

Computer Graphics

팀 프로젝트 최종 발표

Unity와 Arduino를 이용한 혼합현실 콘텐츠 제작

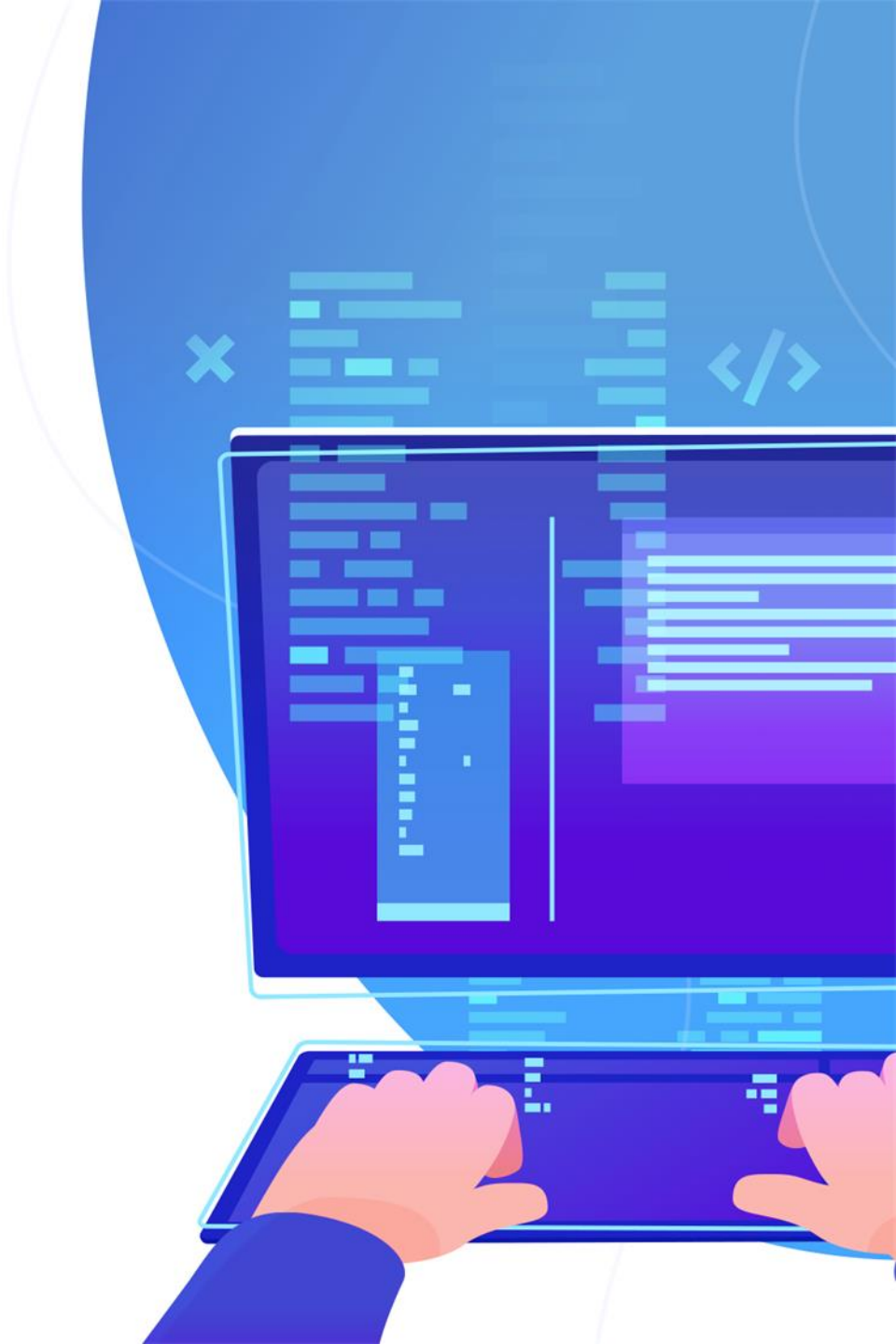
2019.12.19

201411691 김승수

201411711 류찬규 (팀장)

