

**Projektmunka 2.**(GKNB\_INTM005)

**Gesztusvezérlés Windows rendszerhez gépi látás segítségével**

**Barszcz Dániel, Szakál Gyula Richárd,  
Varga Balázs**

**Győr, 2024**

[1. Bevezetés 3](#_Toc178598570)

[1.1 Ötlet és motiváció 3](#_Toc178598571)

[1.2 Feladatok és tevékenység-felelősség mátrix 4](#_Toc178598572)

[2. Követelményspecifikáció 8](#_Toc178598573)

[2.1 Kompatibilitási követelmények 8](#_Toc178598574)

[2.2 Funkcionális követelmények 8](#_Toc178598575)

[2.2.1 Parancs hozzárendelése meglévő gesztushoz 8](#_Toc178598576)

[2.2.2 Gesztusvezérlés 9](#_Toc178598577)

[2.2.3 JSON fájl beolvasása 9](#_Toc178598578)

[2.3 Teljesítmény követelmények 9](#_Toc178598579)

[2.4 Megbízhatósági követelmények 9](#_Toc178598580)

# 1. Bevezetés

Dokumentációnkban a Projektmunka I. c. tantárgy keretein belül megfogalmazott mesterséges intelligencia és gépi látás által támogatott, Windows-os rendszerhez készített gesztusfelismerő alkalmazásunk tervezési és fejlesztési folyamatát, valamint a program részletes specifikációját szeretnénk ismertetni.

## 1.1 Ötlet és motiváció

Az ötlet főként a mesterséges intelligencia és a gépi látás, illetve tanulás iránti érdeklődésből alakult ki. Csapatunk tagjai számára a gépi látás által lehetővé tett objektum felismerés kifejezetten hasznosnak bizonyult már korábbi tanulmányaik során is. A múltbeli projektekből és azok sikerességéből, valamint tapasztalataiból kiindulva döntöttünk amellett, hogy ezt a témát fogjuk választani.

A projektfeladat témájának konkretizálása során figyelembe vettük azt, hogy bizonyos személyek valamilyen betegség vagy éppen a munkakörnyezet miatt nem képesek használni egy számítógép hagyományos kezelőszerveit, az egeret és a billentyűzetet. Ennek a problémának a megoldására szeretnénk egy olyan szoftvert fejleszteni a Projektmunka II. c. tárgy keretein belül, amellyel a felhasználó gépi látással támogatott gesztusvezérlés segítségével lehet képes egy Windows-os rendszer vezérlésére, irányítására. A cél tulajdonképpen tehát az, hogy a szoftver használója képes legyen kézmozdulatokkal irányítani a számítógépét és ne kelljen a számára problémát, megerőltetést jelentő bemeneti perifériákat használni. Mindezt megtámogatnánk egy használatot segítő grafikus alkalmazással, amely kvázi egyfajta vezérlőpultként szolgálná a felhasználó érdekeit. A user itt tudja majd beállítani, hogy az egyes kézmozdulatokkal, gesztusokkal milyen utasítást kíván kiadni, majd pedig elvégeztetni.

Saját szemszögünkből tekintve a fő motiváció pedig az, hogy el tudjunk még jobban mélyülni a mesterséges intelligencia és a gépi látás által nyújtott lehetőségek nagyon tág halmazában, valamint közösen megtapasztaljuk egy komplex szoftverrendszer tervezési és fejlesztési munkafeladatait.

## 1.2 Feladatok és tevékenység-felelősség mátrix

|  |  |
| --- | --- |
| **Feladat** | **Résztvevő(k)** |
| Projektvezetés | Szakál Gyula Richárd |
| Frontend fejlesztés | Varga Balázs, Barszcz Dániel |
| Backend fejlesztés | Szakál Gyula Richárd |
| Tesztelés | Barszcz Dániel, Szakál Gyula Richárd,  Varga Balázs |
| Dokumentáció elkészítése | Barszcz Dániel, Szakál Gyula Richárd,  Varga Balázs |
| Prezentáció elkészítése | Varga Balázs |

A projekt megvalósításához szükséges feladatokat nagyobb csoportokba rendeztük, amelyek a feladat sikeres megvalósításához elengedhetetlenek voltak. A projektvezetést a projektcsapat vezetője látja el, ez a tevékenység magában foglalja a konzulens oktatóval való kapcsolattartást, a meetingek szervezését, a csapattagok informálását, a feladatok kiosztását, illetve azok teljesítésének ellenőrzését.

