

캡스톤 디자인 | 최종결과 보고서

프로젝트 제목(국문): 블록코딩 및 드론을 활용한 자기주도학습 서비스

프로젝트 제목(영문): Autodidacticism Service using Block Coding and Drone

프로젝트 팀(원):	학번: 20191743	이름: 이민주
프로젝트 팀(원):	학번: 20191775	이름: 백아름
프로젝트 팀(원):	학번: 20191777	이름: 송민지

1. 중간보고서의 검토결과 심사위원의 '수정 및 개선 의견'과 그러한 검토의견을 반영하여 개선한 부분을 명시하시오.

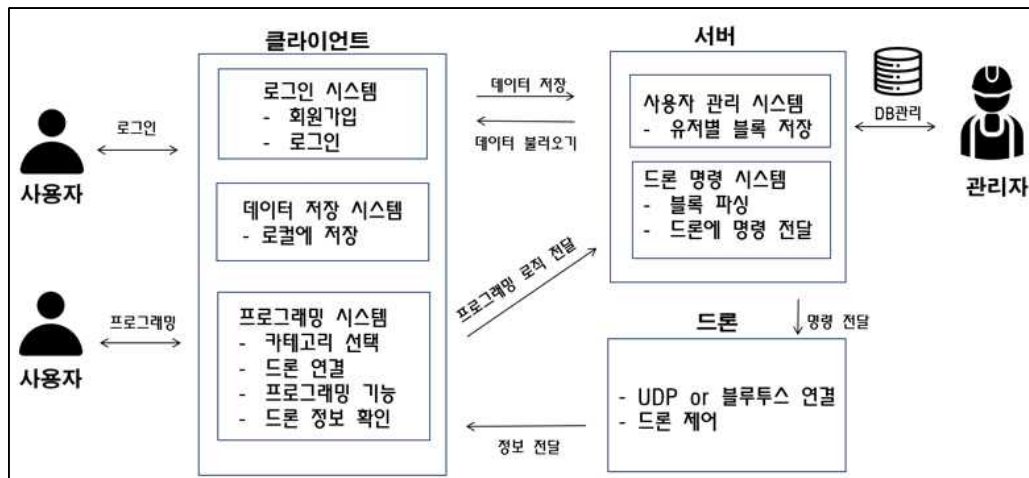
(1) 주제와 목표가 모호하다는 의견

'블록 코딩 및 드론을 활용한 프로그래밍 교육용 프레임워크 개발'이라는 주제가 모호하다는 의견에 '블록 코딩과 드론을 활용한 자기주도 학습 서비스'로 주제를 구체화하였다. 또한 그에 맞게 목표를 '학습'에 초점을 두어 재설정하였다. 수정 후 목표는 학습을 위한 '동기유발, 학습 콘텐츠 제공, 자기주도적 학습을 위한 평가시스템 제공'으로 설정하였다.

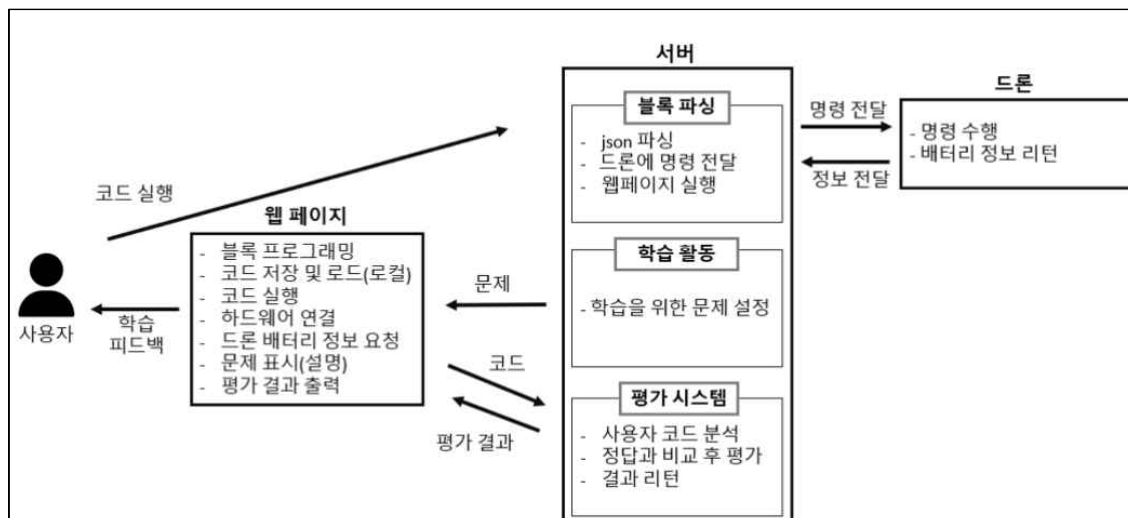
(2) 시스템 구성도의 정확성이 떨어진다는 의견

중간보고서의 시스템 구성도를 기준으로 클라이언트 쪽에 있는 내용을 서버 쪽으로 놓고 서버와 드론이 통신하는 구성으로 수정하는 것이 좋을 것 같다는 의견을 반영하여 수정하였다.