201411730 성봉제

201411742 원동찬



CONTENTS

Unity와 Arduino를 이용한 혼합현실 콘텐츠 제작

- 01 프로젝트 개요
- 02 Coin Dozer
- 03 설계
- 04 구현
- 05 최종 결과물 시연





Unity와 Arduino를 이용한 혼합현실 Contents 제작



프로젝트 목표

- 카메라를 통해 타겟 이미지를 인식하고 오브젝트 출력
- 오브젝트 텍스처 맵핑
- 오브젝트간의 상호작용
- 아두이노의 센서와 블루투스 통신 모듈을 이용해 현실세계와 게임 속 변화를 실시간으로 적용 (Mixed Reality 구현)



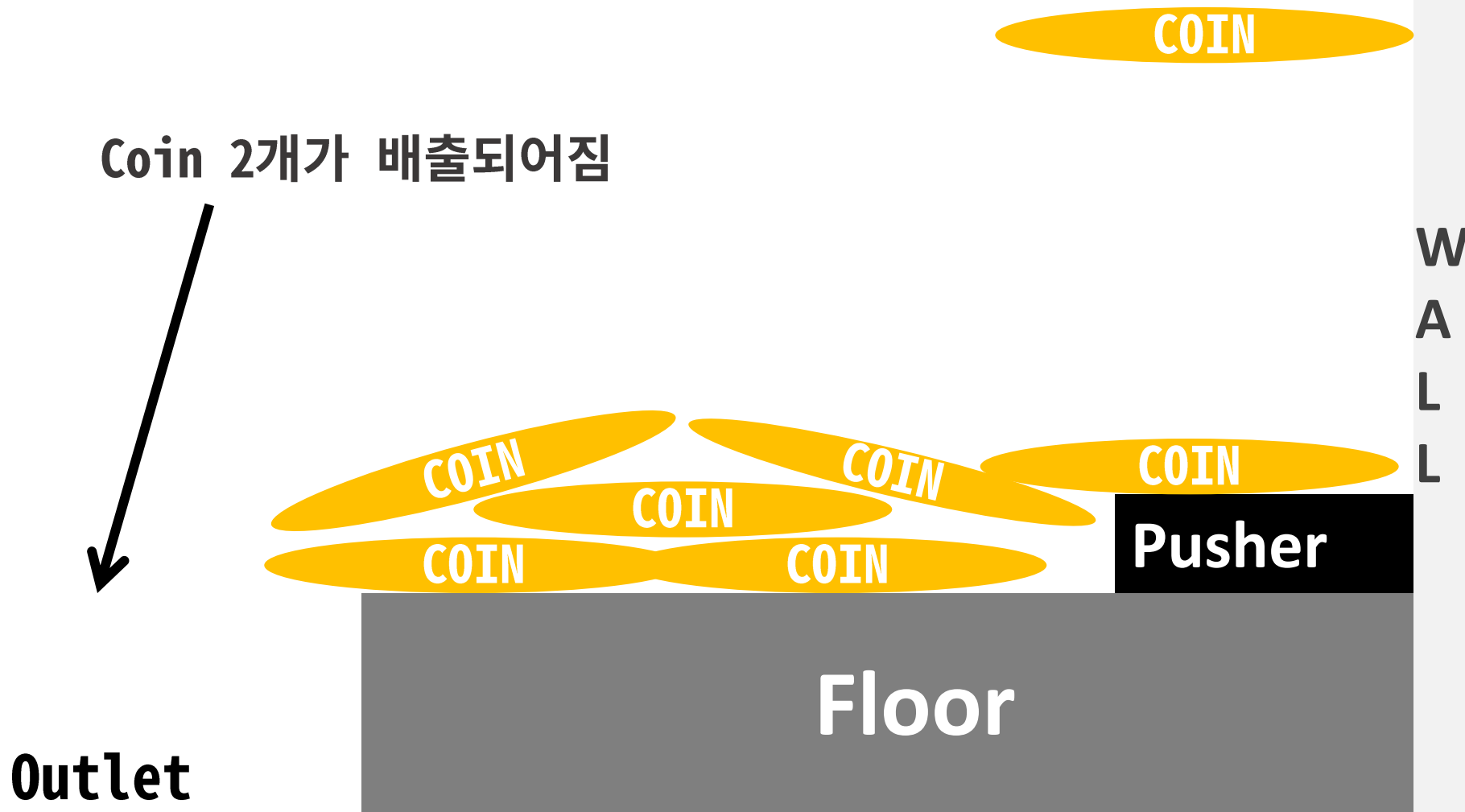
Coin Dozer 오락기

- 오락기에 동전을 넣으면 내부에 동전이 쌓임
- 오락기 내부 Pusher가 움직이면서 동전을 밀어냄
- Pusher에 밀린 동전은 배출구에 떨어지게 되고 그 돈은 게임 유저가 갖게 됨



