

Futásteljesítmény mérő alkalmazás

Jónás Eszter Laura

ML9MF9

Horváth Milán

B1V655

Kovács Levente

BK29F1

Futásteljesítmény mérő: A feladat egy olyan mobil alkalmazás megvalósítása, amely asszisztál futás során (pl. figyelmeztetés időtartamra, alacsony sebességre, megállásra), illetve lehetővé teszi a teljesítmény utólagos megtekintését (átlagsebesség, vertikális változások, útvonal térképes alapon, stb.).

Fejlesztői dokumentáció

Leírás

A célunk egy olyan mobilalkalmazás készítése, ami futás közben képes asszisztálni és motiválni a felhasználót. Ennek az alkalmazásnak a segítségével időintervallum és távolság alapján folyamatosan nyomon követhetjük az utunkat, szinten tarthatjuk a sebességünket, hogy a kívánt idő alatt elérjük célunkat.

Platform:

Mobileszközök, okostelefonok

Támogatott operációs rendszerek:

Android, iOS

Használt technológiák:

React Native: egy nyílt forrású mobil applikációs keretrendszer, ami a Facebook által lett készítve. Ennek segítségével Android, iOS, macOS, Web, Windows operációs rendszerekre vagyunk képesek fejleszteni alkalmazásokat és lehetővé teszi a fejlesztőknek, hogy React keretrendszert használjanak a natív platform képességeivel.

Expo: Egy keretrendszer és platform univerzális React alkalmazásokhoz. Előnye, hogy könnyű tesztelni éles eszközök, hisz egyből a saját eszközünkön jelenik meg az általunk megírt alkalmazás.

Redux: A Redux egy JavaScript appokhoz szánt alkalmazás szintű "kiszámítható" állapot tároló, ami segít a fejlesztőknek egy konzisztens viselkedésű alkalmazás megírásában.

Egyéb fontosabb használt csomagok:

- **expo-location:** a folyamatos pozíció változtatás követéséhez használt eszköz.
- **react-native-paper:** az alkalmazás stílusozásához használt eszköz.
- **react-native-maps:** a térképes megjelenítéshez használt eszköz.
- **react-native-speedometer:** a sebességmérő szimulálására használt eszköz.
- **asyncStorage:** a permanens tároláshoz.
- **react-native-chart-kit:** diagram készítés a futás adataiból.

Futtatás

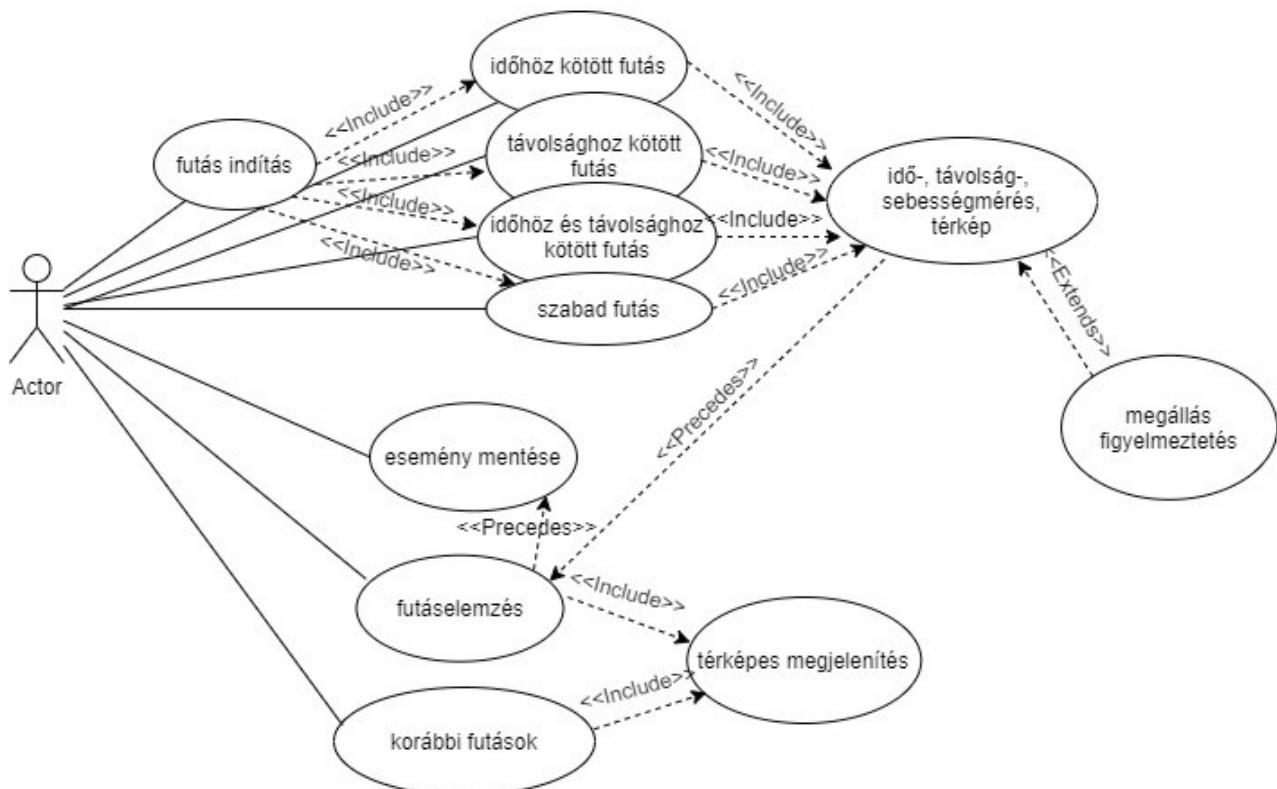
Az alkalmazás futtatásához az Expo Go mobil app szükséges.