아래는 수정 전 시스템구성도이다.



아래는 수정 후 시스템 구성도이다.



중간보고서 피드백 이후, 불필요한 기능들은 삭제하였다.

2. 기능, 성능 및 품질 요구사항을 충족하기 위해 본 개발 프로젝트에서 적용한 주요 알고리즘, 설계방법 등을 기술하시오.

1) 주요 알고리즘

학습자가 웹 페이지에서 블록코딩을 한 후 Run 버튼을 누르면 웹 페이지의 작업 공간에 있는 코드가 Json 형식으로 서버에 전달된다. 전달받은 Json 형식은 아래 그림과 같다. 서버 측에서는 전달받은 Json에서 type, fields 등과 같은 필요한 정보를 파싱해서 list에 저장하고 저장된 명령을 tello drone에게 전달한다.

```
"blocks": {
  "languageVersion": 0,
  "blocks": [
    {
      "type": "takeoff",
      "id": "vhaV;U.j?+hBnETQr_)",
      "x": 48,
      "y": 25,
      "next": {
        "block": {
          "type": "flight_move",
          "id": "navf{1aPutz-@Eg/Taup",
          "fields": {
            "direction": "forward",
            "distance": 20
          },
          "next": {
            "block": {
              "type": "down",
              "id": "BzK1!zJT}#HXX;2y}ARf",
              "fields": {
                "NAME": 20
              }
            }
          }
        }
      }
    }
  ]
}
```

```
def json_blocks_parse(data):
    jsonArray = data.get("block")
    name = jsonArray.get("type")
    if jsonArray.get("type") == "controls_repeat_ext":
        json_blocks_for(jsonArray)
        command = ''
    elif jsonArray.get("type") == "flight_move":
        command = list_save(list(jsonArray.get("fields").values()))
        print(command)
        tello.send_command(command)
    elif jsonArray.get("type") == "wait":
        command = ''
        val = list_save(list(jsonArray.get("fields").values()))
        wait(val)
    else:
        if jsonArray.get("fields") is not None:
            val = list_save(list(jsonArray.get("fields").values()))
            command = name + ' ' + val
        else:
            command = name

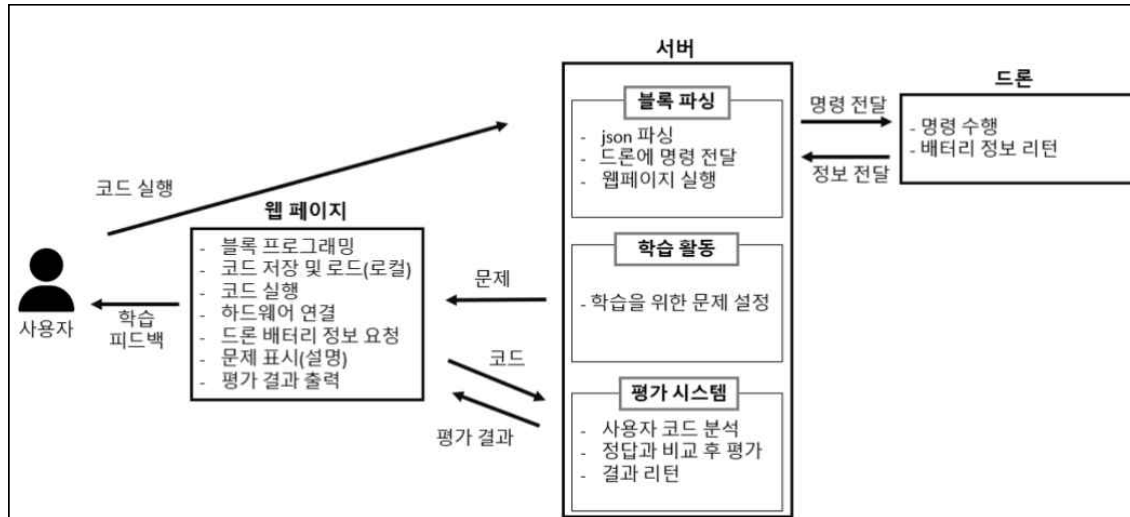
print(command)
if command != '':
    tello.send_command(command)
```

