**목차**

1. **서론**
   1. 배경
   2. 목적
2. **본론**
   1. 배경 지식
   2. 프로그램 구성 – **간략한 프로그램 소개**
   3. 세부 컴포넌트 기능 – **코드 & GUI 소개**
   4. 프로그램 사용 방법
   5. 프로그램 검증
   6. 결과 및 분석
3. **결론**
   1. 결론
   2. 활용 방안 및 기대 효과
   3. 향후 과제
4. **기타**
   1. 조원 기여도
   2. 수업 관련 느낀 점 및 건의사항
   3. 참고 자료
5. **서론**

1.1 배경

포켓몬 GO는 게임상의 가상 지도를 기반으로 실시간 데이터를 이용하여 자유롭게 플레이 할 수 있는 스마트폰 이용자를 위한 게임이다. 이 게임은 ‘niantic’과 ‘nintendo’의 합작으로 만들어진 게임으로 기본의 포켓몬스터 게임의 형식과는 다른 방식으로 개발되었다. 이는 플레이어가 캐릭터를 선택적으로 움직이는 것이 아닌, 플레이어가 mobile device에서 수신하는 GPS상의 위치 데이터를 받아 게임상의 지도에 나타내고 이 지도상에 포켓몬, 특수 아이템 등이 나타나게 되어 사용자의 GPS 위치가 근처에 도달하였을 때 컨텐츠를 이용가능 하도록 한 게임이다. 또한 모든 이용자가 스마트폰만 있다면 무료로 이용할 수 있도록 하여 기존의 게임과는 다르게 접근성을 높였다.

포켓몬 go 은 실제 위치를 기반으로 한 증강현실 게임인데, 이 파급력은 세계인을 놀라게 할 정도로 크게 나타났다. 실제로 인기지역, 인구가 많은 지역을 기반으로 특수한 컨텐츠인 ‘포켓스탑’ 이 지도상에 표시가 되었는데 이는 게임 플레이를 위한 아이템을 주기도 하고, 근처에 포켓몬을 유인하기도 하는 컨텐츠였다. 이는 게임 내에서의 인기지역이 되는 것 뿐 아니라 실제 이용자들이 직접 찾아가야만 하는 게임상의 조건에 의해 실제로 인기를 끌게 되는 현상을 야기하기도 했다. 또한 포켓몬 플레이어들이 스마트폰 화면만을 주시하고 움직이면서 생겨난 여러 해프닝들이 크게 이슈화 되기도 했다.

위와 같은 이유로 포켓몬 go 는 인기 컨텐츠 포켓몬스터를 이용하여 새로운 방식의 게임을 구현함 으로써 큰 인기와 많은 이슈를 불러오는 결과를 초래 했다. 하지만 GPS 기반 게임의 고전적인 문제인 게임내 컨텐츠 부족을 해결하지 못하면서 또 다시 대중의 관심을 점차 잃게 되었고, 지금은 실제 이용자가 많지 않은 실정이다.

1.2 목적

포켓몬 GO게임은 현재 이용자가 거의 없지만, ‘포켓몬’이라는 컨텐츠는 전 세계적으로 연령에 상관없이 사랑받고 있으며 사람들에게 많은 향수를 불어 일으킨다. 이에 인기를 끌었던 ’포켓몬 GO’의 흥미 요소 중 하나인 다양한 진화가 가능하다는 점은 남겨두고, 유저들의 희망사항을 추가해 컨텐츠가 부족한 ‘포켓몬 GO’의 문제점을 해소하기 위해 ‘BE 포켓몬’을 개발하려 한다. 게임의 주체가 트레이너였던 기존 방식에서 ‘BE 포켓몬’은 게임의 주체를 포켓몬 캐릭터 자체로 설정하여 직접 플레이 하는 것이 게임의 방식이다. 이로 인해 새로운 방식의 또다른 하나의 컨텐츠를 즐길 수 있을 것이며, 포켓몬을 무자비하게 잡기만 하는 방식과는 달리 스스로 선택하여 성장할 수 있는 게임을 만들도록 한다. 또한 User Interface의 목적에 맞도록 사용자들이 쉽게 이해하고 실행 할 수 있도록 제작한다.

1. **본론**

2.1 배경지식

포켓몬스터는 포켓볼(poket ball) 이라는 가상용 휴대용 캡슐로 몬스터를 포획한 다음에 간편하게 데리고 다니며 키우는 게임의 내용을 표현한 제목이다. 원작의 1차적인 목표는 각 지역에 있는 8개 체육관의 관장을 이겨 배지를 모은 후 최종적으로 챔피언이 되는 것이다. 이러한 진행 과정에서 스토리를 깨는 RPG적인 요소와 귀여운 캐릭터의 다양성 또한 게임의 흥행에 크게 작용했다.

이번 프로젝트의 기본 조건으로는 스마트폰을 이용한 NMEA logging이 가능한 앱을 이용하여 1시간 이상의NMEA data를 logging 한다 위의 데이터를 기반으로 GUI를 구현하는데, 어떠한 방식으로의 제한은 없다.

NMEA logging data에서 알 수 있는 여러가지 정보로는 먼저 위성의 ID로 $GPGGA 와 같은 문자로 시작을 하게 되는데 이는 앞의 2개의 문자는 장치의 ID를 나타내고 뒤의 3개의 문자는 문장을 구분하는 용도로 사용된다. 이에 아래의 데이터만을 받아서 프로그램 구성에 이용했다.

$GPGGA 의 경우 각 섹터별로 UTC Time, 위도, 북위/남위, 경도, 동경/서경/ position fix, 평균해수면 기준 고도, 고도 단위를 나타낸다.

-position fix 의 값이 0이 나오면 신뢰 할 수 없는 데이터라는 의미로 볼 수 있다.

$GPRMC 의 경우 각 섹터별로 UTC Time, 상태, 위도, 북위/남위, 경도, 동경/서경, 속력 ( 단위 : knots ), 진행방향, UTC Date 를 나타낸다.

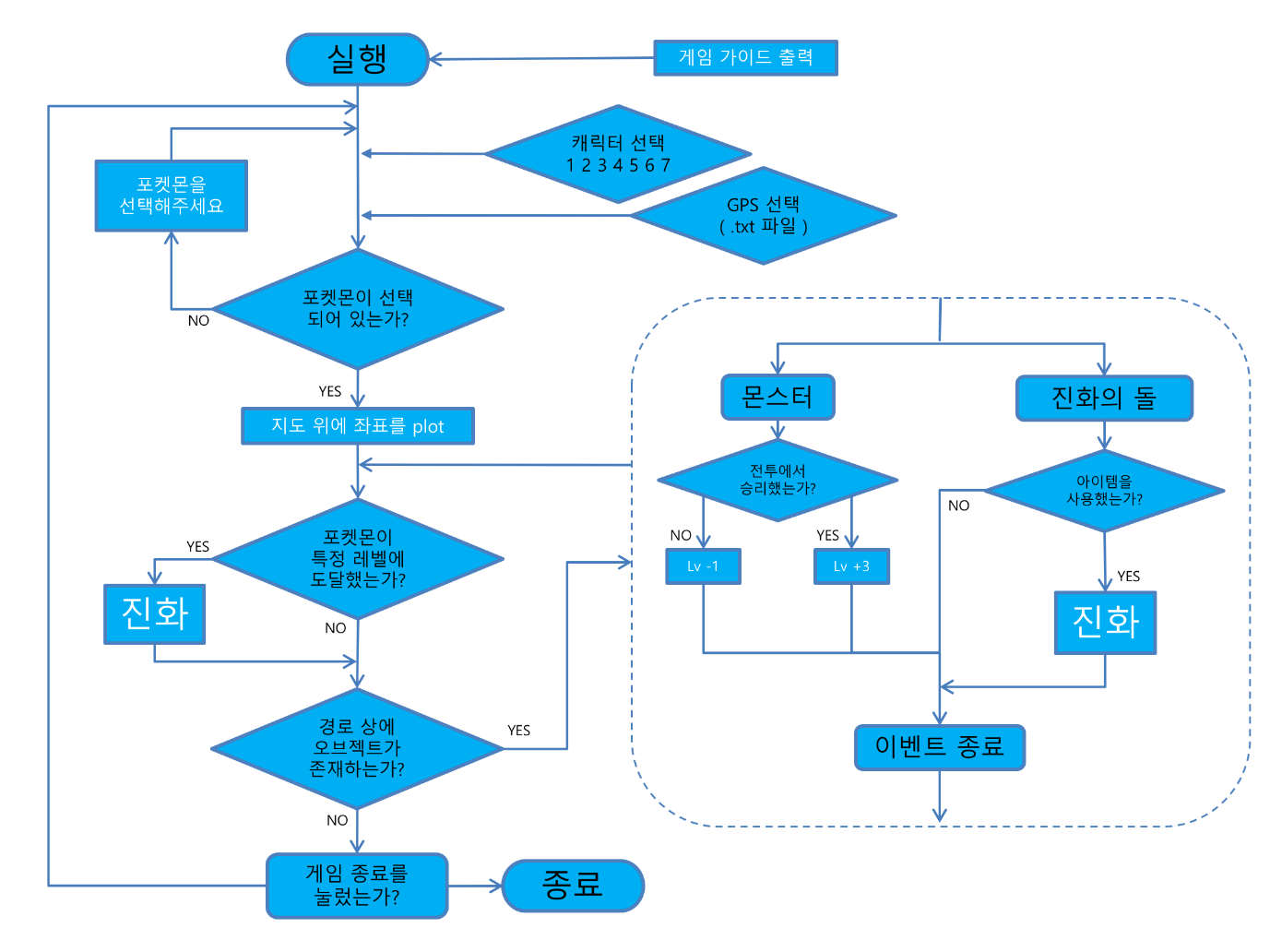
-UTC Time은 국제표준시로 hhmmss.sss의 형태로 표현된다.

-위도는 ddmm.mmmm의 형태로 표현된다

-경도는 ddd.mm.mmmm의 형태로 표현된다

2.2 프로그램 구성

‘포켓몬스터’의 spinoff 를 지향하여 포켓몬을 잡고 키우는 방식이 아닌, 내가 포켓몬이 되어 여행하고 자라는 형식의 새로운 게임방식으로 진행되며, 이러한 구성을 ‘matlab’의 GUI를 통해 제작 구성한다. 이 때 포켓몬스터의 흥행요소인 스토리와 RPG적인 요소와 귀여운 캐릭터의 다양성을 살리고 자유도를 주어 사람들의 흥미를 유도한다.



게임 이용시의 간단한 순서도 이다. 게임 실행 시 게임 이용방법을 보여주고 실행이 되며, 먼저 포켓몬을 선택하여 주체를 화면상에 띄워준다. 포켓몬이 선택되어 있을 경우, GPS 데이터를 선택하여 순차적으로 읽어가며 지도상에 표현된다. 이 때 이동거리에 비례하려 경험치를 얻게 되고 레벨업이 가능하다. 또한 특수 이벤트가 존재하는 지역의 근처에 도달 했을 경우 몬스터의 출현 또는 아이템을 획득하게 되고 배틀 모드 또는 진화가 진행이 된다.

GUI 의 기본 목적에 맞도록 누구나 쉽게 이용이 가능하도록 하기위한 조건을 충족하기 위해 자체적으로 제시한 조건은 다음과 같다.

- 게임 이용방법 : 게임 실행이 되면 간단한 이용 방법을 보여준다.