Funkciók

1. Az alkalmazás megnyitását követően a "Get started!" screen-re irányítja a felhasználót.
2. A képernyő alján egy navigációs fül található, ahol a felhasználó a 2 alapvető képernyő között navigálhat tetszőlegesen
3. A két alapvető képernyő a "Get started" illetve a "Your runs" fül alatt található.
4. A "Get started" képernyőn a felhasználó beállíthatja a tervezett futásának paramétereit, például a futás típusát: szabad futás, időhöz kötött futás, távolsághoz kötött futás, időhöz és távolsághoz kötött futás.
5. Miután kiválasztotta a futás típusát, megjelennek a futáshoz szükséges egyéb paraméterek, például időhöz kötött futás esetén az időintervallum választó bemenet. Távolság alapú futás esetén a távolságot kell megadnia a felhasználónak, és értelemszerűen ha mindkettő szerint fut, akkor mindkettőt meg kell adni. Szabad futás esetén nem jelenik meg effajta extra paraméter.
6. Ezen az oldalon elhelyezkedik egy olyan gomb, amivel a beállított paraméterek alapján megkezdheti a felhasználó a futást egy új képernyőre viszi őt eközben. Ezen a képernyőn megjelenik egy stopperóra, ahol az idejét tudja figyelni, kiolvashatja a közelítő sebességét, valamint, hogy eddig mennyit futott. Ezen kívül megjelenik egy sebességmutató, ahol nézheti, hogy az átlag sebesség (idő és távolság alapú futás esetén pedig a kívánt átlagsebesség) alapján mennyivel fut, valamint egy térkép ahol nyomon követheti futását.
7. Miután befejezte futását, a megfelelő gomb lenyomásával leállíthatja a folyamatot. Ekkor kérhet elemzést a programtól a felhasználó, ahol megkapja az átlagsebességét, a távolságot amennyit futott, a maximális sebességét, és egy térképen megjeleníti a futását. A térképen egyenletesen ellenőrzőpontok segítségével további adatokat olvashat le a futásának részleteiről az adott pontban.

8. Miután végzett az elemzéssel, a felhasználó kérheti hogy elmentse az adott futás vizsgálatát a helyi tárolóba.
9. A másik fő képernyőn a "Your runs"-ban megtekintheti a felhasználó az elmentett eredményeit és ezeket újra megvizsgálhatja a térképen (mint ahogy a futás elemzésénél is).
10. Az alkalmazás futás közben figyelmeztet az esetleges céldő közeledtére 20%-kal a vége előtt.
11. Az alkalmazás futás közben figyelmeztet az esetleges céltáv elérésének közeledtére 5%-kal a vége előtt.
12. Az alkalmazás figyelmeztet ha a kívánt sebesség (esetleges átlagsebesség) alatt teljesít a felhasználó.