2) 설계방법

블록 프로그래밍을 위한 작업 공간과 툴 박스를 구현한 로컬 웹 페이지에서는 먼저 드론과 연결을 하고 연결이 잘 되었는지 확인이 가능하도록 설계했다. 드론과 정상적으로 연결된 상태에서 사용자가 블록 프로그래밍을 하고 Run 버튼을 누르면 작업 공간에 있는 블록 데이터를 Json 형태로 서버에 전송하게 된다. 이후 Json 데이터를 파싱해서 드론에 명령을 전달해 드론 제어가 가능하도록 설계했다. 웹 페이지에 있는 show battery 버튼을 누르면 드론으로부터 배터리 정보를 받아서 남은 배터리의 용량을 확인할 수 있게 하였다. 또 사용자의 블록 코드의 재사용성을 높이기 위해 블록 코드 데이터를 key 값을 포함해서 로컬에 저장하고 불러올 수도 있다. 블록 코딩 후 Run 버튼을 누를 때 블록 코드를 로컬에 자동으로 저장하는 기능도 추가해서 설계했다.

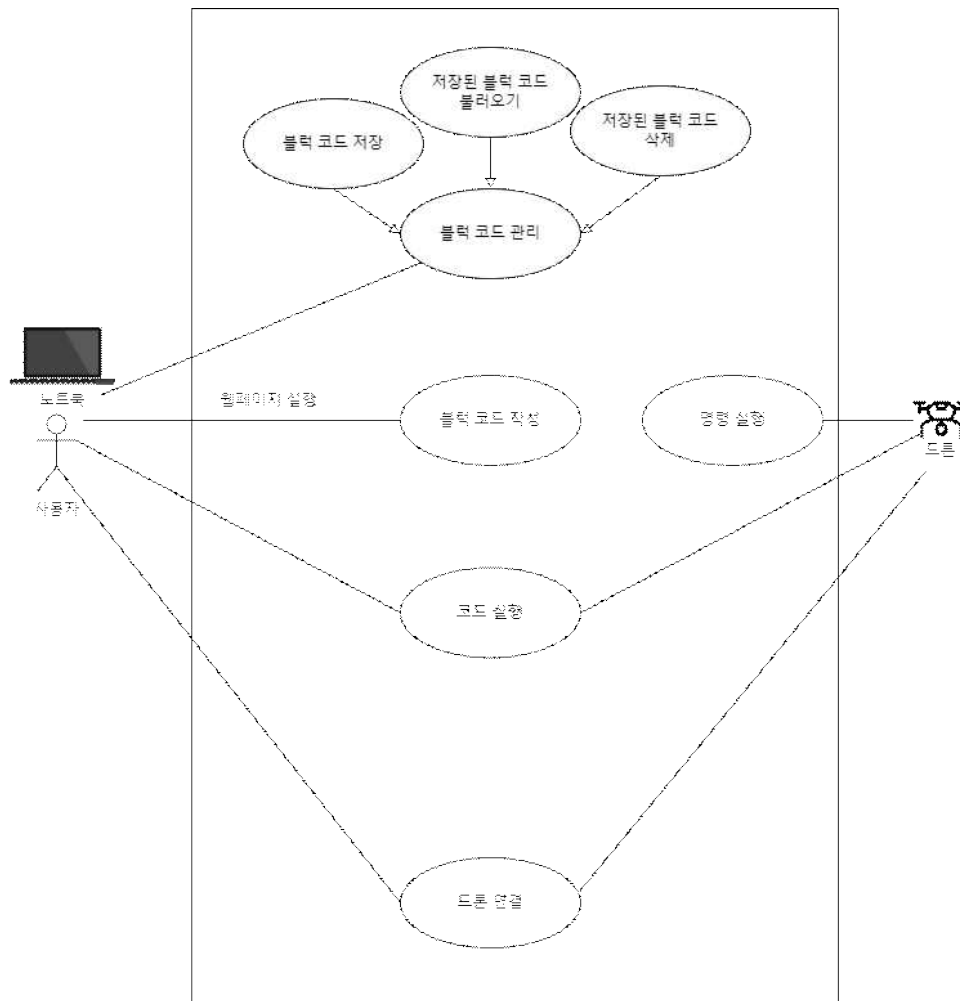
3. 요구사항 정의서에 명시된 기능 및 품질 요구사항에 대하여 최종 완료된 결과를 기술하시오.

(1) 전체 시스템 구성도



(2) 현재 구현 범위에 대한 유스케이스

- 위 시스템 구성도에서 최종적으로 웹 페이지는 문제 표시 및 평가 결과 출력을 제외한 부분, 서버는 블록 파싱 부분까지 구현하였다. 아래 그림은 이에 대한 유스케이스이다.