설계 - 구조 및 동작 과정

1. User의 동전 투입으로 오락기 내부에 동전이 쌓임
2. 오락기 Pusher의 움직임에 따라 동전 배출





설계 - 동전이 투입되었을 때

4. 게임에서 동전 오브젝트가 추가 됨



Target Image

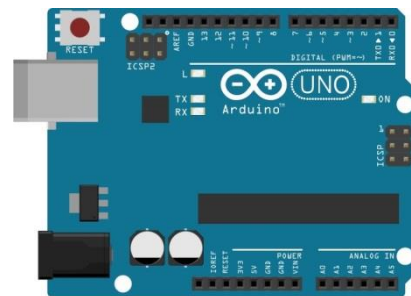


1. 동전 투입

2. 아두이노 센서 감지



3. 블루투스 모듈을 이용해
동전이 들어왔음을 전달





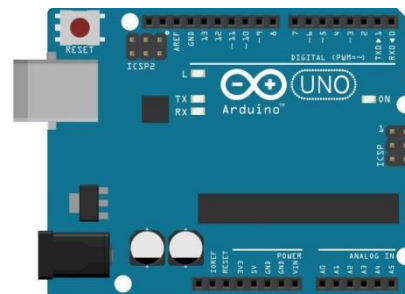
설계 - 동전이 배출 될 때

1. 게임에서 동전이 배출구에 빠짐

Target Image



2. 블루투스 모듈을 이용해 동전이 배출되어야 함을 전달



3. 서보모터를 이용해 동전 배출



4. 동전 배출



Unity

오브젝트 생성
텍스처 맵핑
AR 구현 (Vuforia)