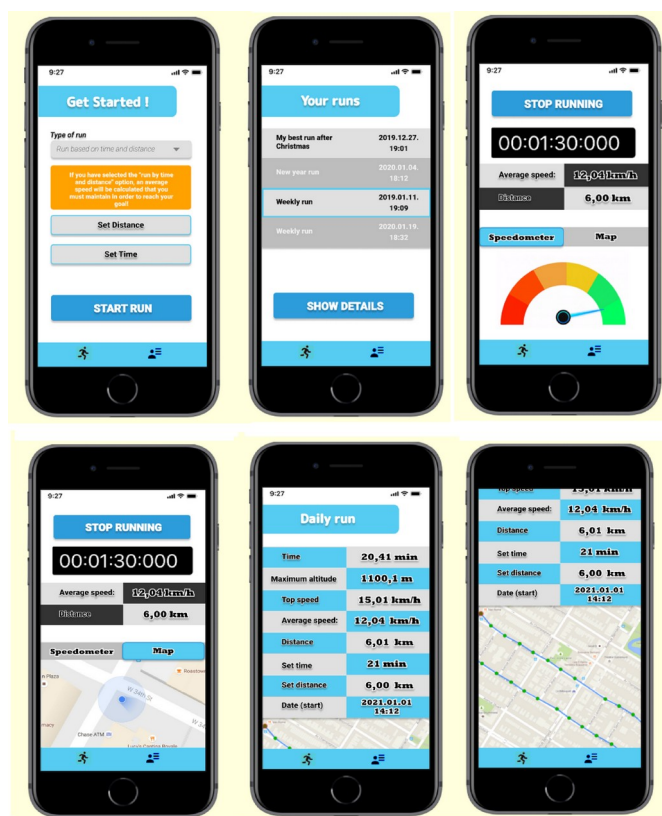
Használat - eset diagram



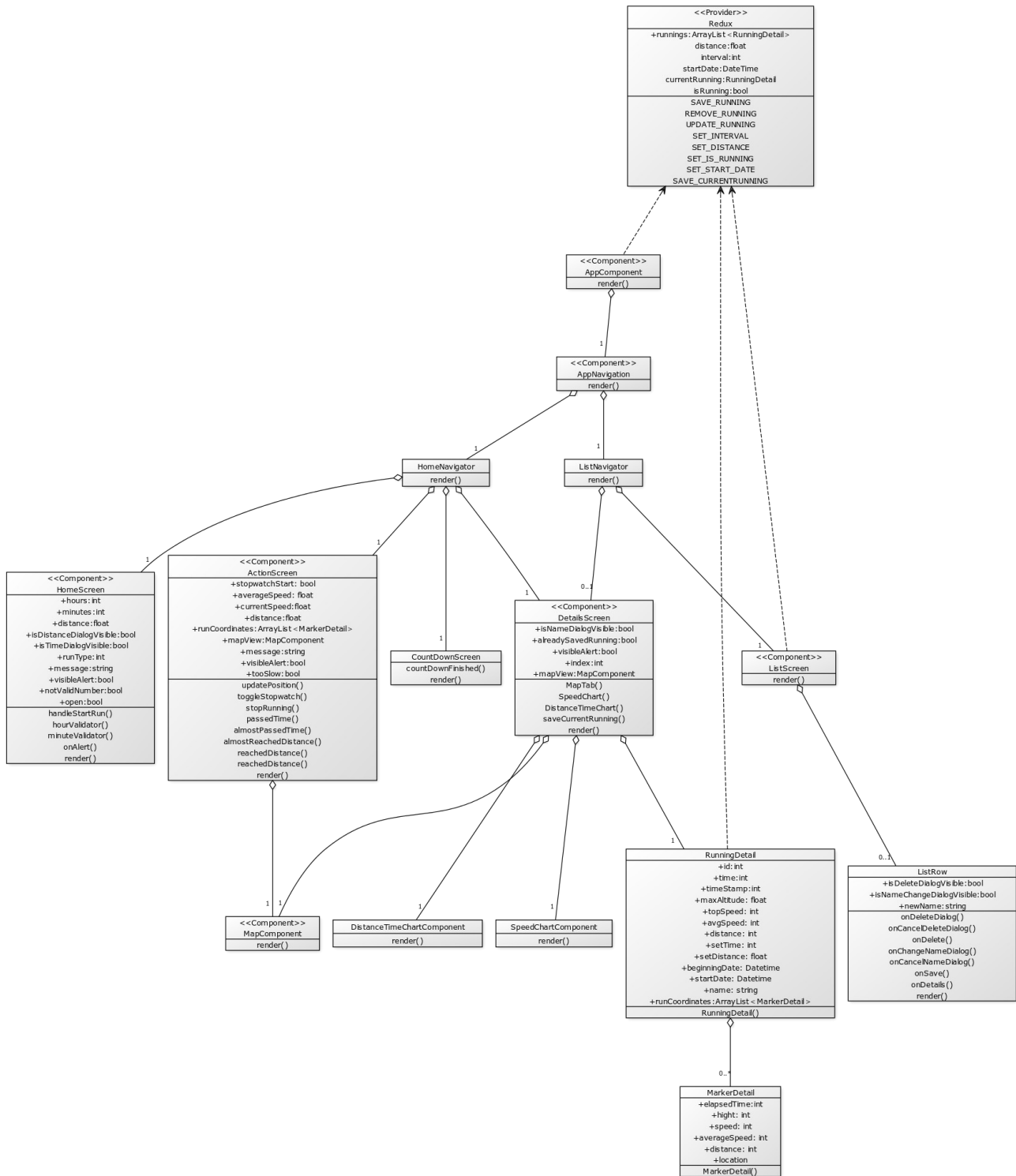
Az alkalmazás felületi terve

Az alkalmazás alapvetően 4 oldalt tartalmaz, ebből 2 érhető el közvetlenül a képernyő alján lévő menüből (1. , 2.)

1. **"Getting started"**- Ezzel nyitódik meg az alkalmazás miután megnyitottuk. Itt kezdheti el a felhasználó a futását, ahol beállíthatja hogy milyen típusú futást akar kezdeni, és ez alapján beállíthatja a szükséges opciókat.
2. **"Your runs"**- Itt találja a felhasználó kilistázva az elmentett futásait névvel és dátummal ellátva. Egy futását kiválasztva és a "Show details" gomb lenyomásával átnavigálja a következő (3.) oldalra a felhasználót a megfelelő adatokkal.
3. **"Details"**- Az adott futás adatait listázza ki és jeleníti meg térképen, ahol bizonyos pontoknál további részleteket kaphat a futásáról.
4. **"Action view"**- A futás közben ezt a képernyőt látja a felhasználó, itt két alnézetet is megtekinthet ahol egy sebességmérőn nézheti, hogy az átlagsebességhez képest mekkora a sebessége, valamint a térképen követheti végig mozgását. Mindemellett a futásának jelenlegi adatait is nyomon követheti.



Az alkalmazás osztálydiagramja