4. 구현하지 못한 기능 요구사항이 있다면 그 이유와 해결방안을 기술하시오,

최초 요구사항	구현 여부(미구현, 수정, 삭제 등)	이유(일정부족, 프로젝트 관리미비, 팀원변동, 기술적 문제 등)
태스크 수행기능	삭제	프로젝트 주제 변경으로 요구사항 삭제
사용자 로그인 기능	삭제	프로젝트 주제 변경으로 요구사항 삭제
로그 작성 기능	미구현	프로젝트 관리미비

5. 요구사항을 충족시키지 못한 성능, 품질 요구사항이 있다면 그 이유와 해결방안을 기술하시오.

분류(성능, 속도 등) 및 최초 요구사항	충족 여부(현재 측정결과 제시)	이유(일정부족, 프로젝트 관리미비, 팀원변동, 기술적 문제 등)
동시 접속자 수	삭제	불필요한 기능으로 판단하여 삭제
느린 작업에 대한 사전 경고	미구현	일정부족으로 인한 미구현
도움말 제공	미구현	일정부족으로 인한 미구현
인증	삭제	프로젝트 주제 수정으로 요구사항 삭제

6. 최종 완성된 프로젝트 결과물(소프트웨어, 하드웨어 등)을 설치하여 사용하기 위한 사용자 매뉴얼을 작성하시오.

(1) 소스코드 실행

- 1) Github 홈페이지 <https://github.com/matchaing/Blockly>에서 소스코드를 폴더로 다운로드
- 2) VisualStudio Code IDE를 실행시킨 후 Blockly폴더 내에서 app.py을 실행
- 3) 코드 실행 후 콘솔 창에 뜨는 주소(<http://127.0.0.1:5000/>)에 접속하여 웹 페이지 띄우기

(2) 하드웨어 연결

- 1) 텔로 드론 배터리 삽입 부분에 적힌 와이파이 이름 확인
- 2) 배터리 넣은 후 몸체 옆 부분 전원 버튼 누르기
- 3) 드론 앞 부분에 있는 LED 색상이 노란색으로 일정하게 깜빡거리면 기기(노트북)와 와이파이 연결
- 4) 와이파이 연결 확인 후 웹 페이지에 있는 Connect 버튼 클릭
- 5) 연결되었다는 알림창이 뜨면 연결 완료
- 6) 웹 페이지에서 프로그래밍 후 Run 버튼을 누르면 실행 가능
- 7) 실행 하면 코드에 맞게 드론이 동작함

(3) 그 외 기능

- 1) Run 버튼 : 코드를 실행하기 위한 버튼.
- 2) Connect 버튼 : 드론과 연결하기 위한 버튼.
- 연결이 되지 않을 경우 연결되었다는 알림창이 뜨지 않는다.
- 3) Save 버튼 : 현재 작업공간에 있는 블록을 저장하는 버튼. key 값을 입력하여 저장한다.
- 4) Load 버튼 : 저장했던 블록을 불러오는 버튼. key 값을 입력하여 불러온다.

- load하게 되면 기존에 작업공간에 있는 코드가 사라지기 때문에 주의가 필요하다.
- 5) Remove 버튼 : key 값을 입력하면 해당 코드를 삭제한다.
- 6) Clear 버튼 : 저장되어있는 모든 코드를 삭제한다.
- 7) Show Battery 버튼 : 연결된 드론의 현재 남은 배터리 용량을 퍼센트로 보여준다.

7. 캡스톤디자인 결과의 활용방안

2018년부터 초·중·고등학교의 소프트웨어 교육이 강화되고 정규 교육 과정으로 편성되면서 프로그래밍 교육에 대한 중요성이 대두되고 있다. 그러나 소프트웨어 교육 기기는 비교적 고가라는 단점이 있어 지도자 입장에서는 다양한 수업을 시도하기가 어렵다. 또 단순 컴퓨터 프로그래밍만 하기에는 학생들의 흥미를 유발하는 데 한계가 있다는 단점도 존재한다. 이러한 문제점을 해결하고자 드론을 결합시킨 프로그래밍 서비스를 개발하였다.

본 서비스에서 사용하는 하드웨어는 비교적 저렴한 가격대인 텔로 드론을 선택하여 기존에 소프트웨어 교육 기기가 고가라는 단점을 보완하였다. 또한 학습자는 프로그래밍 결과를 드론을 통해 즉시 시각적으로 확인할 수 있어 학습을 위한 동기 유발이 가능하다. 학습자는 학습 콘텐츠 및 평가 시스템을 통해 자기 주도적 학습이 가능하다. 프로그래밍 문제를 해결하는 과정에 있어서 블록 코딩 단독의 학습 보다는 블록 코딩에 드론을 혼합한 방식이 학습 효과가 더 클 것으로 기대한다. 궁극적으로, 소프트웨어 교육의 중요성이 대두된 현 시점에서 본 프로젝트의 서비스가 소프트웨어 교육의 다양성에 이바지 할 수 있는 것을 기대효과로 본다.