 | 0 | 9 | 2 |
| 10 | 0 | 0 | 2020-09-29-14-34-17.724894 | 0 | 10 | 2 |
| 11 | 0 | 0 | 2020-09-29-14-41-27.381833 | 0 | 11 | 2 |
| 12 | 0 | 0 | 2020-09-29-14-41-27.381833 | 0 | 12 | 2 |
| 13 | 0 | 0 | 2020-09-29-14-41-37.576594 | 0 | 13 | 2 |
| 14 | 0 | 0 | 2020-09-29-14-41-37.576594 | 0 | 14 | 2 |
| 15 | 0 | 0 | 2020-09-29-14-41-42.590526 | 0 | 15 | 2 |
| 16 | 0 | 0 | 2020-09-29-14-41-42.590526 | 0 | 16 | 2 |
| 17 | 0 | 0 | 2020-09-29-14-41-52.618363 | 0 | 17 | 2 |
| 18 | 0 | 0 | 2020-09-29-14-41-57.634113 | 0 | 18 | 2 |
| 19 | 0 | 0 | 2020-09-29-14-42-02.653102 | 0 | 19 | 2 |
| 20 | 0 | 0 | 2020-09-29-14-42-07.665423 | 0 | 20 | 2 |
| 21 | 0 | 0 | 2020-09-29-14-42-12.679285 | 0 | 21 | 2 |
| 22 | 0 | 0 | 2020-09-29-14-42-17.700462 | 0 | 22 | 2 |
| 23 | 0 | 0 | 2020-09-29-14-42-22.716910 | 0 | 23 | 2 |
| 24 | 0 | 0 | 2020-09-29-14-42-27.729262 | 0 | 24 | 2 |
| 25 | 0 | 0 | 2020-09-29-14-42-32.742299 | 0 | 25 | 2 |
| 26 | 0 | 0 | 2020-09-29-14-42-37.764783 | 0 | 26 | 2 |
| 27 | 0 | 0 | 2020-09-29-14-42-42.781083 | 0 | 27 | 2 |

미끄럼틀 DB

새 데이터베이스(N) 데이터베이스 열기(O) 급변경 사항 저장하기(W) 종료

데이터베이스 구조 데이터 보기 Pragma 수정 SQL 실행

데이터베이스(T): swing

| | swing_id | case | startTime | duration | id | day |
|----|----------|------|----------------------------|----------|-----|-----|
| | 필터 | 필터 | 필터 | 필터 | ... | ... |
| 1 | 0 | 0 | 2020-09-29-15-40-30.667327 | 0 | 1 | 2 |
| 2 | 0 | 0 | 2020-09-29-15-40-35.681747 | 0 | 2 | 2 |
| 3 | 0 | 0 | 2020-09-29-15-40-40.700358 | 0 | 3 | 2 |
| 4 | 0 | 0 | 2020-09-29-15-40-45.774618 | 0 | 4 | 2 |
| 5 | 0 | 0 | 2020-09-29-15-40-50.809051 | 0 | 5 | 2 |
| 6 | 0 | 0 | 2020-09-29-15-40-55.830057 | 0 | 6 | 2 |
| 7 | 0 | 0 | 2020-09-29-15-41-00.848118 | 0 | 7 | 2 |
| 8 | 0 | 0 | 2020-09-29-15-41-05.873388 | 0 | 8 | 2 |
| 9 | 0 | 0 | 2020-09-29-15-41-10.891699 | 0 | 9 | 2 |
| 10 | 0 | 0 | 2020-09-29-15-41-15.916694 | 0 | 10 | 2 |
| 11 | 0 | 0 | 2020-09-29-15-41-20.966060 | 0 | 11 | 2 |
| 12 | 0 | 0 | 2020-09-29-15-41-25.990844 | 0 | 12 | 2 |
| 13 | 0 | 0 | 2020-09-29-15-41-31.007535 | 0 | 13 | 2 |
| 14 | 0 | 0 | 2020-09-29-15-41-36.023254 | 0 | 14 | 2 |
| 15 | 0 | 0 | 2020-09-29-15-46-38.000908 | 0 | 15 | 2 |
| 16 | 0 | 0 | 2020-09-29-15-46-43.573802 | 0 | 16 | 2 |
| 17 | 0 | 0 | 2020-09-29-15-46-48.594431 | 0 | 17 | 2 |
| 18 | 0 | 0 | 2020-09-29-15-46-53.674793 | 0 | 18 | 2 |
| 19 | 0 | 0 | 2020-09-29-15-46-58.689772 | 0 | 19 | 2 |
| 20 | 0 | 0 | 2020-09-29-15-47-03.703033 | 0 | 20 | 2 |
| 21 | 0 | 0 | 2020-09-29-15-47-08.724793 | 0 | 21 | 2 |
| 22 | 0 | 0 | 2020-09-29-15-47-13.739046 | 0 | 22 | 2 |
| 23 | 0 | 0 | 2020-09-29-15-47-18.754092 | 0 | 23 | 2 |
| 24 | 0 | 0 | 2020-09-29-15-47-23.788680 | 0 | 24 | 2 |
| 25 | 0 | 0 | 2020-09-29-15-47-59.139202 | 0 | 25 | 2 |
| 26 | 0 | 0 | 2020-09-29-15-48-09.351233 | 0 | 26 | 2 |
| 27 | 0 | 0 | 2020-09-29-15-48-17.122815 | 0 | 27 | 2 |

그네 DB

기대효과

사고 상황을 예방하기 위한 알림을 주어 사고발생률을 줄일 수 있다.

보호자들이 안심하고 아이들을 놀이터로 내보낼 수 있다.

놀이터 이용에 대한 실시간 정보 등을 제공 한다.

사용자가 직접 놀이터에 가지 않고도 사용자에게 놀이터에 관한 정보를 미리 알려 주어 놀이터 이용에 도움을 줄 수 있다.

놀이터의 시간에 따른 이용여부, 이용유형 등의 정보들을 수집, 분석하여 통계를 낼 수 있다.

추후 이러한 데이터들을 활용하여 시스템을 확장시킬 수 있다.

역할분담

| 팀원 | 담당 영역 |
|-----|------------------------|
| 김소림 | processing, server, DB |
| 현재정 | JSON, 미끄럼틀과 그네, 제출자료작업 |
| 이효정 | 미끄럼틀, 그네 arduino |

진행상황, 추후진행

| 월 | 화 | 수 | 목 | 금 | 토 | 일 |
|------------------|------------|--------------------|----|---|--|---|
| | | | | 30 양천구 공모전 발표준비(온라인) | 31 양천구 공모전 발표준비(30일에 완성 못하면) | 1 |
| 2 양천구 공모전2차심사 | 3 | 4 | 5 | 6 SW창업경진대회 본선(수업시간) 계룡공모전준비(9일 17시 마감) | 7 | 8 0캡스톤디자인경진 대회 준비-3분이내 동영상 등 (11일 15시 제출마감) |
| 9 | 10 | 11캡스톤경진대회제 출 마감 | 12 | 13 | 14 | 15 |
| 16멘토링서류마감 | 17기술블로그 마감 | 18 | 19 | 20 | 21 | 22 |
| 23 | 24 포스터 | 25 | 26 | 27 최종발표(state diagram 추가) | 28 | 29 |

감사합니다.