Visual
Studio

오브젝트 Event
생성 및 움직임 제어
개발 언어 : C#



Android
Studio

Android Plugin
제작
개발 언어 : JAVA

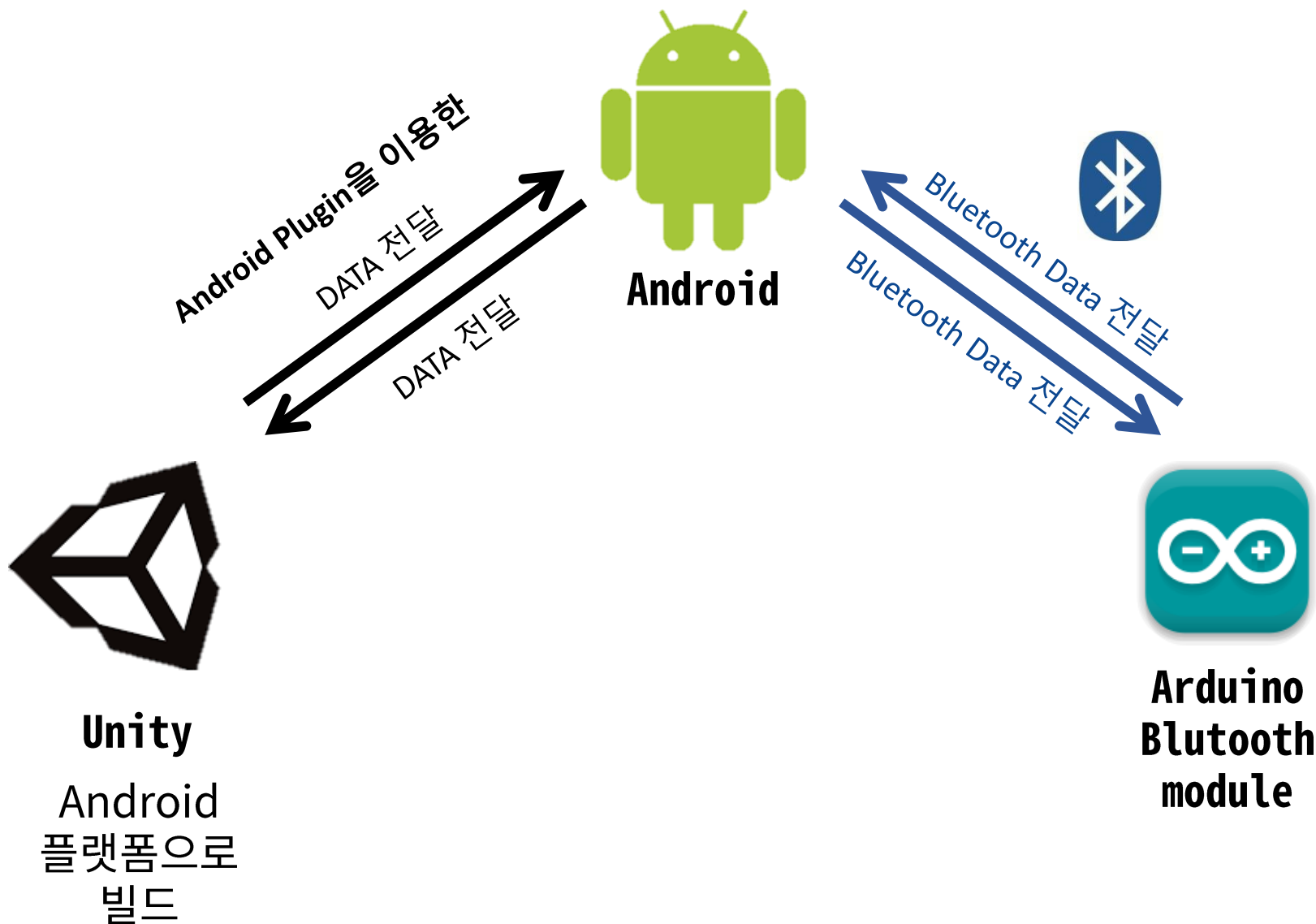


Arduino

블루투스 통신 제어부
및 H/W 동작 구현
개발 언어 : C++



Data 통신 과정