- 나의 캐릭터 선택 : 원하는 캐릭터를 먼저 선택하여 프로그램 화면에 나타낸다.

- 이동했던 GPS 데이터 선택 : 이동했던 GPS데이터 정보를 프로그램에 연동시켜 게임을 진행한다

- 게임내 컨텐츠 이벤트 발생시 선택 : 게임을 진행하는 도중에 특수한 이벤트가 발생했을 경우 그 이벤트를 선택적으로 진행하도록 한다.

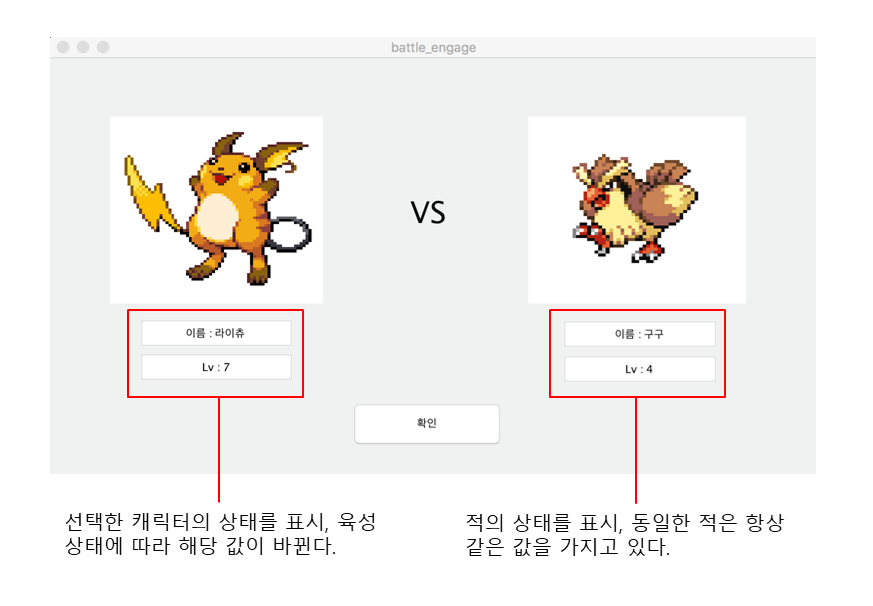
- 캐릭터 배틀 및 진화 : 캐릭터가 특정조건을 만족하면 새로운 이벤트를 진행하게 된다.



GUI에서 가장 중요한 요소는 누구나 쉽게 이해하고 사용 할 수 있게 하는 간결성이다. 따라서 하나의 GUI에 최대한 많은 정보가 나타나도록 하였다. 위의 사진은 ‘BE 포켓몬’의 주된 화면이 되는 메인 화면이다. 지도화면을 나타내고 위에 위치 정보를 plot을 이용하여 지도위에 표시했다.

아이템은 전 좌표를 지정하여 항상 같은 위치에 나타나도록 하였으며, 몬스터는 plot전 rand를 이용하여 항상 다른 위치에 무작위로 나타나도록 설계하였다. 이후 해당이벤트가 진행되고 난 후 좌표에 검은색의 Marker를 한번 더 plot하여 해당부분에서의 이벤트를 끝냈음을 알 수 있도록 하였다. 이동경로 부분에서는 for문을 이용해 연속적인 plot을 하여 실시간으로 움직이며 작동될 당시의 화면이 되도록 했다.

지도를 plot한 Axis와는 별개로, 캐릭터를 표시해주는 Axis가 하나 더 존재한다. 이는 처음 캐릭터를 선택할 때와 캐릭터가 진화조건을 충족하여 진화했을 경우 해당 캐릭터에 맞는 그림으로 새로 plot하게 된다. 이 두개의 Axis가 하나의 main GUI 화면에 있어서 프로그램을 실행하는 데 간결하게 구성했다.

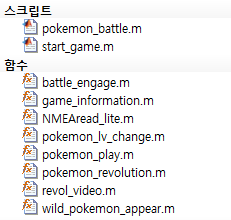


ex. 포켓몬 배틀 GUI

게임을 진행하다 보면 몬스터와 마주하게 되는데 , 이 경우 몬스터와의 조우를 확인시켜주는 새로운 화면이 뜬다. 해당 GUI는 내 캐릭터와 몬스터의 상태를 알려줄 Axis와 텍스트박스로 이루어져 있다. 이 때 앞의 데이터를 그대로 유지하여 이용한후 다시 데이터가 돌아 갈 수 있도록 하기 위해 global 함수를 이용하여 주요한 캐릭터, 레벨, 경험치, 진화, 배틀 승패를 지정하여 지속적으로 해당 변수를 로드 할 수 있도록 했다.

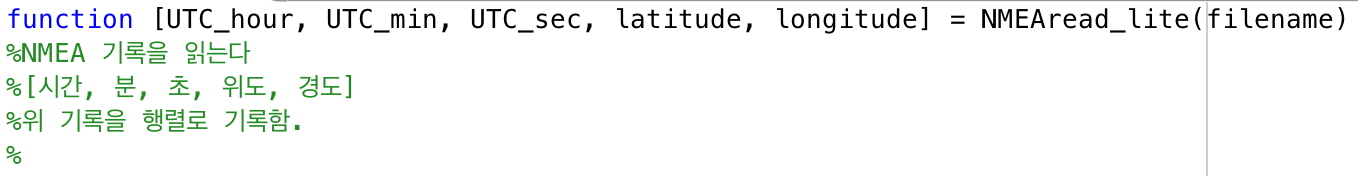
2.3 세부 컴포넌트 기능

(1) 파일 구성

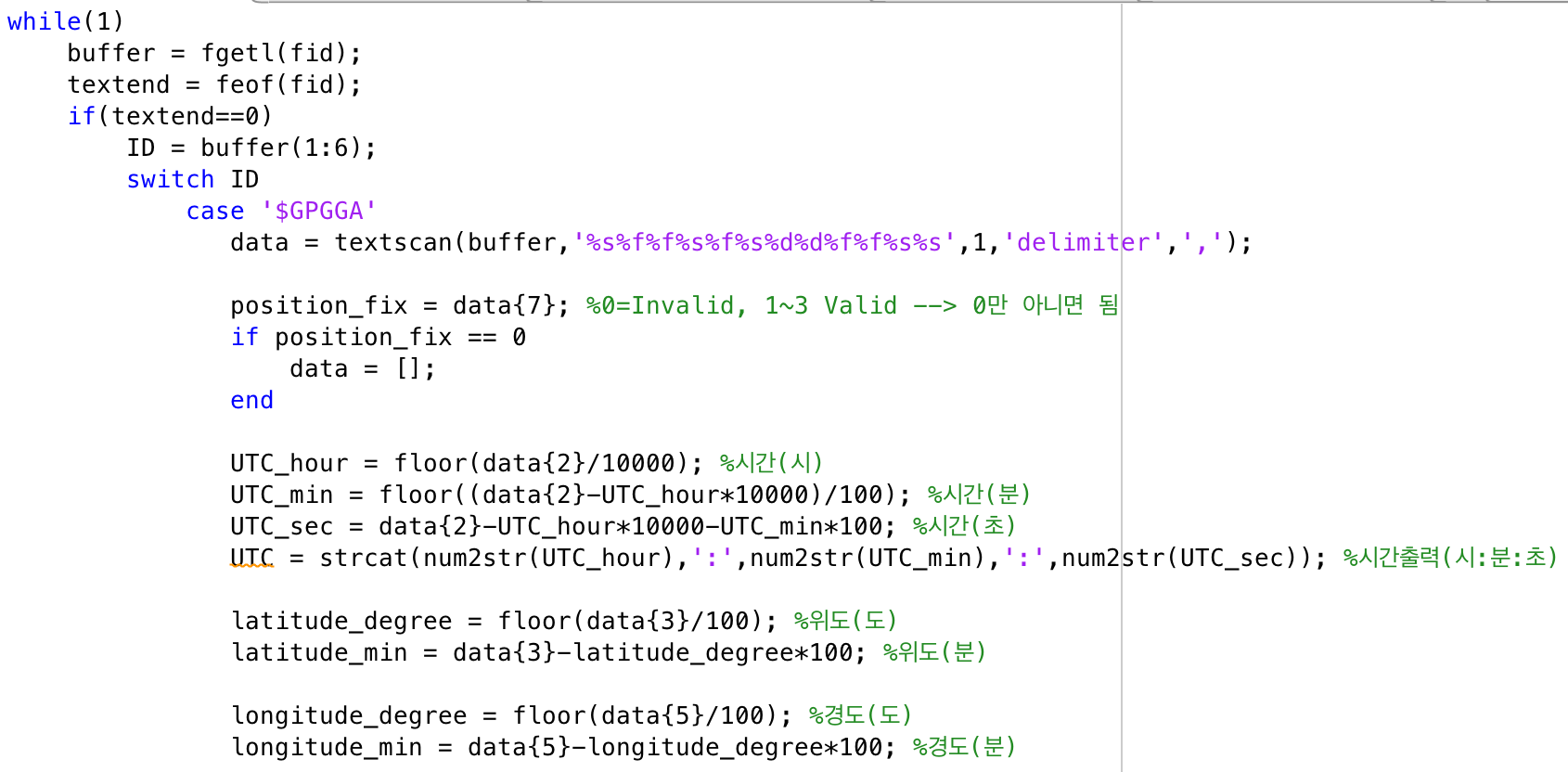


파일은 이렇게 스크립트 2개, 함수 8개로 이루어져 있으며, 함수 8개 중 6개는 GUI 파일이다.

(2) NMEAread\_lite.m



먼저 ‘NMEAread\_lite.m’ 파일은, 안드로이드 기반 스마트폰에서 어플을 이용해 받은 NMEA 텍스트 파일을 읽어 들이기 위한 함수이며, 게임에서 필요한 UTC 시간, 위도, 경도를 읽어올 수 있다.

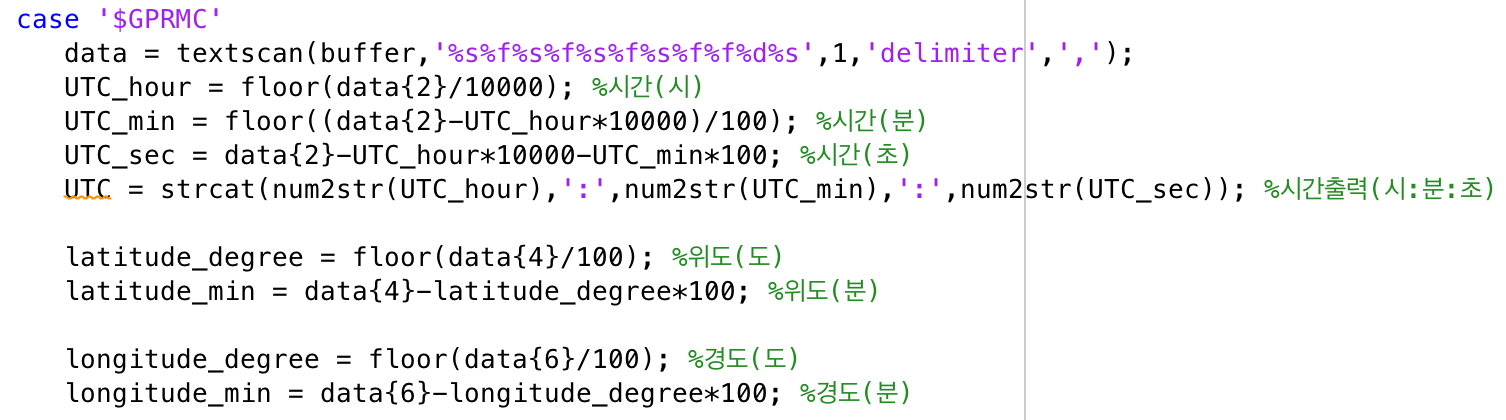


코드를 살펴보면, 먼저 while 반복문 안에서 fgetl(fid)로 txt 파일의 내용을 한 줄씩 읽어서 필요한 정보를 읽을 수 있게 했다. 이 프로그램에서 사용할 기록은 ‘$GPGGA’, ‘$GPRMC’ 두 종류이므로, switch 함수를 이용해서 읽을 정보를 나누어주었다.