A frontend fejlesztés a szoftver grafikus felületének elkészítését takarja. Az intuitívabb szoftverhasználathoz és a növelt szoftver ergonómiához elengedhetetlen egy modern grafikus felület. Jelen esetben nem volt elegendő egy konzolos felület a program használatához, ezért szükséges volt egy grafikus interfész elkészítése a PyQt python könyvtár használatával. Ehhez az alapot nagy mértékbe egy, a Figma programban készített UI terv adta.

A backend fejlesztés a szoftver fő funkcionalitás megvalósítását jelenti. A program írása több python könyvtár segítségével valósult meg. Ez teszi lehetővé a tényleges működést, az újabb gesztusok felvitelét, az ismertek felismerését és az azokhoz rendelt billentyűparancsok kiadását.

A tesztelésben a projektcsapat minden tagja aktívan rész kell, hogy vegyen. A tesztelés fontos részét képezi a funkcionális tesztek írása, valamint alkalmazása a program megfelelő működéséről való megbizonyosodás végett. AI-projekt lévén a tanító mechanizmus egyfajta tesztként is funkcionál, ugyanis ennek végeztével meghatározható a modell megbízhatósága és pontossága.

A projektdokumentáció elkészítésében szintén minden csapattag közreműködik, a program specifikációit, illetve működését részletező dokumentum a projekt előrehaladásával folyamatosan bővül. A dokumentáció megfelelő alapot kell, hogy biztosítson a fejlesztőknek a későbbi funkcionális bővítések-, illetve hibakeresések esetében, ennek értelmében mindenképpen szükséges egy jól kidolgozott, részletes dokumentáció elkészítése.

A Projektmunka II. c. tantárgy szorgalmi időszak utolsó heteiben zárul az elkészült program bizottság előtti bemutatásával. A prezentációt a tagok által elvégzett feladatok alapján egy személy készíti el, a védésben, az elkészült munka bemutatásában azonban természetesen a projekt aktívan közreműködő személyek mindegyike részt kell, hogy vegyen és az általa elvégzett feladatokat részletezze.

|  |  |  |  |
| --- | --- | --- | --- |
| **Tevékenységek/Nevek** | **Szakál Gyula Richárd** | **Barszcz  Dániel** | **Varga  Balázs** |
| **Frontend fejlesztés, grafikus felhasználói felület tervezése** | J,B,I | V,B,I | V,B,I |
| **Backend fejlesztés és AI megoldások alkalmazása** | V,J,B,I | B,I | B,I |
| **Tesztek írása és tesztelés elvégzése** | V,J,B,I | V,B,I | V,B,I |
| **Adminisztráció, dokumentáció kezelése** | V,J,B,I | V,B,I | V,B,I |
| **Kapcsolattartás az oktatóval** | V,J,B,I | B,I | B,I |
| **Projekt menedzselése** | V,J,B,I | B,I | B,I |
| **Projektfeladat védése** | V,J,B,I | V,B,I | V,B,I |

A tevékenység-felelősség mátrixban (RACI-mátrix) szereplő tevékenységek szinte megegyeznek a már felvázolt feladatokkal, azokon kis mértékű változtatásokat eszközöltünk, illetve ahol szükséges volt, ott kibontottuk, részekre osztottuk az adott feladatot. Az egyes tevékenységek a projekt tagjainak azok kompetenciái alapján kerültek kiosztásra, így mindenki a saját szakterületének, valamint tudásának megfelelő feladatot végez a féléves munka egésze alatt.

A RACI-mátrixban négy különböző feladatkört különböztettünk meg, a jóváhagyásit, a végrehajtásit, a megbeszélésit és az informálásit. A jóváhagyás joga a projektvezetőt illeti meg, a tagok által elvégzett feladatokat ő ellenőrzi és hagyja jóvá. A végrehajtási felelősség azt a tagot érinti, akinek kötelessége az adott feladat elvégzése a projektvezető által megjelölt határidőig. A megbeszélési és informálási kör a csapat minden tagjához hozzárendelésre került, ugyanis kis létszámú fejlesztő team esetében lényeges az információk egymás közötti megosztása és a munkát érintő kérdések közös megvitatása az előrehaladás érdekében.