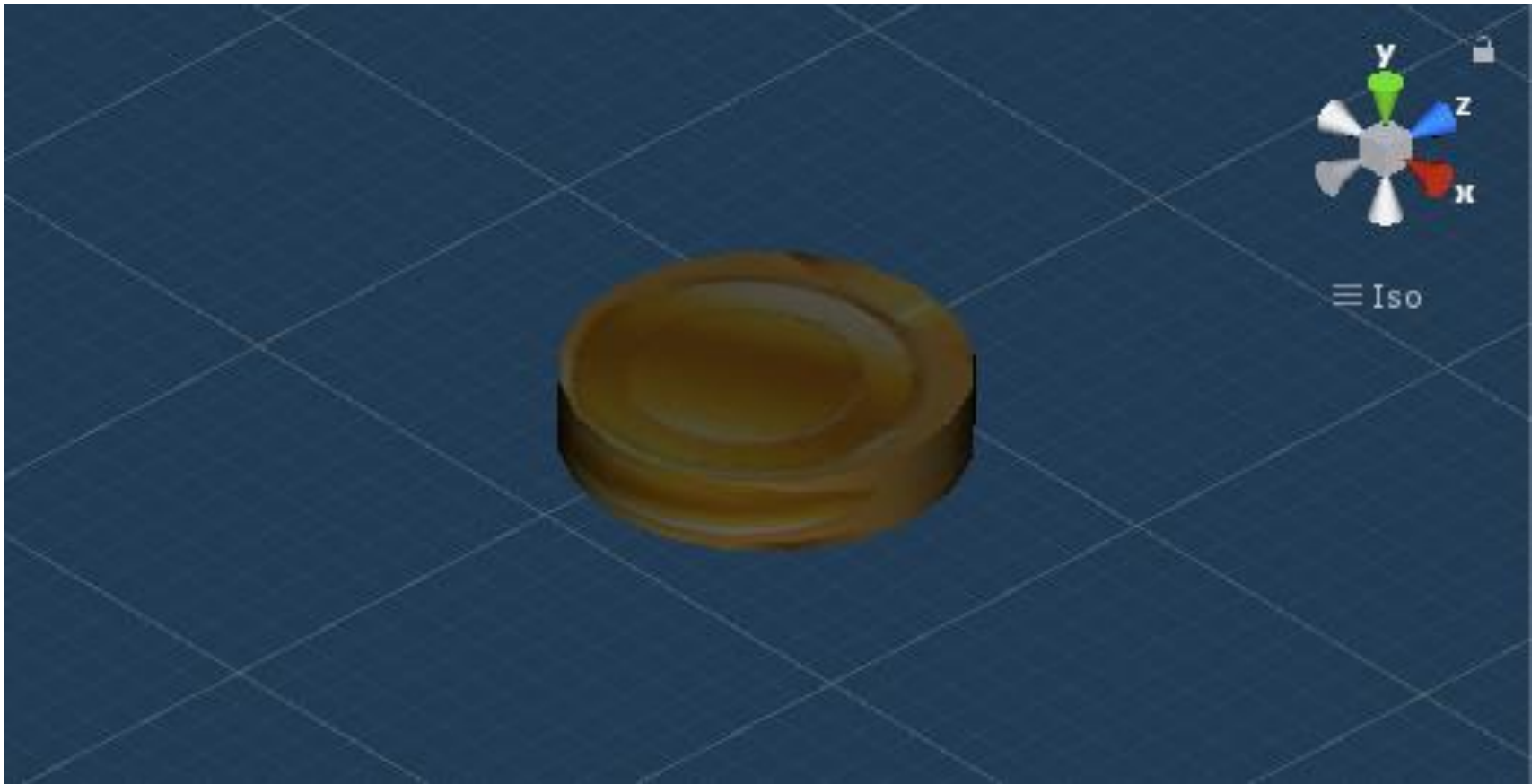
COMPUTER GRAPHICS

COIN

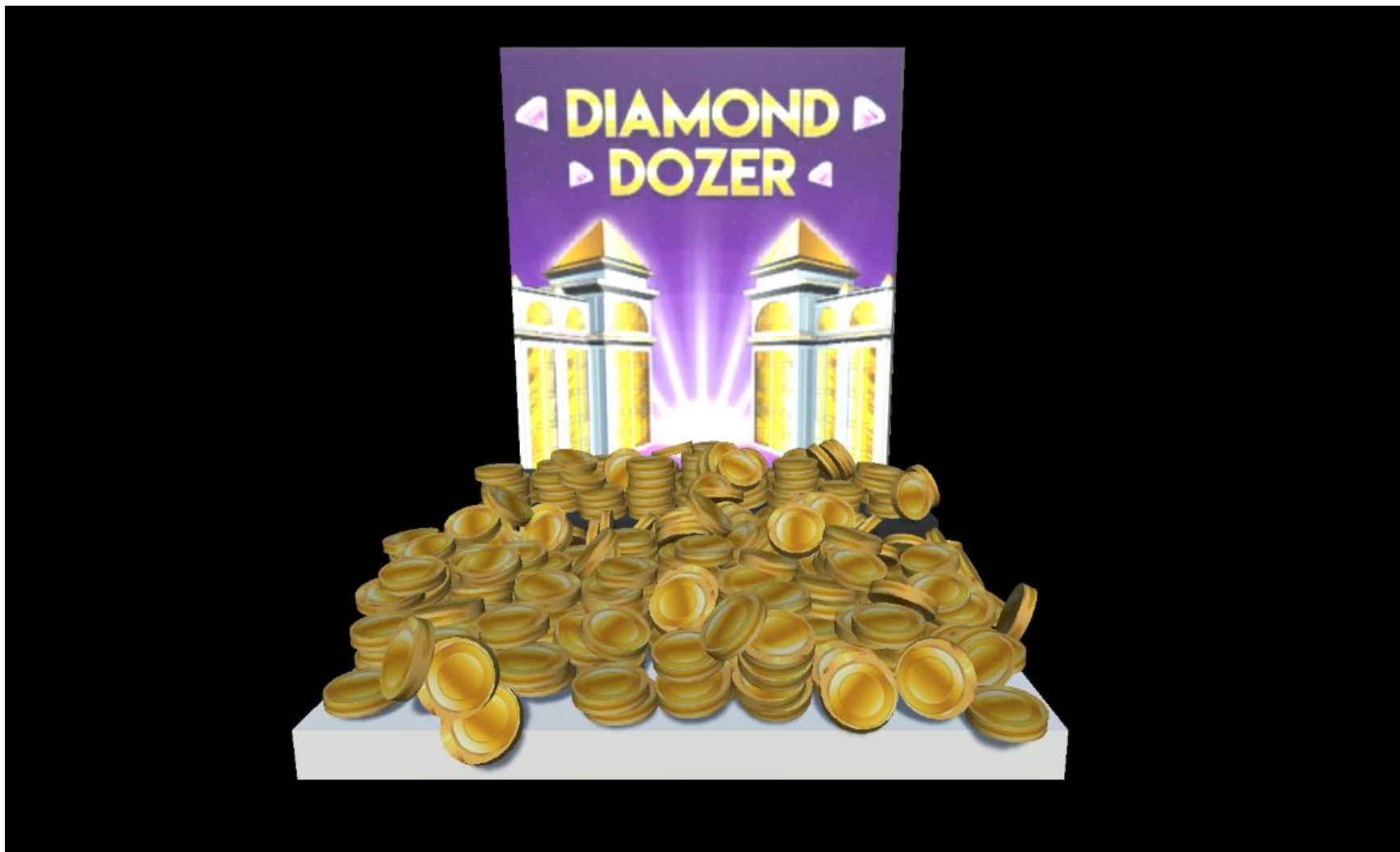
DOZER

사용할 Image Target

카메라에 이 Target이 포착되면
Target 위에 오브젝트들이 생성,
위치함



Coin Object 생성 및 텍스처 맵핑



Unity 내에 오브젝트 생성