‘$GPGGA’에는 ‘position fix’라는 부분이 있는데, 이 부분이 0이어선 안된다. 따라서 position fix가 0일 경우 그 줄의 데이터를 없애주도록 했다.

그리고 UTC(국제표준시)는 ‘hhmmss.sss’ 형태로 표시되기 때문에 읽어오려면 약간의 계산이 필요했다. 먼저 이 수를 10000으로 나누어서 ‘hh.mmsssss’형태로 바꾼 뒤, floor 함수를 이용해서 소수점 아래 수를 내림해서 ‘hh’ 즉 시간을 얻었다. 같은 방식으로 ‘hhmm’을 얻어서, 앞에서 얻은 ‘hh’에 100을 곱해 빼고 ‘mm’ 즉 분을 얻었다. 같은 방식으로 ‘ss.sss’ 즉 초까지 얻을 수 있었다.

다음은 위도와 경도인데, 위도는 ‘ddmm.mmmm’, 경도는 ‘dddmm.mmmm’ 형태로 기록되어 있으므로, UTC를 구했던 방식과 똑같이 ‘dd’ 혹은 ‘ddd’, ‘mm.mmmm’을 나누어 구할 수 있었다.



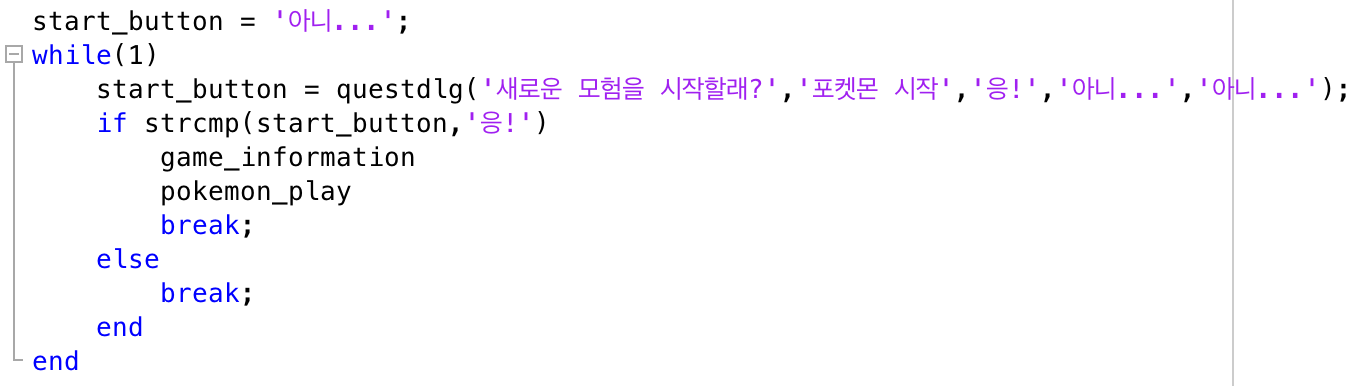
‘$GPRMC’의 경우도 ‘$GPGGA’와 완전히 같은 방식으로 시간, 위도, 경도를 뽑아낼 수 있도록 만들어주었다.

하지만 실제로 GPS 기록을 지도에 투영하려면 위도와 경도를 수정해줄 필요가 있었다. NMEA에서 읽어온 위도와 경도는 모두 DMS 단위로 되어있기 때문에, Degree 단위로 변환해줘야 지도에 투영할 수 있기 때문이다.

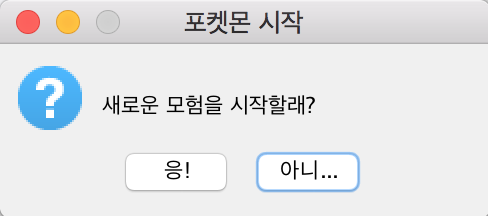
C:\Users\Administrator\Desktop\코딩이미지\5_DMS 환산.png

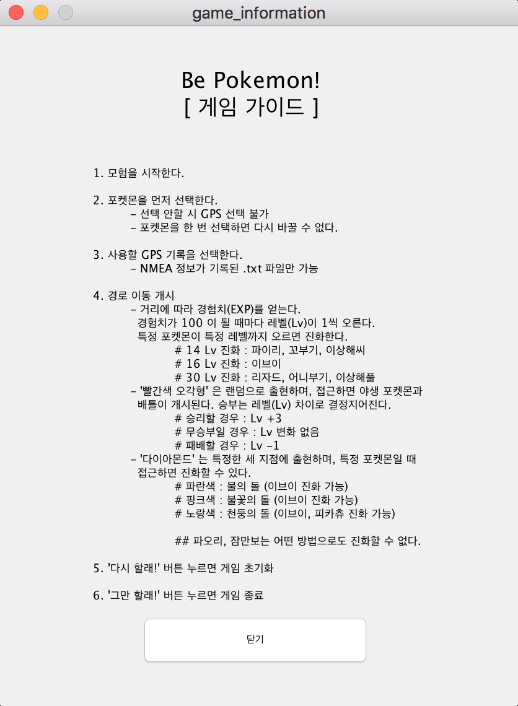
따라서 결과 data를 구성할 때 위도와 경도를 Degree 단위로 환산해주기 위한 식을 입력해주었다. 따라서 이 ‘NMEAread\_lite.m’ 함수를 이용하면 따로 계산 필요 없이 바로 Degree 단위의 위도와 경도 좌표를 얻을 수 있도록 했다.

(3) start\_game.m



이 파일은, matlab에서 게임을 실행할 때 정말 실행할지 말지를 물어보는 창을 띄우고, 게임을 실행시키기 위해 만든 함수이다. 반복문 안에 questdlg 함수를 이용해 질문을 넣어주었고, ‘응!’을 선택하면 게임 가이드를 띄우면서 게임 실행, ‘아니…’를 선택하면 break로 실행을 중지하도록 했다.

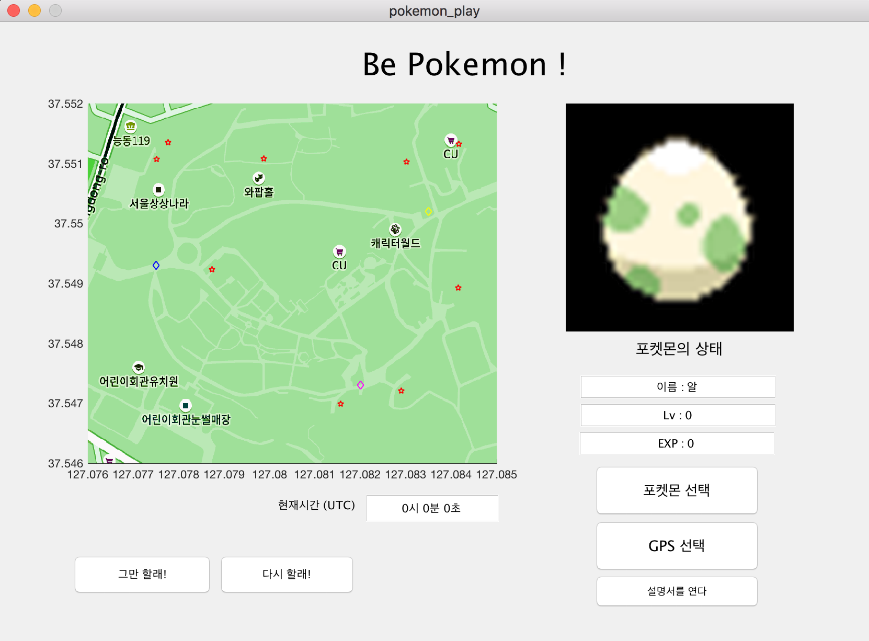


(4) game\_information.m

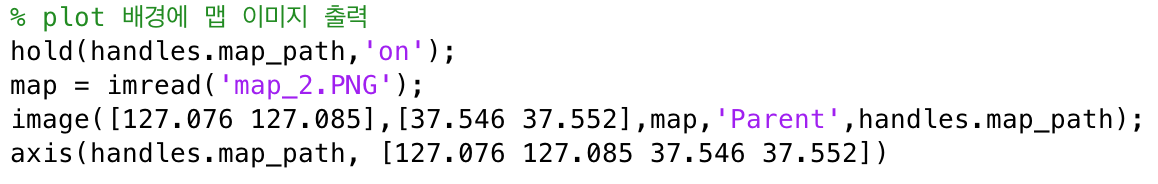
이 파일은 단순히 게임 가이드를 표시해주는 GUI이다.

‘닫기’를 누르면 창이 꺼지도록 만들었다.

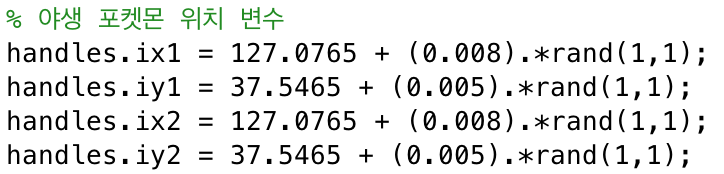
(5) pokemon\_play.m



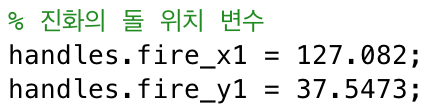
‘pokemon\_play.m’ 파일은, 이 게임의 본체라 할 수 있는 GUI 파일이다. 이 GUI는 어린이대공원을 표시해주는 지도, 현재 포켓몬 이미지, 포켓몬의 상태, 포켓몬과 GPS 파일 선택 버튼, GPS에 기록된 당시 시간 등 많은 정보들로 구성되어 있다.



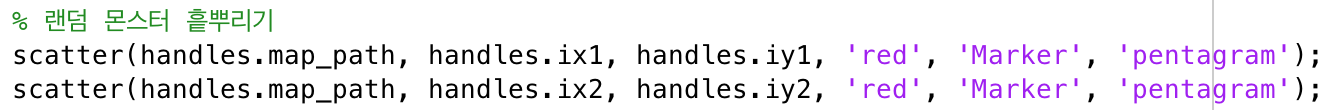
먼저 배경이 될 지도는 ‘구글맵’에서 우리가 이용할 좌표 범위 만큼을 잘라낸 이미지 파일(map\_2.png)에 딱 맞게 설정한 좌표축에 띄우도록 했다. 좌표축은 GPS 기록을 그대로 이용할 수 있는 경도, 위도 좌표축이다.