Felhasználói dokumentáció – használati útmutató

A program futtatás után a főképernyőre dob. Ezen a képernyőn kiválaszthatjuk a futással kapcsolatos célkitűzéseinket.

- **Run based on time and distance:** idő és távolság alapú futás. Kötelező paramétereinek szükséges megadni értéket, **Set distance**, **Set time**.
- **Run based on time:** Idő alapú futás. Megválaszthatjuk, hogy mennyi ideig szeretnénk futni. Kötelező paraméterének – **Set time** – szükséges értéket adni.
- **Run based on distance:** Távolság alapú futás. Ebben a módban kötelezően meg kell választanunk a céltávolságot **Set distance** paraméternek megadva.
- **Free run:** Szabad futás opció, ha ezt választjuk korlátlanul futhatunk.

A - **Start run!** - gombbal indíthatjuk el a futást.

Futás akció képernyő

Visszaszámlálás után az akcióképernyőt láthatjuk. Megjelenik számunkra egy sebességmérő, egy stopper, mely másodperc alapon számlálja az eltelt időt és egy térkép, amelyen megjelenik az aktuális pozíciónk. A felületen továbbá látható az átlagsebességünk **Average Speed** illetve a megtett távolság **Distance**. Ha lassan futunk, vagy megálltunk a program figyelmeztet bennünket, hogy lassan haladunk **You are too slow!** felirattal.