Manager.cs



Pusher.cs



RemoveOut.cs



Remover.cs



Score.cs



Spawner.cs

Manager.cs : 블루투스 통신 제어

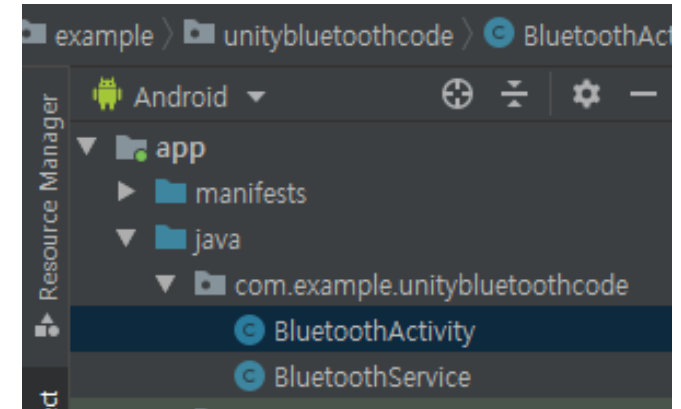
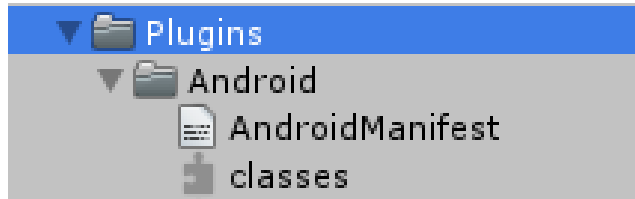
Pusher.cs : Pusher Object의 움직임