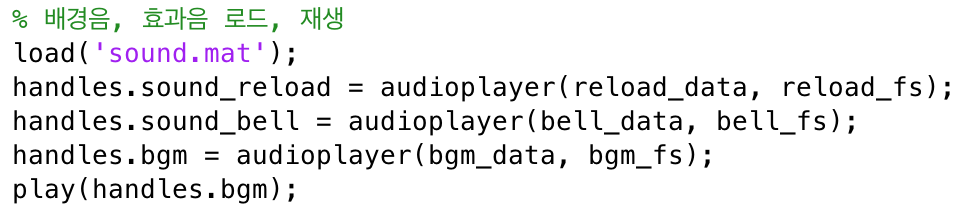
맵 중간중간에 보이는 빨간색 오각형 오브젝트는 모두 ‘랜덤 출몰하는 야생 포켓몬’을 표현해준다. 이들의 위치는 위 이미지처럼 rand 함수를 이용해서 지도 범위 내 아무 곳에서나 출몰하도록 만들었다.



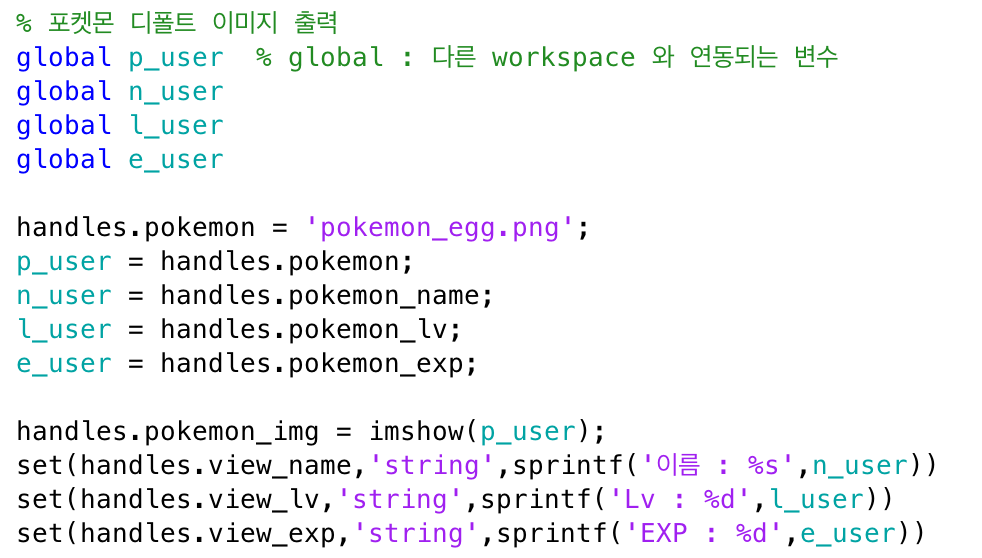
반면 다이아몬드 형태의 오브젝트는 ‘위치가 고정된 아이템’을 표현해주므로, 위치를 고정시켜주었다.



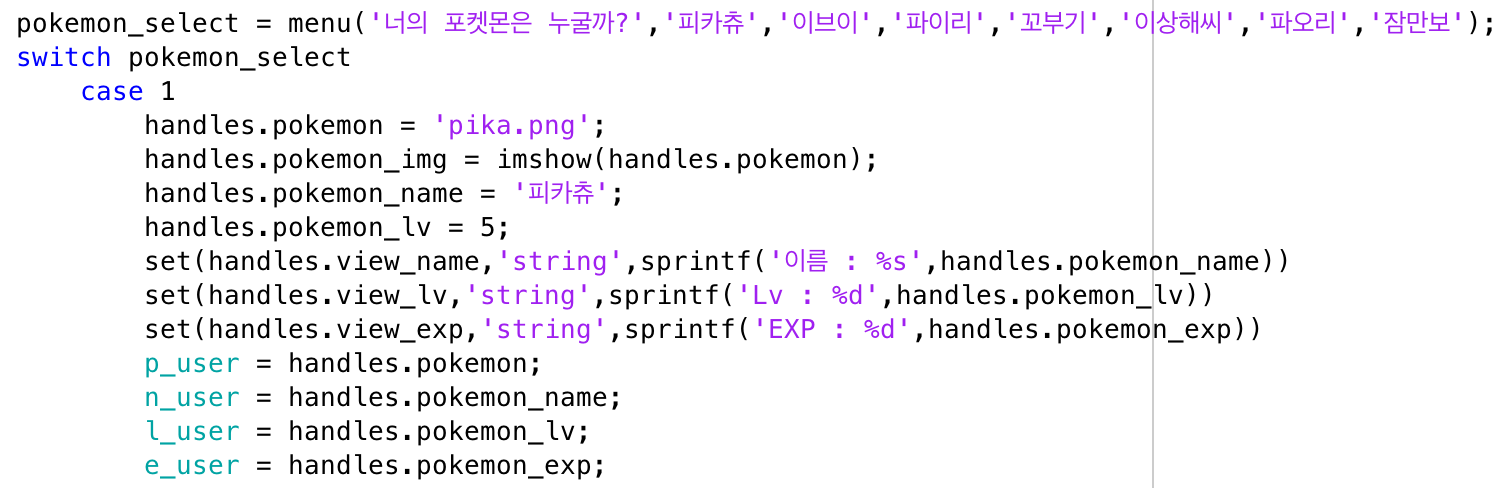
이렇게 값을 설정해준 오브젝트들은 scatter 함수를 이용해서 지도에 뿌려주었다.



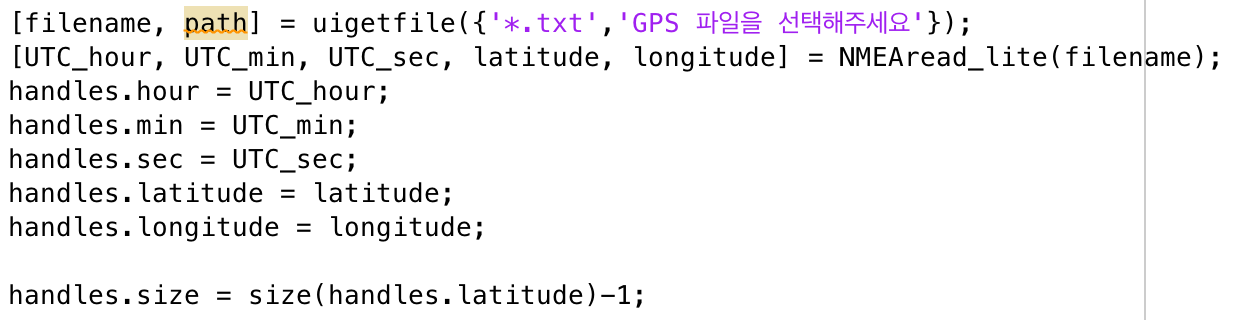
또한 ‘matlab’ 내부 툴박스를 이용해서 배경음악이 재생되도록 만들었다. 여기서 사용하는 음악 파일은 .mat 파일로, wav 파일을 행렬로 변환한 것을 사용한다.



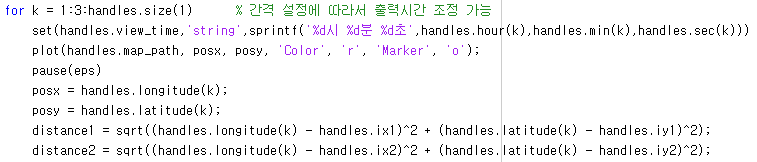
이 부분은 GUI 우측에 포켓몬 이미지, 포켓몬 상태를 출력해주는 부분이다. 여기에 global 함수가 보이는데, 이는 여러 개의 GUI에서 정보를 공유해야 하는 게임 특성 상 필요했던 함수이다. global 함수를 이용하면 여러 workspace에서 같은 변수를 활용할 수 있다. 이렇게 global 로 선언해준 변수들에 현재 포켓몬의 이미지, 이름, 레벨, 경험치 기록을 넣어주고, imshow로 이미지를, set으로 포켓몬의 상태 값들을 GUI에 출력해주도록 했다.



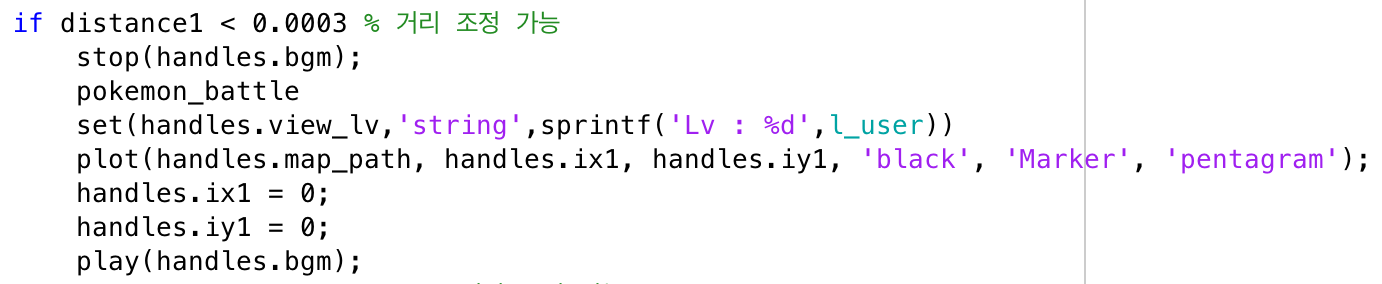
GUI에서 ‘포켓몬 선택’ 버튼을 누르면 실행되는 부분이다. 메뉴에서 7종류의 포켓몬을 선택할 수 있고, 각각의 포켓몬 상태 값, 해당 포켓몬의 이미지를 GUI에 출력해준다. 또한 그 값들을 global 변수에 즉시 적용시켜주도록 만들었다.



GUI에서 ‘GPS 선택’ 버튼을 누르면 실행되는 부분이다. uigetfile 함수를 이용해서 NMEA 정보가 기록된 .txt 파일을 불러올 수 있게 했다. 파일을 불러오면 (2)에서 보았던 ‘NMEAread\_lite’ 함수를 이용해서 필요한 정보들을 읽어오도록 한다. 아래쪽에 보이는 handles.size 변수는, 반복문을 얼마만큼 돌려야 하는지 알려주기 위해 넣은 변수이다.

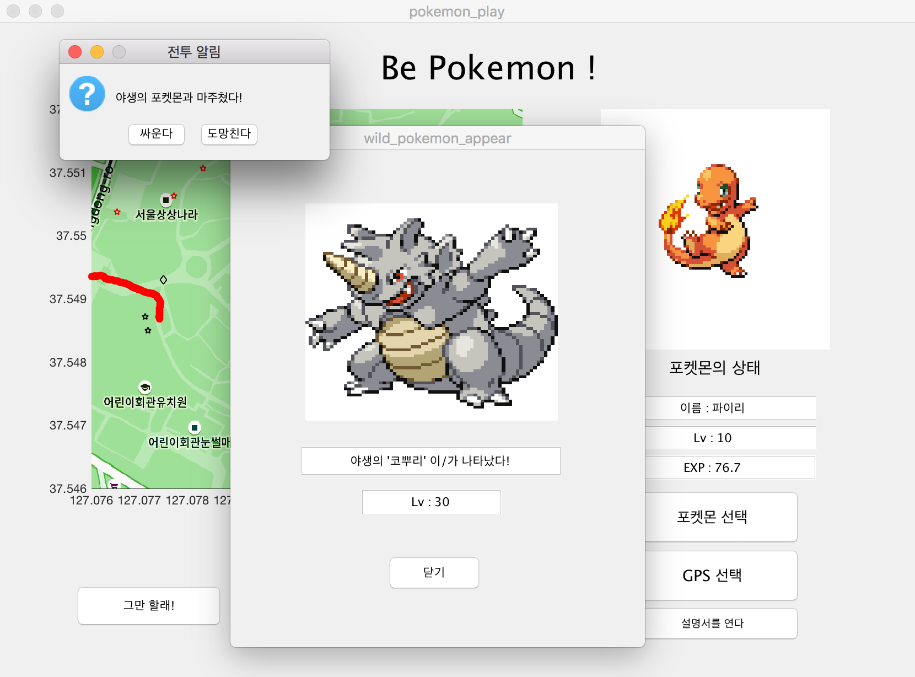


이처럼 for문을 통해 위치를 반복 출력해서 마치 실시간으로 움직이는 듯이 플레이어의 움직임을plot 해주도록 만들었다.