Amennyiben a távolság vagy idő alapú futást választottuk, a program a céltávolság elérésekor, és 95% teljesítése esetén is figyelmeztet bennünket, hogy sikerült elérnünk, illetve közeledünk a cél eléréséhez. Az idő alapú futás esetén a program az idő leteltének 80%-ánál figyelmeztet bennünket hogy hamarosan letelik az idő, illetve a cél elérésénél ismételtén, hogy lejárt az idő.

A **Stop run!** gombbal leállíthatjuk a futást ekkor befejeztük, és a program átnavigál bennünket egy másik képernyőre.

Futás összegző, és elemző képernyő – **Your run**

Ezen a képernyőn láthatjuk futásunk eredményességét. Legfelül a mentéshez adhatunk meg nevet, **Name of my run** ahol megválaszthatjuk az elnevezést. Ezalatt egy táblázat található a futásunk eredményeivel, melyek a következőket jelentik:

- **Time:** Futás alatt eltelt idő.
- **Maximum altitude:** Maximum tengerszint feletti magasságunk
- **Top speed:** Maximális elért sebességünk a futás során.
- **Average speed:** Az átlagsebességünk a futás során.
- **Distence:** A távolság melyet futásunk során megtettünk.
- **Set time?:** A kiválasztott célidőt. Abban az esetben látható ha adtunk meg célidőt.
- **Set distance?:** A kiválasztott céltávolság. Abban az esetben látható ha adtunk meg céltávolságot.
- **Date (start):** A futás kezdetének időpontja.
- **Number of stops:** Megállások száma, a megtett út során hányszor álltunk meg.

Ezalatt láthatóak az elemzőpanelek.

- **Map:** Térkép, amely jelzi a megtett útvonalunkat. Egyes jelzésre rányomva látjuk az aktuális adatokat.
- **Speed chart:** A sebességünk változását jelzi az idő függvényében, ezzel könnyen leolvashatjuk a gyorsulásunkat.
- **Achivment:** A célidő és távolság esetén megjelenik hogy a kívánt távolságból mennyit sikerült elérnünk.

Ezen az oldalon végül a **Save running!** feliratot látjuk amellyel menthetjük a futásunk eredményeit.

Futásokat megjelenítő oldal – **Your runs**

A képernyő alján található **Your runs** fülre nyomva navigálhatunk erre az oldalra, ahol a korábban mentésre került futásaink listáját láthatjuk, melyekből egyet kiválasztva legördül egy menü melynél a következő lehetőségek közül választhatunk.

- **Rename running:** Futás átnevezése, rányomva megváltoztathatjuk a futásunk nevét.
- **Delete running:** Futás törlése. Ezt választva eldobhatjuk a futást a mentett elemek közül.
- **View details:** A korábbi futásunk eredményeit tekinthetjük meg az előző részben leírt **Your run** oldalon, melyre navigál bennünket az alkalmazás.

Tesztelés

A tesztelést egységtesztekkel végezzük. A tesztek minden segédeljárásra elvégezzük amelyek visszatérési értékekkel rendelkeznek, illetve az `updateLocation` eljárásra, amelynek segédváltozóit teszteljük. Az Expo jest tesztkörnyezet segít bennünket a tesztelésben.

calculateAvg

Három esetet vizsgálunk ebben a tesztben.

Teszteset	Paraméterek	Elvárt eredmény
1	Dist: 1km, time: 3600000ms	1
2	Dist: 10km, time: 3600000ms	10
3	Dist: 1km, time: 360000ms	10

GetCopiedLocation

Ebben a fázisban ellenőrizzük, hogy az utolsó tömbben eltárolt location értékei másolva lesznek-e a paraméterben megadott aktuális pozícióba. Az eredmény json minden elemét ellenőrizzük.