# 2. Követelményspecifikáció

A dokumentáció egyik legfontosabb részét képezi a különböző követelmények részletes meghatározása és rögzítése, amely elengedhetetlen a projekt sikeres előrehaladása és a program kialakítása, majd későbbi, éles környezetben való (lehetőleg) hibamentes működése érdekében is. A követelményeket négy nagy csoportba soroltuk, jelen fejezetben kompatibilitási-, funkcionális-, teljesítmény- és megbízhatósági követelményekkel foglalkozunk.

## 2.1 Kompatibilitási követelmények

A gesztusvezérlő program alapvetően Windows operációs rendszerhez készül, a jelenlegi verzió más (Linux-disztribúciók, stb.) operációs rendszerekkel nem kompatibilis. A felhasználó szempontjából nincsenek jelentős elvárások a hardverrel és a szoftverrel szemben sem. A user-nek rendelkeznie kell egy Windows operációs rendszert futtató számítástechnikai eszközzel (számítógép, laptop), valamint beépített kamerával vagy mobilapplikáción keresztül használt IP kamerával. A kamerának a gesztus érzékelésénél van jelentősége a programban.

## 2.2 Funkcionális követelmények

### 2.2.1 Parancs hozzárendelése meglévő gesztushoz

A felhasználó a GUI megnyitását követően a „Beállítások” menüpontra kattintva éri el a „Vezérlőpultot”, ahol a megadott gesztusokhoz hozzárendelhet Windows-rendszerben értelmezett parancsokat. Ezt követően szükséges kiválasztani a bemeneti forrást, a kompatibilitásnál már említett beépített-, vagy IP-kamerát. A „Beállítások mentése” gombra kattintva a program a „preferences.json” fájlba fogja elmenteni a felhasználó által választott gesztus-parancs párosításokat, amely alapját képezi a későbbi gesztusok által történő vezérlésnek.

### 2.2.2 Gesztusvezérlés

A felhasználó a GUI megnyitását követően a „Beállítások” menüpont alatt található „Használat” gombra kattintva kezdheti meg az alkalmazás használatát. Ennek előfeltétele a fentebb leírt gesztus-parancs összerendelés, enélkül a felhasználó nem tudja gesztusokkal vezérelni az operációs rendszert, ugyanis alapértelmezés szerinti gesztus-parancs párosítások nincsenek rendelkezésére bocsátva. A felhasználó által előnyben részesített, használt kamera segítségével valósul meg a gesztusvezérlés, az operációs rendszer a JSON fájlba elmentett preferenciák alapján fogja elvégezni a megfelelő parancsot, műveletet. A „Megállítás” gombra kattintással a gesztusalapú vezérlés befejezhető.

### 2.2.3 JSON fájl beolvasása

A felhasználó képes a GUI-n keresztül JSON formátumú fájl kiválasztására, mely tárolja, hogy a program jelen esetben milyen gesztusokat ismer fel. A felhasználónak ezeket előre biztosítjuk a programmal együtt.

## 2.3 Teljesítmény követelmények

A teljesítményt elsősorban a felhasználó által használt eszközök befolyásolják. IP kamera használata esetén, gyenge internetkapcsolat mellett előfordulhat laggolás, illetve a gesztushoz tartozó elvégzendő művelet végrehajtása több időt vehet igénybe. A fejlesztők által meghatározott maximális végrehajtási szintidő 6 mp.

## 2.4 Megbízhatósági követelmények

A projektfeladatunk a mesterséges intelligencián, illetve a gépi látáson alapul, ennek megfelelően a megbízhatóság erősen modellfüggő, illetve nagyban függ a minták számától, változatosságától és a tanítási mechanizmusoktól. Annak érdekében, hogy a modell a lehető legnagyobb megbízhatósággal működjön, a fejlesztőcsapat minden tagjának kezeivel készítettünk felvételeket a gesztusokról és ezeket használtuk fel a tanító algoritmusban a nagyobb pontosság elérése érdekében.

A program hibamentes működését szem előtt tartva, mind a backend-, mind a frontend fejlesztésben igyekeztünk a megfelelő, modern szoftverfejlesztési eszközöket és technikákat alkalmazni. A jól részletezett dokumentáció elkészítése szintén a megbízhatóság érdekeit szolgálja, a későbbiekben esetlegesen felmerülő problémák, hibák okának feltárását, illetve javítását jelentősen egyszerűsítheti.