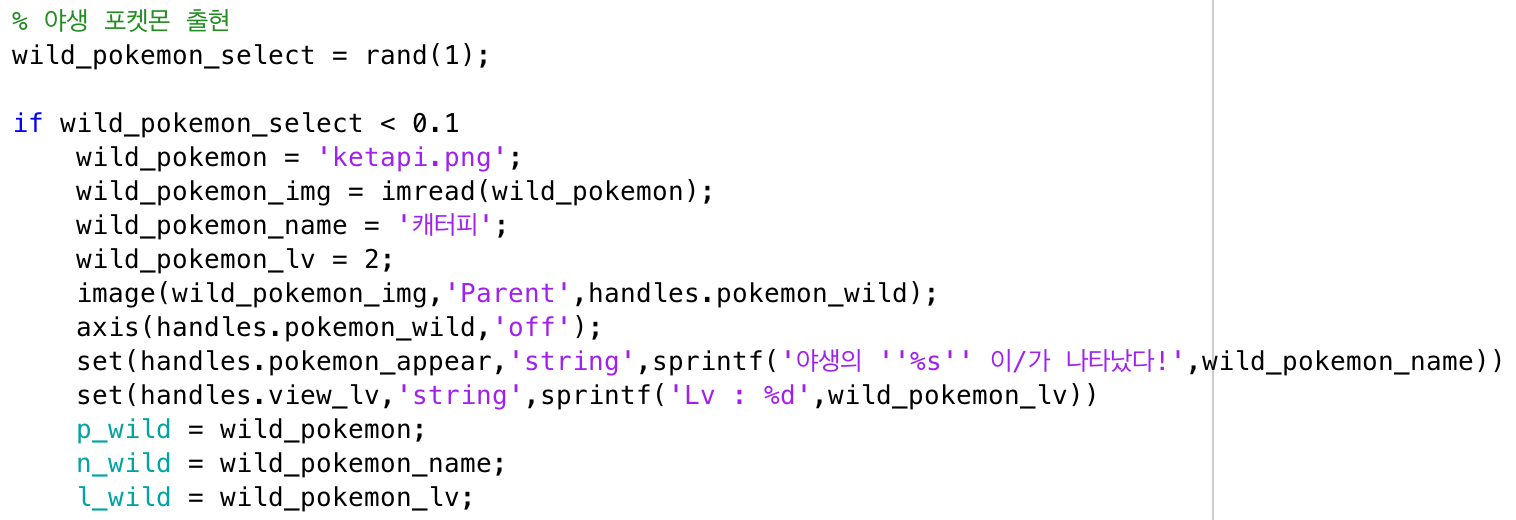


이 부분은 아이템, 야생 포켓몬 오브젝트와 만났을 때 실행되는 부분이다. ‘distance1’은 현재 플레이어의 위치와 오브젝트의 위치 사이의 거리를 의미한다. 따라서 이 거리가 일정 거리 이상 가까워지면 배틀 또는 아이템 이벤트를 실행하도록 했다. 그리고 이벤트가 일어나면 포켓몬의 상태가 변하기 때문에 global 변수 값을 바꾸어 주도록 했다. 또한 한번 만난 오브젝트에서는 다시 이벤트가 일어나지 않도록 위치를 지도 범위 밖으로 바꾸어 버리고, 마커 색을 검정으로 바꾸어 주도록 했다.

(5) wild\_pokemon\_appear.m

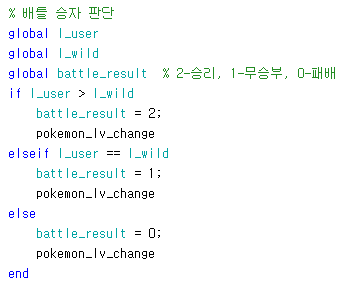


이 파일은 플레이어가 ‘야생 포켓몬’ 오브젝트와 만났을 때 실행하는 프로그램이다. 나타난 야생 포켓몬의 상태와, questdlg 함수로 포켓몬과 싸울지 도망갈지를 선택하도록 하는 메시지를 출력 한다.



플레이어의 포켓몬 선택 부분과 같이 global 변수에 각각의 야생 포켓몬 상태 값을 넣어주었다. 또한 rand 함수를 이용해서 야생 포켓몬이 랜덤으로 나타나도록 만들었다.

(6) battle\_engage.m



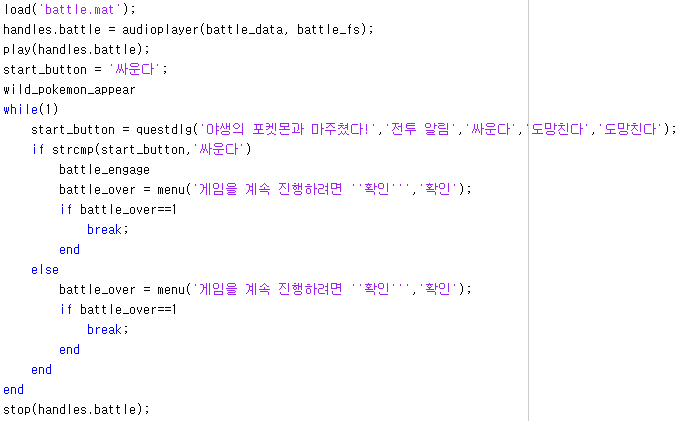
야생 포켓몬과의 배틀이 시작되면 실행하는 파일이다. 플레이어와 야생 포켓몬 각각의 상태 값이입력된 global 레벨 변수를 비교해서 승패를 가리도록 만들었다.

(7) pokemon\_lv\_change.m



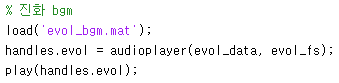
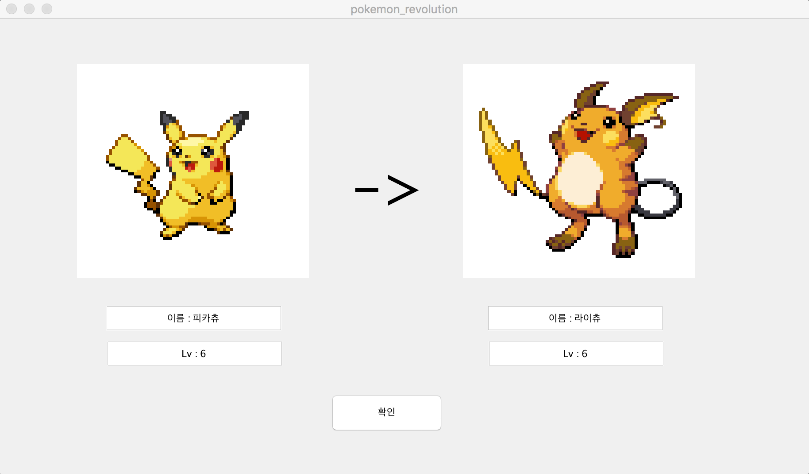
배틀 종료 후의 결과를 출력해주는 파일이다. global 변수로부터 배틀의 승패를 판단하고, 결과에 따라 플레이어의 global 상태 값을 변화시키도록 만들었다.

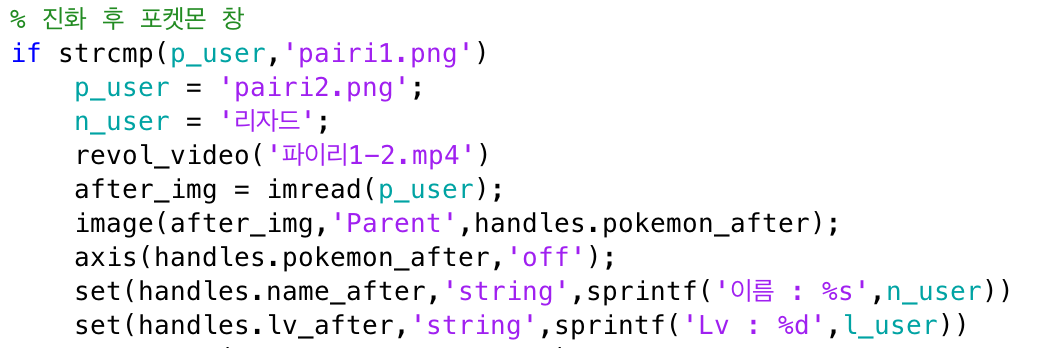
(8) pokemon\_battle.m



야생 포켓몬이 출현했을 때, 배틀 배경음을 재생해주면서 싸울지 말지 여부를 묻는 메시지를 출력한다. ‘싸운다’ 를 선택하면 ‘battle\_engage.m’ 파일을 실행하고, ‘도망친다’ 를 선택하면 실행을 멈추고 계속 게임을 진행할 수 있도록 만들었다.