RemoveOut.cs : 배출구가 아닌 곳으로 떨어진 object 제거

Remover.cs : 배출구로 떨어진 Object 제거 및 스코어 카운트

Score.cs : 점수 변화 감지

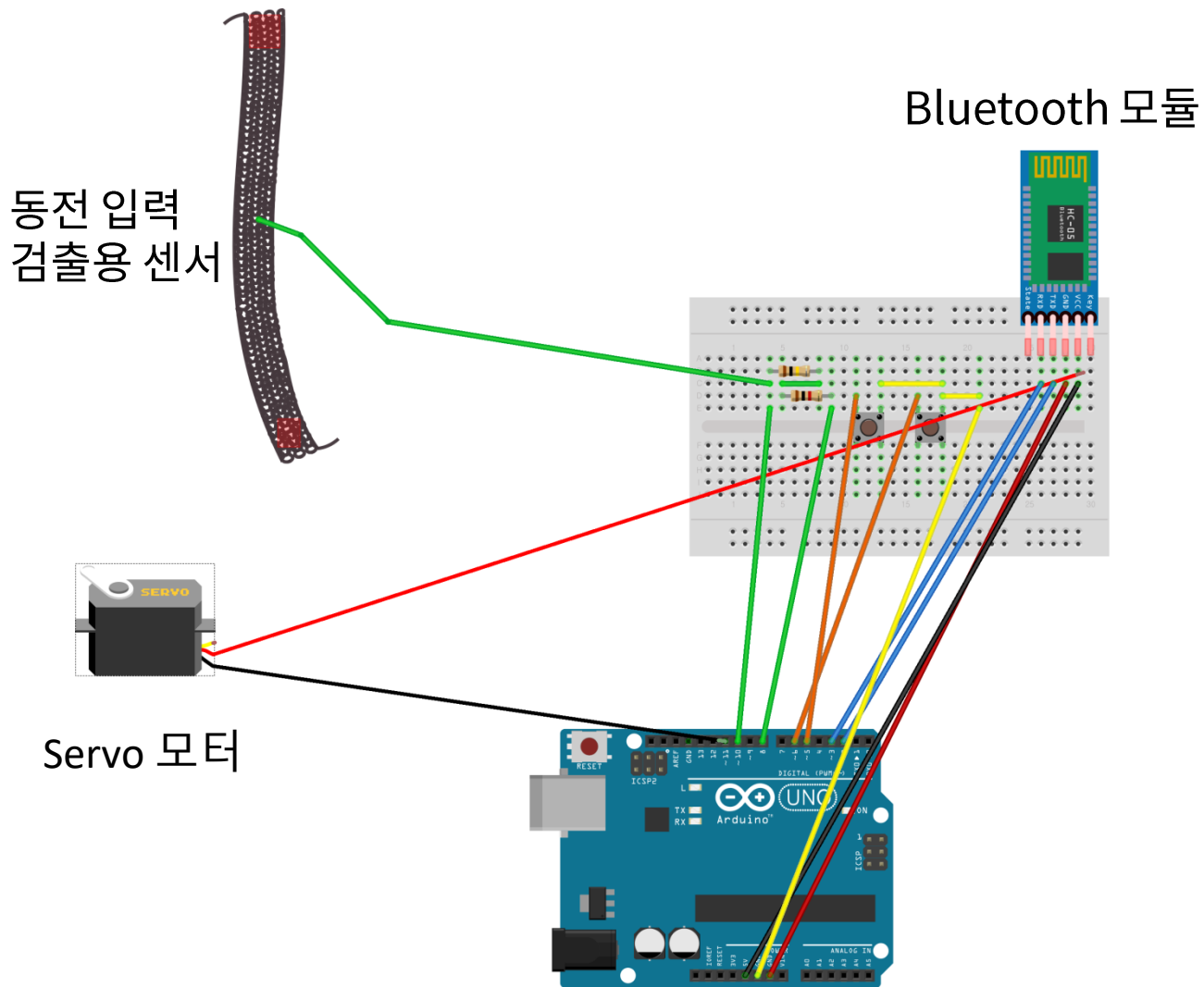
Spawner.cs : Coin Object Clone 생성



Classes.jar

```
<?xml version="1.0" encoding="UTF-8"?>
- <manifest android:installLocation="preferExternal" xmlns:tools="http://schemas.android.com/tools"
package="com.unity3d.player" xmlns:android="http://schemas.android.com/apk/res/android">
    <supports-screens android:anyDensity="true" android:xlargeScreens="true"
        android:largeScreens="true" android:normalScreens="true" android:smallScreens="true"/>
- <application android:label="@string/app_name" android:icon="@mipmap/app_icon"
android:theme="@style/UnityThemeSelector">
    - <activity android:label="@string/app_name"
        android:name="com.unity3d.player.UnityPlayerActivity">
        - <intent-filter>
            <action android:name="android.intent.action.MAIN"/>
            <action android:name="android.bluetooth.device.action.ACL_DISCONNECTED"/>
            <category android:name="android.intent.category.LAUNCHER"/>
        </intent-filter>
        <meta-data android:name="unityplayer.UnityActivity" android:value="true"/>
    </activity>
</application>
<uses-permission android:name="android.permission.BLUETOOTH"/>
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN"/>
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
<uses-feature android:name="android.hardware.bluetooth_le" android:required="true"/>
</manifest>
```

AndroidManifest.xml



fritzing

Arduino Board 회로도



```
sketch_oct28a

#include <CapacitiveSensor.h> //정전식 센서 라이브러리 추가
#include <Servo.h> //servo 라이브러리 추가
#include <SoftwareSerial.h>

CapacitiveSensor cs_8_10=CapacitiveSensor(8,10);
SoftwareSerial hc06(2,3); //Tx, Rx
Servo servo;

const unsigned int POTENTIAL=A0;

int sensor_value=0;

int btnPin=5;
int btnPin2=6;
int ledPin=13;
int ledPin2=8;
int buttonstate=0;
char data;

int servomove() {
    servo.write(100);
    delay(200);
    servo.write(0);
    delay(400);
    return 1;
}

void setup() {
    Serial.begin(9600);
    hc06.begin(9600);