Bemenő paraméterek:

Tömb:

```
[{"latitude":0, "longitude":0, "altitude":1, "accuracy":12, "altitudeAccuracy": 21  
"heading":10, "timestamp":1000}];
```

Aktuális pozíció:

```
'{"coords": {"latitude":100, "longitude":100, "altitude":100, "accuracy":100,  
"altitudeAccuracy":100, "heading":100}, "timestamp":2000}'
```

Teszteset	Json elem	Elvárt eredmény
1	latitude	0
2	longitude	0
3	altitude	1
4	accuracy	12
5	altitudeAccuracy	21
6	heading	10
7	timestamp	2000

GetDistance

Ebben a tesztesetben az eredményt centiméter pontossággal várjuk. Vizsgáljuk a helyes működést a tesztesettel.

Teszteset	Json elem	Elvárt eredmény
1	'[{"latitude":0, "longitude":0}, {"latitude":0.00001, "longitude":0}, {"latitude":0.00003, "longitude":0}]'	Az első és utolsó elem távolsága háromszor akkora mint az első és második elem közti távolság.

GetDistanceOfLastElements

Ebben a fázisban a tömb hét elemének távolságait vizsgáljuk. A tömb utolsó x elemének távolságát hasonlítjuk össze az adott elem és az utolsó elem távolságával. A távolság lineárisan növekedik így egyeznie kell a kettő eredménynek.

Teszteset	Teszt paraméterek	Elvárt eredmény
1	from Lat.: 0.00005, long.: 0 to Lat.: 0.00006, long.: 0	getDistance(testJSON[5],testJSON[6])
2	from Lat.: 0.00004, long.: 0 to Lat.: 0.00006, long.: 0	getDistance(testJSON[4],testJSON[6])
3	from Lat.: 0.00003, long.: 0 to Lat.: 0.00006, long.: 0	getDistance(testJSON[3],testJSON[6])
4	from Lat.: 0.00002, long.: 0 to Lat.: 0.00006, long.: 0	getDistance(testJSON[2],testJSON[6])
5	from Lat.: 0.00001, long.: 0 to Lat.: 0.00006, long.: 0	getDistance(testJSON[1],testJSON[6])
6	from Lat.: 0, long.: 0 to Lat.: 0.00006, long.: 0	getDistance(testJSON[0],testJSON[6])

GetHHMMSS

Az idő HHMMSS formátumban. Minden számjegyét megvizsgáljuk.

Teszteset	Idő (ms)	Elvárt eredmény	Kapott ered.
1	1000	00:00:01	00:00:01
2	10000	00:00:10	00:00:10
3	60000	00:01:00	00:01:00
4	600000	00:10:00	00:10:00
5	3600000	01:00:00	01:00:00
6	36000000	10:00:00	10:00:00
7	86400000	00:00:00	00:00:00

countStops

Ebben a tesztben megnézhetjük hogy működik a countStops függvény.

Teszteset	Teszt tömb	Elvárt eredmény
1	[0,0,0,0,0,0]	0
2	[0,1,1,0,1,1]	1
3	[0,1,0,1,0,1]	2
4	[0,1,0,0,0,1]	1

toFixeding

Ebben a tesztben megnézhetjük hogy működik a toFixing (alsó rész) kerekítőfüggvény.

Teszteset	Paraméterek	Elvárt eredmény
1	Num: 1.125, dec: 0	1
2	Num: 1.125, dec: 1	1.1
3	Num: 1.125, dec: 2	1.12
4	Num: 1.125, dec: 3	1.125

updateLocation

A legfontosabb tesztfázis. A teszt 6 részből áll. Ezek a tömbök hosszának ellenőrzése, segédváltozók eredménye, és a segédfüggvények visszatérési értékei.

Segl dtömb

A segédtömb növekedését vizsgáljuk. Ha meghívjuk a függvényt egy adathalmazzal, a tömb mérete növekedik eggyel.

Distance változó

Ebben a tesztetben megnézzük hogy a pozíció változásával a távolság növekedik-e.

Szükségtelen adatok eldobása

Az első két adat az alkalmazás tesztkörnyezetéhez való optimalizálás miatt szükségtelen.

TooSlow változó

Az alacsony haladási sebességet az alkalmazás figyelmeztetéssel jelzi. Ebben a tesztben ellenőrizzük, hogy valóban működik-e.

RunCoordinates tömb

A méret növekedését figyeljük ebben a fázisban, hasonlóan az előző segédtömbbhez. A tesztelési taktika megegyezik az első esetben levőével.