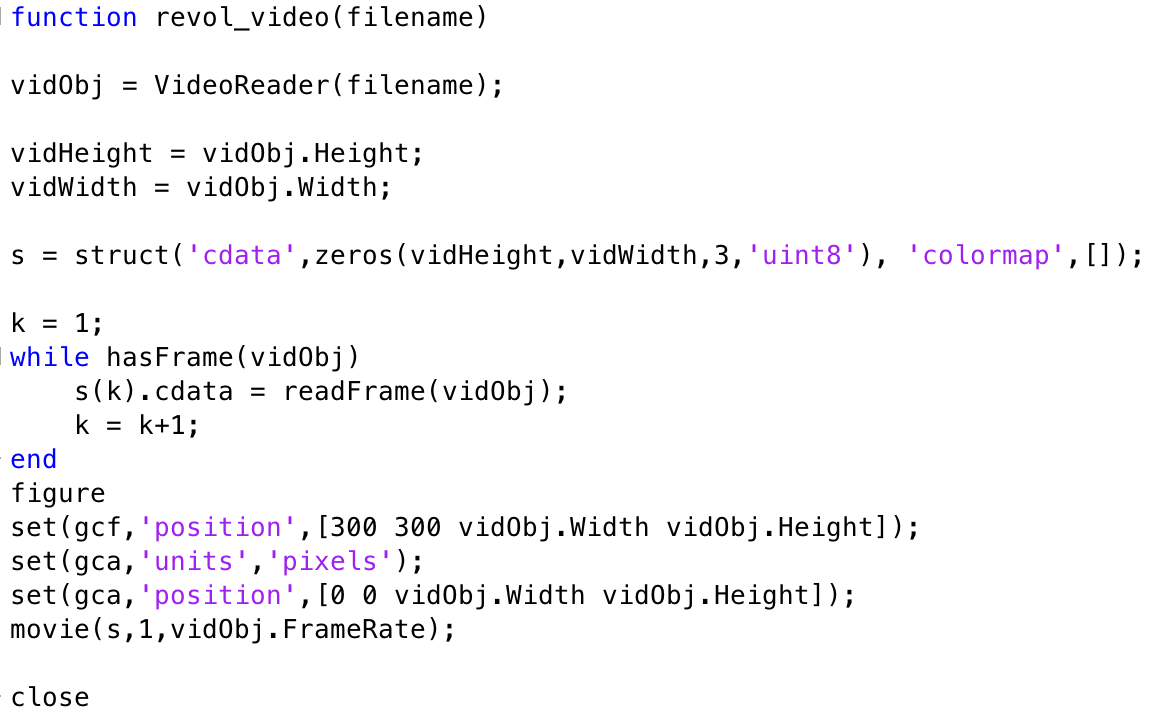
(9) pokemon\_revolution.m





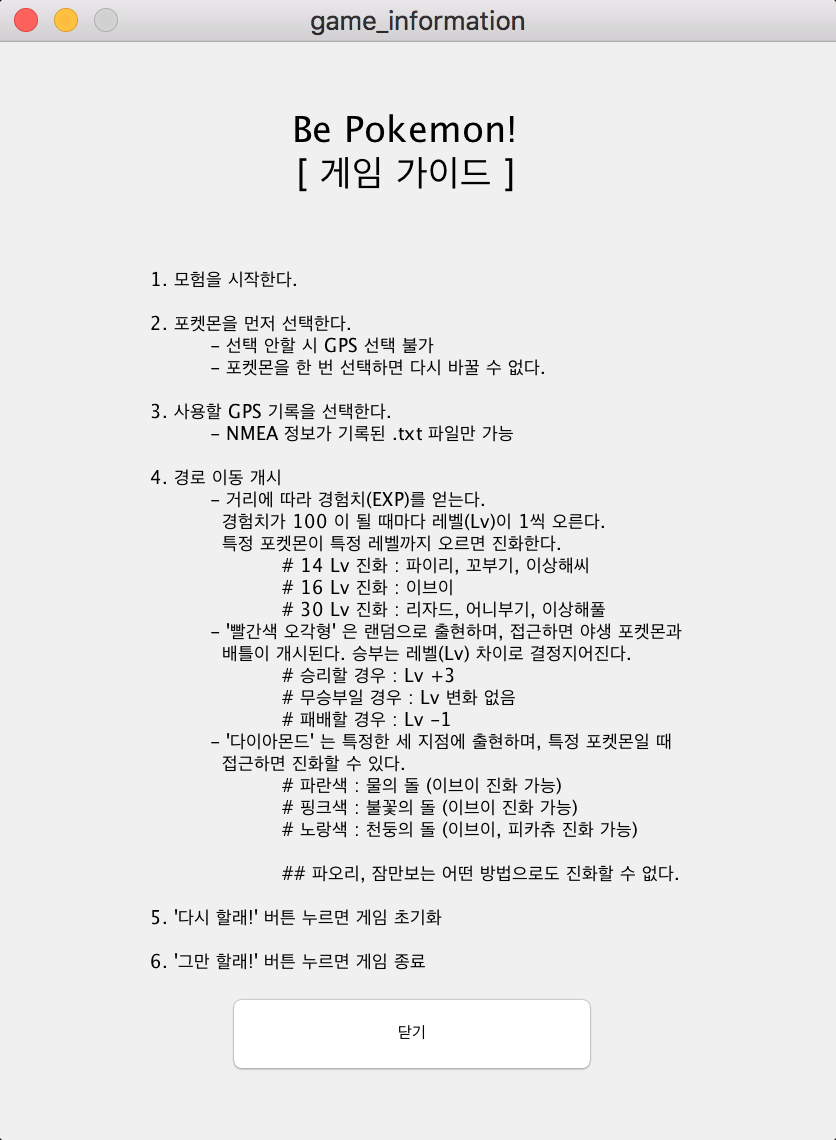
플레이어가 특정 조건을 만족했을 때 포켓몬을 진화 시켜주는 파일이다. 진화 배경음을 재생하는 동시에 revol\_video.m’ 으로 진화영상을 재생시켜준 뒤, global 변수를 이용해서 진화 전, 후의 상태 값을 출력해준다. 진화는 영어로 ‘evolution’이지만, 코딩 당시 착각해서 ‘revolution’으로 만들어버렸다. 수정하기에는 너무 많이 진행되었기 때문에, 오류를 감수하고서 수정할 수는 없었다.

(10) revol\_vedio.m



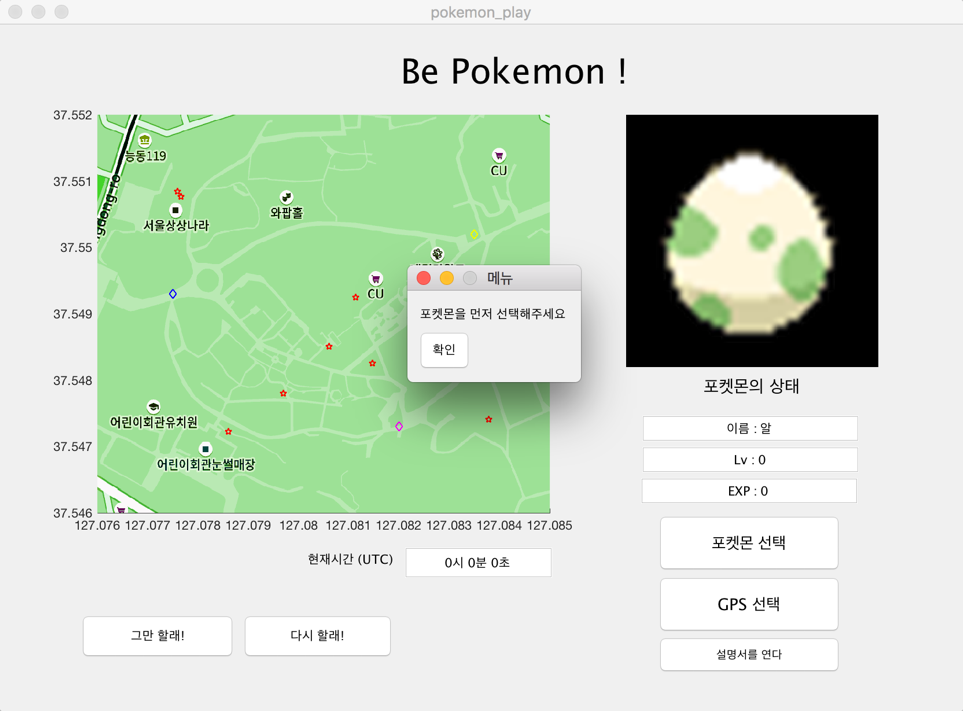
Mp4 파일인 진화 영상을 figure 상에서 재생시켜주기 위한 파일이다. 코드는 mathworks 홈페이지에서 예시로 나온 코드를 그대로 따와서 작성했다.

2.4 프로그램 사용방법



2.5 프로그램 검증

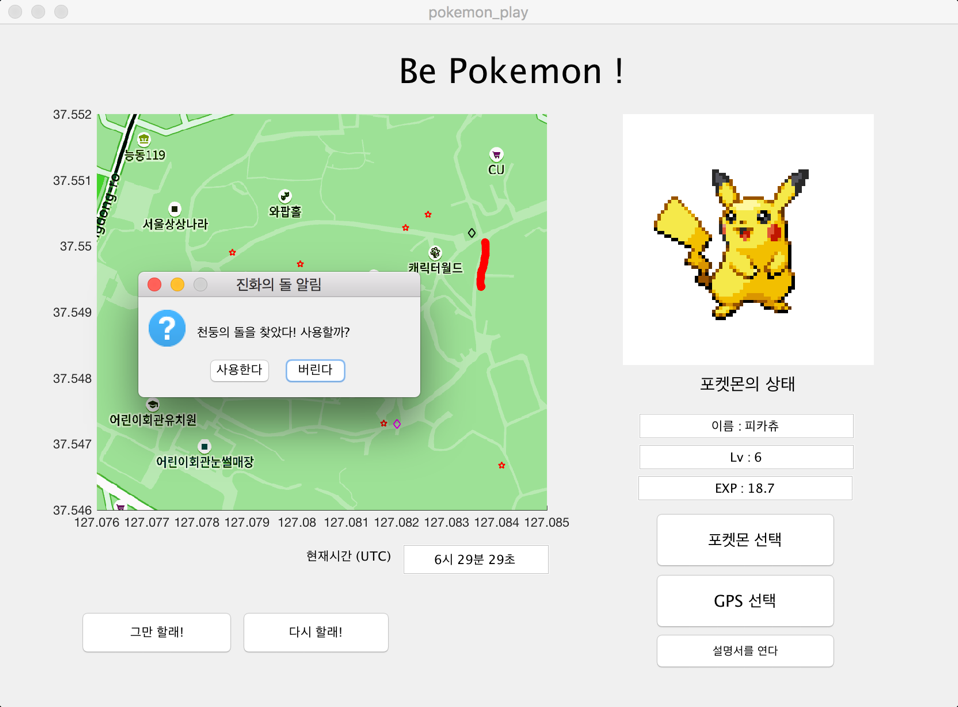
(1) 시작 화면



시작 화면에서 포켓몬을 선택하지 않은 채 ‘GPS 선택’을 누르면, ‘포켓몬을 먼저 선택해주세요’ 라는 안내문이 출력된다.

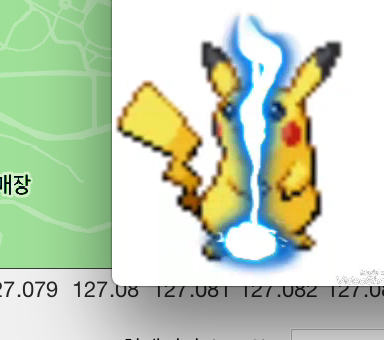
‘GPS 선택’ 밑에 있는 ‘설명서를 연다’ 버튼을 누르면 게임 시작과 동시에 출력되는 설명서를 다시 열어볼 수 있다.

(2) 진화의 돌 시스템

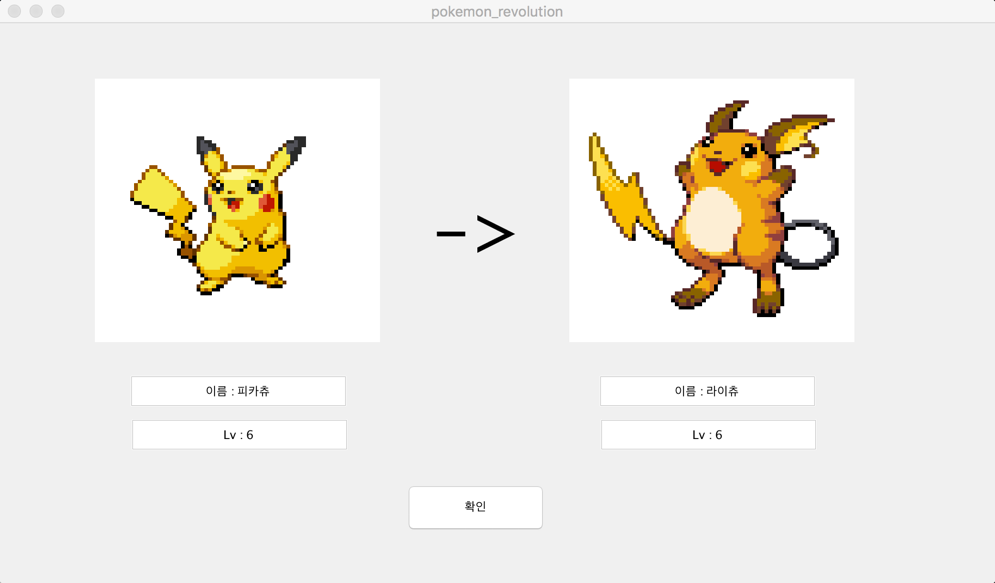


아이템과 조우하게 되면 선택 화면이 뜬다. 사용자는 아이템을 사용하거나, 버릴 수 있다. 아이템을 버리면 그냥 게임이 계속 진행되지만, 사용하면 포켓몬의 종류에 따라서 실행되는 이벤트가 달라지게 된다.