    servo.attach(11); //servo 핀 설정
    Serial.println("Enter AT commands:");

    cs_8_10.set_CS_Autocal_Millis(0xFFFFFFFF);

    pinMode(ledPin, OUTPUT);
    pinMode(ledPin2, OUTPUT);
    pinMode(btnPin, INPUT);
    pinMode(btnPin2, INPUT);
}
```

```
void loop() {
    sensor_value=analogRead(POTENTIAL);

    long total1 = cs_8_10.capacitiveSensor(30);
    if(total1>80){
        hc06.println("input Money");
        Serial.println("input Money");
        delay(100);
    }
    if(hc06.available()){
        //Serial.write(hc06.read()); //블루투스 내용을 시리얼에 쓰기
        char value = hc06.read(); //- '0';
        //Serial.write(value);
        if(value==1){
            Serial.println("msg = "+value);
            digitalWrite(ledPin2,LOW);
        }
        else if (value==10){
            Serial.println("서보모터 동작");
            while(1){
                if(servomove()==1){
                    break;
                }
            }

            digitalWrite(ledPin2,HIGH);
        }else
        {
            Serial.println(value);
        }
    }

    if(Serial.available()){
        hc06.write(Serial.read()); //시리얼 모니터 내용을 블루투스에 쓰기
        Serial.println(Serial.read());
    }

    //delay(500);
}
```

```
data=Serial.read();
if (data=='A') {
    digitalWrite(ledPin2,HIGH);
    delay(100);
    digitalWrite(ledPin2,LOW);
    delay(100);
}

buttonstate=digitalRead(btnPin);

if(digitalRead(btnPin)==LOW) {
    //Serial.print(2);
    char value1 = '1';
    //hc06.write(value1);
    hc06.println("on");
    Serial.write(Serial.read());
    //hc06.write(Serial.read());
    //hc06.println(Serial.read());
    Serial.flush();

    //임시 테스트
    while(1){
        Serial.write("test구간 진입");
        if(servomove()==1){

            break;
        }
    }

    //digitalWrite(ledPin2,HIGH);
    //digitalWrite(ledPin,HIGH);
    //delay(100);
}

if (digitalRead(btnPin2)==LOW) {
    //Serial.print(3);
    //Serial.write(2);
    Serial.flush();
    //hc06.write("3");
    hc06.println("off");
}
```

Arduino code



CHPATER 05

최종 결과물 시연 영상





CHPATER 05

최종 결과물 시연 영상

최종 결과물 시연 영상



Coin Dozer MR

감사합니다

레스토랑 예약 프로그램