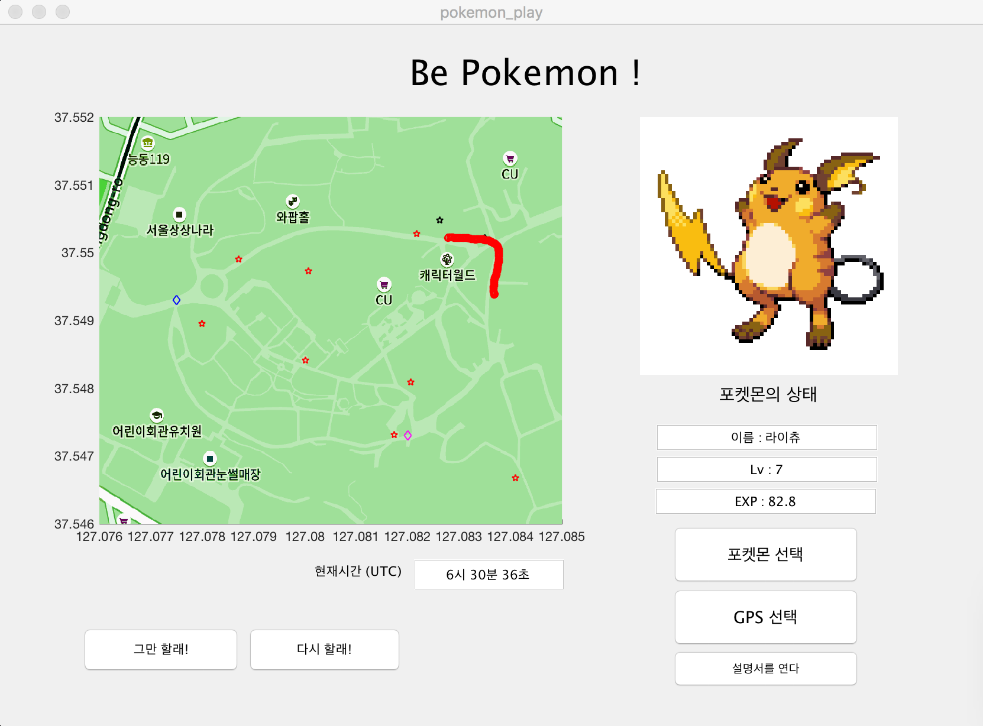
1. 진화가 가능할 경우



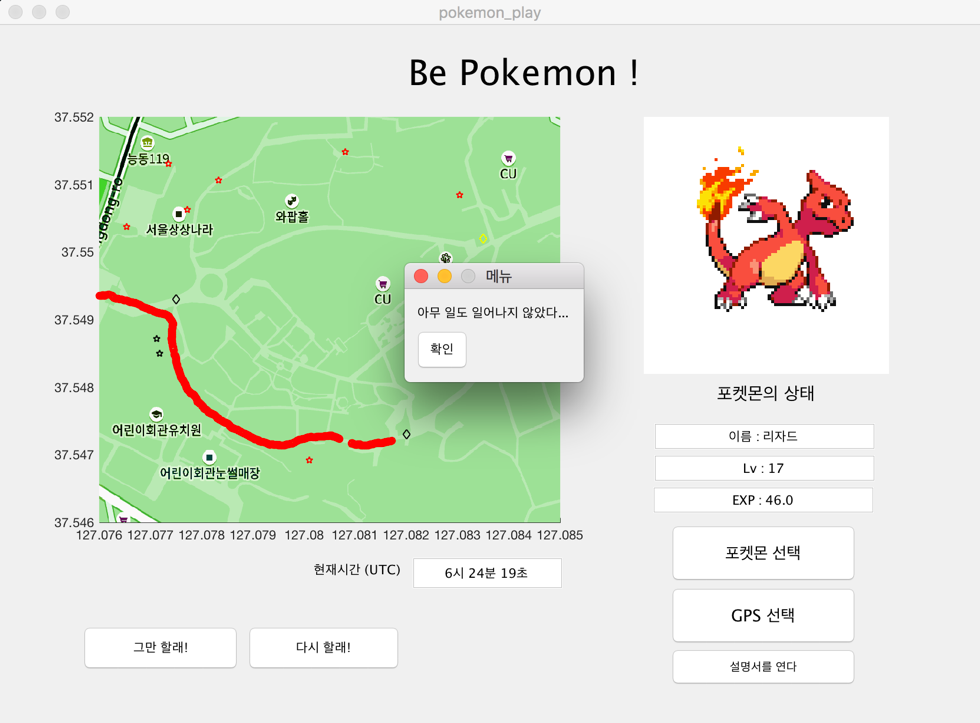
진화 음악과 함께 진화 영상이 재생된다.



영상은 재생 후 자동으로 꺼지며, 꺼짐과 동시에 다음과 같은 진화 결과 창이 출력된다. 확인을 누르면 ‘포켓몬 상태’ 창의 포켓몬 정보들이 진화 후의 정보들로 바뀌어 제대로 출력되는 것을 확인할 수 있다.

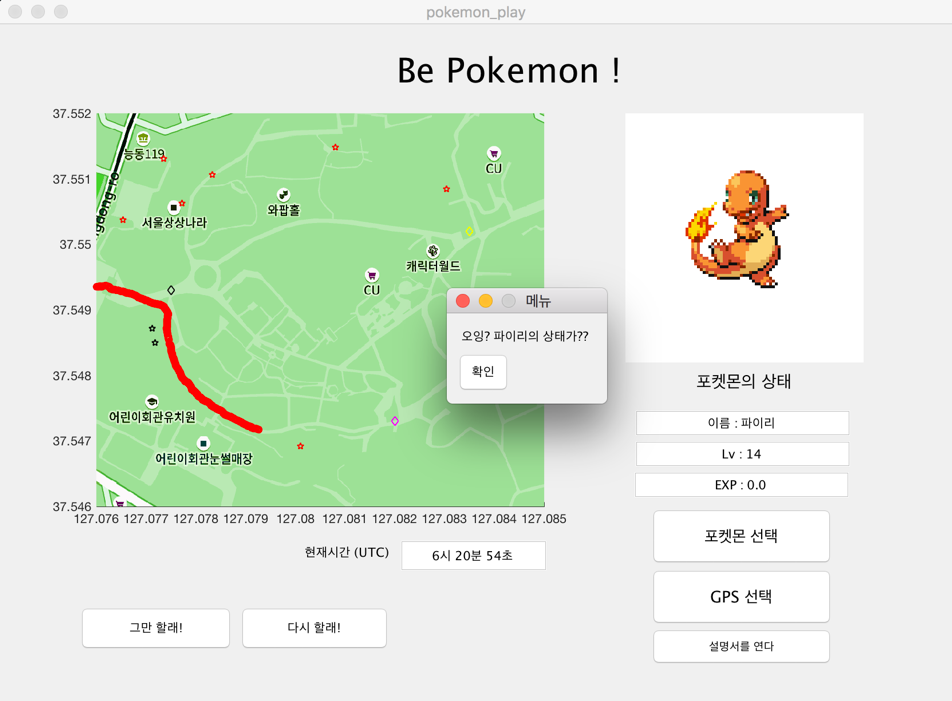


1. 진화가 불가능할 경우



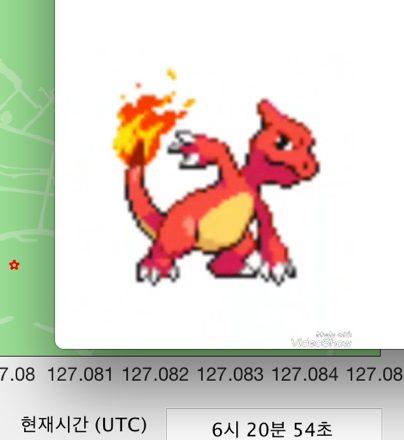
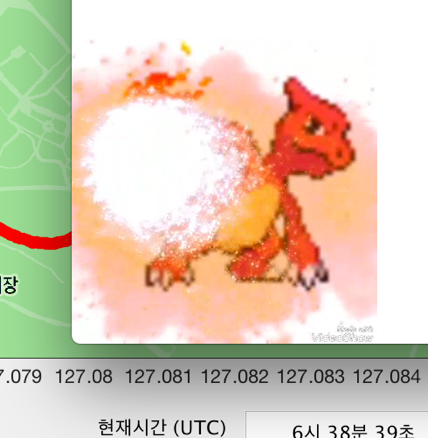
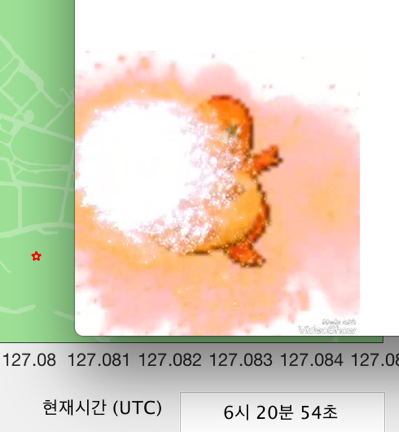
‘아무 일도 일어나지 않았다…’ 라는 글만 출력될 뿐, 정말 아무 일도 일어나지 않는다. 확인을 누르면 게임이 계속 진행된다.

(3) 레벨업 진화 시스템



포켓몬스터에는 진화의 돌 시스템과 함께 레벨 업, 진화 시스템이 있다. 특정 포켓몬이 특정 레벨에 도달하면 자동으로 진화하는 시스템이다. 포켓몬은 NMEA 정보로부터 구한 누적 이동 거리를 통해서 경험치(EXP)를 얻게 되며, 경험치가 100에 도달하면 레벨이 1씩 오르게 된다.

포켓몬이 진화 레벨에 도달하면 ‘오잉? 포켓몬의 상태가??’ 라는 메시지가 출력된다.

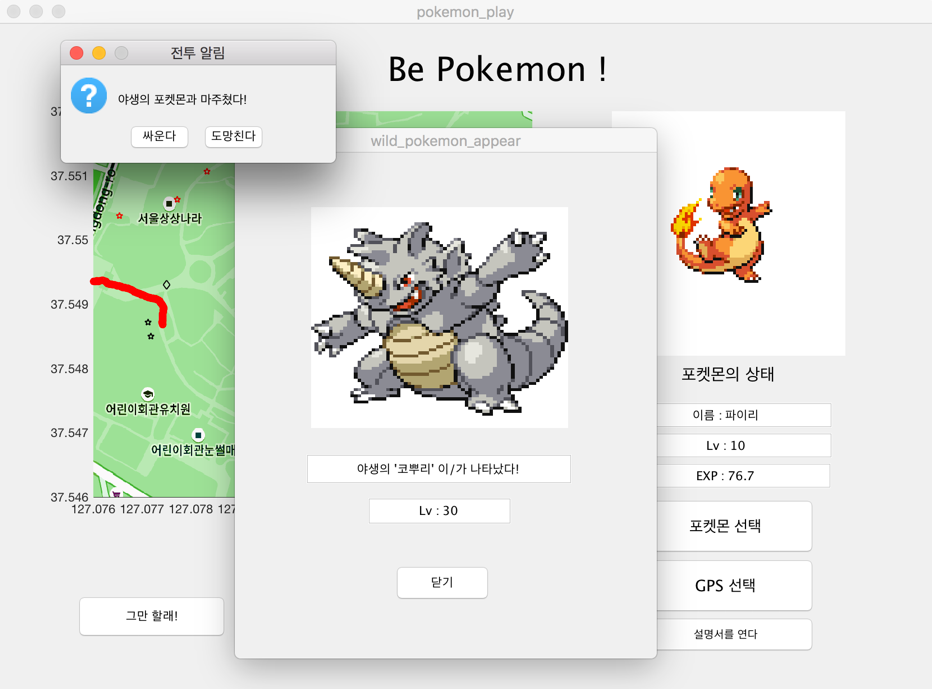


진화의 돌을 사용했을 때와 같이 진화 영상이 재생된 후 진화 결과 창이 뜬다.



이처럼 진화 결과 창까지 제대로 출력됨을 알 수 있다.

(4) 야생 포켓몬과의 배틀



야생 포켓몬과 조우하게 되면 전투 알림과 함께 출현한 야생 포켓몬의 정보가 출력된다. 야생 포켓몬은 총 9종류가 있으며, 랜덤으로 출력된다. 출력된 정보를 확인하고 싸울지 도망갈지를 판단할 수 있다. ‘도망친다’ 를 선택하면 조우한 야생 포켓몬을 무시하고 계속 게임을 진행하게 된다. 하지만 ‘싸운다’ 를 선택하면 ‘포켓몬 배틀’이 시작되게 된다.



배틀이 시작되면 위와 같이 플레이어와 야생 포켓몬을 비교해주는 창이 출력되고, 확인을 누르면 배틀 결과를 볼 수 있다. 배틀 결과는 두 포켓몬의 레벨 차이로부터 결정지어진다.

승리할 경우 (상대방보다 레벨이 높을 경우), 레벨이 3 오른다.

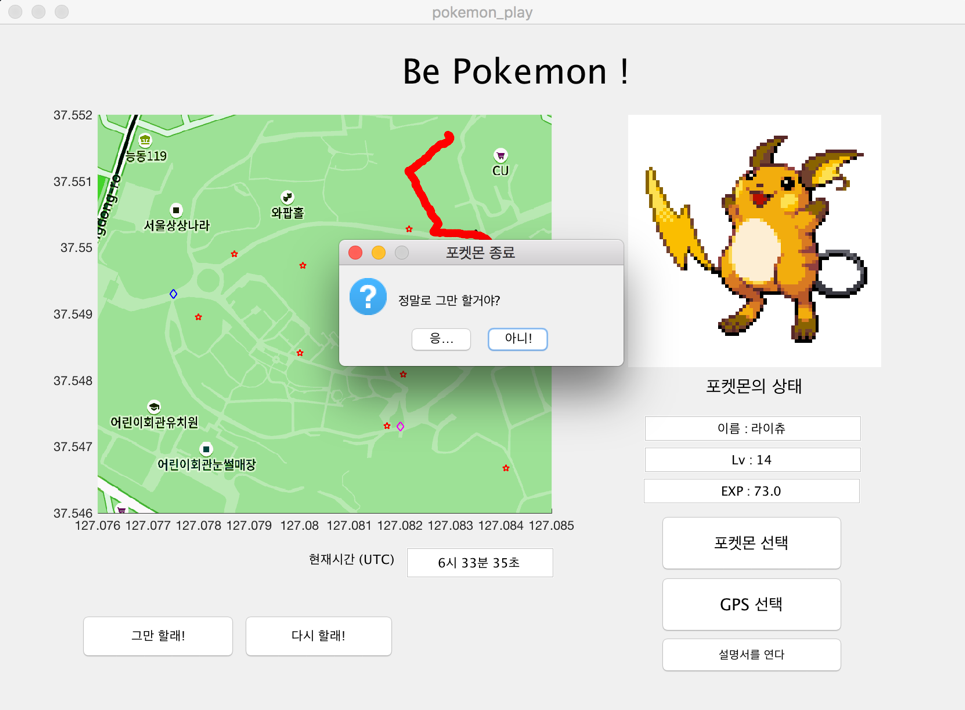
패배할 경우 (상대방보다 레벨이 낮을 경우), 레벨이 1 떨어진다.

무승부일 경우 (상대방과 레벨이 같을 경우), 레벨이 변하지 않는다.

패배할 경우 승리할 경우

배틀 결과가 제대로 출력됨을 볼 수 있다.

1. 

좌측 하단 ‘다시 할래!’ 버튼을 누르면 플레이어의 포켓몬이 알로 초기화되고, 야생 포켓몬 분포가 랜덤으로 재배치 된다.

좌측 하단 ‘그만 할래!’ 버튼을 누르면 ‘정말 종료 할 꺼야?’ 라는 안내문이 뜨고, 확인을 누르면 ‘Be 포켓몬’ 게임 창이 모두 사라지며 종료된다.

이 모든 부분에서 큰 오류 없이 잘 실행됨을 확인할 수 있었다.

2.6 결과 및 분석

‘matlab’ 을 이용하여 제작한 ’BE 포켓몬’은 제작 환경상 포켓몬 go와 완전히 같은 느낌의 게임을 제작할 수는 없었다. 하지만 새로운 게임의 컨셉이 될 수 있는 데모버전을 통해 이 게임의 가능성을 알 수 있었다. 또한 포켓몬스터 라는 주제로 만들어진 많을 데이터들이 쉽게 쌓인 것이 아닌 만큼 그 양과 무게가 매우 컸다. 실제로 게임을 제작 하면서 여러가지 코드와 데이터들이 누적되면서 데이터가 처리되어 진행되는 속도가 느려 지고 원하는 데이터가 표현되지 못하는 경우가 발생하기도 하였다. 이 부분은 사용자의 컴퓨터의 성능에 따라 차이가 발생하지만, 프로그램 제작 환경상 문제 라고도 생각한다. 이를 pause 를 이용하여 데이터 처리를 잠시 쉬게 해주어 문제를 해결 하였다.

데이터가 쌓이고 누적되는데 처리하는 속도는 일정하기 때문에 그 처리를 도와줄 특정한 방법이 없거나 코드가 복잡하게 누적이 되어 있다면 문제는 해결되지 않을 것이다. 따라서 원활한 게임을 제작하기 위해 다양한 캐릭터들과 컨텐츠를 제작하는 것은 매우 중요한 부분임과 동시에 간략화를 통해 프로그램 과부화를 방지하고 무게를 줄여 주는 것 이 중요하다고 생각하게 한 계기가 되었다.

1. **결론**

3.1 결론

사용자를 위한 새로운 컨텐츠의 개발은 생각보다 쉽지 않았다. 제작하는 입장으로서 당연시 생각하는 과정도 이용자는 쉽게 알지 못할 수 있으며, 제작과정에서 자연스럽게 알게 된 사항들도 이용자는 알 수 없다. 이처럼 GUI는 아무것도 알지 못하는 이용자들이 쉽게 이해하여 사용이 가능하도록 해야만 한다. 따라서 제작자의 입장이 아닌 사용자의 입장에서 보다 세밀하고 편리한 구성을 하는 것이 매우 중요함을 알게 되었다.

또한 제작도중 새로운 부분이 삽입되거나 발전하기 위해서 코드의 간략화와 체계적인 이름부여 작업이 필수적으로 느껴졌다. 실제로 어떠한 데이터로 파생되어 새로운 컨텐츠가 제작되는데 그 이름이 체계적이지 못하다면 점점 발전해 나갈수록 난항을 피치 못했을 것이다. 또한 이러한 작업이 다른 사람들과의 협업으로 이루어지는데 코드제작상의 의사소통에 방해를 초래 할 수 있다고 생각이 되었다. 마지막으로 주석으로 코드의 간략한 설명을 통해 쉽게 전달하게 되면서 시간을 절약하면서 코드 내용에 대한 혼동 역시 줄일 수 있게 되었다.

3.2 활용방안 및 기대효과

이 게임은 어린이 대공원의 지도를 기반으로 하여 위치를 설정하고 게임 데이터를 표현했다. 이를 통해 어린이 대공원을 이용하는 사람들, 특히 어린이들이 쉽게 사용 가능하도록 하여 방문자들에게 하나의 또 다른 즐거움을 가질 수 있도록 한다. 또한 게임의 특성상 귀여운 이미지의 캐릭터들은 어린이에게 많은 관심을 받을 수 있을 것이며, 쉬운 플레이 방식으로 누구든지 이용이 가능한 게임으로서 이용층을 넓게 잡을 수 있는 효과를 만들어 낼 수 있을 거라고 생각한다.

이 게임을 특정 지역 혹은 특정 건물에 연관시켜 컨텐츠를 개발해준다면 다양한 시설의 홍보와 더불어 쉽게 사람들이 찾아가지 못하는 지역까지도 폭넓게 이용가능 하도록 도와주게 할 수 있을 것이다. 또한 대체적으로 관심이 집중되는 동물원의 경우 비슷한 느낌의 포켓몬을 통해 자연스러운 학습효과와 주의사항을 전달하는 새로운 방법을 창출해낼 가능성 또한 충분이 내제 되어 있다고 생각한다

3.3 향후 과제

프로토 타입의 ‘BE 포켓몬’을 상용화 하기 위해서는 기존의 데이터를 축적하여 저장하고 다른 지역에서도 사용이 가능 하도록 발전 시켜야 할 것 이다. ‘matlab’ 만으로 데이터를 실시간으로 활용하는 데에 한계가 존재하기 때문에 다른 프로그램과 연동하여 새로운 어플리케이션 환경을 구축하여 새롭게 발전 시켜야 할 것이다. 또한 게임의 자유도를 이용하여 다른 사용자와의 대전 및 캐릭터 교환이 가능하도록 한다. 게임 배경이 되는 스토리를 추가하여 게임의 흥행 요소를 추가적으로 만들어야 할 것이다. 이 게임을 실제로 스마트폰과 연동하여 실시간 데이터를 통해 플레이가 가능한 게임으로 발전 시킬 수 있다면 처음 추구했던 목표치에 충족하도록 발전 시킬 수 있을 것이다.

1. **기타**

4.1 조원 기여도

4.2 수업 관련 느낀 점 및 건의사항

4.3 참고 자료

Mathworks 홈페이지 함수 가이드 (<https://kr.mathworks.com/help/matlab/functionlist.html)>

Mathworks 홈페이지 유저 커뮤니티 (https://kr.mathworks.com/matlabcentral/answers/52064-displaying-multiple-images-with-a-loop)

프로그래밍 정보 검색 사이트 (https://stackoverflow.com/)

이미지 출력 관련 자료 (<http://blog.naver.com/chansung0602/220781642045>)

실시간 plot 관련 자료 (http://engi-agora.tistory.com/41)