**Eötvös Loránd Tudományegyetem**

**Informatikai Kar**

**Informatikatudományi Intézet**

**Algoritmusok és Alkalmazásaik Tanszék**

Programozható interaktív geometriai ábrák készítése webes

környezetben

Szerző: Témavezető:

Tóth Botond Bálint Csaba

Programtervező informatikus BSc. Tanársegéd

**Budapest, 2023**

A document with black text and white text

Description automatically generated

Tartalomjegyzék

[1. Bevezetés 1](#_Toc87521353)

[2. Felhasználói dokumentáció 2](#_Toc87521354)

[3. Fejlesztői dokumentáció 3](#_Toc87521355)

[4. Összefoglalás és további fejlesztési lehetőségek 4](#_Toc87521356)

[5. Irodalomjegyzék 5](#_Toc87521357)

[6. Melléklet 6](#_Toc87521358)

# Bevezetés

Ebben a szakdolgozatban foglalom össze az általam készített szoftver leírását, dokumentációját. Illetve szeretném, ha az olvasó beletekintést nyerhetne a gondolatmenetbe, amely alapján a programot elkészítettem. A program különböző részeinek megértéséhez, fontos, hogy átlássuk milyen folyamatok is zajlanak a háttérben.

A program szerves részét a Matlab export képezi, amely egy Matlab-ban létrehozott, kétdimenziós koordinátarendszerből, több függvény segítségével képez először egy SVG fájlt. Ezekután, ha a felhasználó úgy állította be, akkor pedig egy script blokkot is létrehoz, ami interaktívvá teszi, az előzőekben említett SVG fájlt. Tehát végül 3 különböző programozási nyelv fordul elő a programban, mivel a script rész javascript segítségével teszi interaktívvá az SVG statikus elemeit.

Az SVG (Scalable Vector Graphics) egy XML alapú leíró nyelv. W3C által definiált nyílt szabvány. Főleg vektorgrafikák meghatározására használják. Szerencsére az objektum modellje az SVG fájloknak a teljes XML DOM-ot tartalmazza, így könnyen manipulálható például ECMAScript használatával.

Mivel az ECMAScript a JavaScript nyelv alapját képezi, ezért a JavaScript nagyszerű eszközöket nyújt egy SVG interaktivizálására. Ennek segítségével kinyerhető, megváltoztatható a statikus SVG-ből a különböző elemek attribútumai. Ami tökéletes megoldásként szolgál a szoftver megvalósításához.

# Felhasználói dokumentáció

# Fejlesztői dokumentáció

## Probléma specifikációja

A dolgozat témája egy Geogebrához hasonló, geometriai összefüggéseket és algoritmusokat vizualizáló program elkészítése. A mozgatható pontoknak és ezektől függő alakzatok interaktív frissítésének köszönhetően hasznos eszköz oktatásban és kutatásban.

Szakdolgozatom keretében megvalósított program a Geogebrával ellentétben numerikus algoritmusokat használ különbféle kétdimenziós alakzatoknak és azok metszéseinek és egyéb műveleteinek megvalósítása érdekében. Az alkalmazás programozhatóságából adódóan olyan algoritmusok vizualizálhatóak vele, mint a Delaunay háromszögelés, illetve Voronoi cellák megjelenítése vagy távolságfüggvényen való gömbkövetés. A dolgozatomban bemutatott alkalmazás kódgenerálási módszerek segítségével interaktív webes formába konvertálja a Matlab scriptben definiált alakzatokat. Ehhez több algoritmust kell hatékonyan megvalósítani például egyenesek, poligonok, körök és Bézier görbék közötti metszéspontok számítására.

A szakdolgozatom célja, hogy egy könnyen kezelhető platformfüggetlen interaktív geometriai vizualizációs appleteket tudjon biztosítani webes felületeken.

## Módszerek, foglamak

A program első körben egy kiválogatást hajt végre. Itt a Matlab-ban létrehozott kétdimenziós koordinátarendszer elemein iterál végig a függvény. A Geomatplot-ban alapvetően az elemek két csoportra vannak osztva. A két csoport a moveable, azaz a mozgatható elemek. Ebből mindössze kettő típusosztály létezik. Van egy mozgatható pont, illetve egy poligon. A poligonnak a csúcsai mozgathatók, és ezzel maga az alakzat is. Illetve vannak a dependent, azaz a más alakzatoktól függő elemek. Mivel a dependens és a mozgatható típusosztályban is van például pont, illetve polygon, ezért a típusosztályok nevében magyar jelöléssel (hungarian notation) különböztetjük meg, hogy milyen típusúak. Viszont számunkra a statikus SVG létrehozásánál nincs szükség, hogy külön kezeljük ezeket az elemeket. Így ezt a két adag elemet majd összefűzzük egy listába.

Első lépésként a program egy kiválogatást hajt végre. A kiválogatás közben, struktúrákat jönnek létre. A struktúrák több különböző változóból állnak. A struktúrákban tárolt változók különbözőek az alapján, hogy milyen típusosztály adatait tartalmazza. Az összegyűjtött adatok később a statikus SVG elemeinek adatait fogják alkotni. A kiválogatás közben minden struktúrát ugyanabba a listába kerül. Mivel első körben az exportált kép nem képes külső interakciókat kezelni, ezért itt nem számít, hogy milyen módon kéne kezelniük a felhasználói bemeneteket.

Második lépésként a program az összegyűjtött adatokból, létrehoz string-eket amelyekbe SVG tag-ek kerülnek. Ezek az tag-ek azok, amik az XML kód értelmezésekor geometriai elemekként jelennek meg. Illetve itt a legalkalmasabb transzformálni a koordinátákat, úgy, hogy az SVG megjelenítési formáinak is megfeleljen. Mint minden koordinátarendszerben, a Matlab-ban is az első síknegyedben, helyezkednek el azok a pontok, amiknek mind x, illetve y koordinátája pozitív számok. Viszont a grafikus programok esetén a nulla-nulla pozíció a bal felső sarokban van, és a megszokottól eltérően az y koordináta pozitív és negatív oldalai felcserélődnek. Tehát a koordinátarendszert x tengelyre tükrözötten kell kezelnünk. Ezért, hogyha a koordinátákat egyszerűen eredeti formájukban használnánk a kép fejjel lefele jelenne meg. Ennek kiküszöbölésére egyszerűen tükrözni kell az y koordinátákat. Ez viszont azt eredményezi, hogy az exportált kép a képernyőn kívül fog elhelyezkedni. Ezért a felhasználóbarát élményért, egy eltolást is végrehajtunk. Nagyítást is végrehajtunk, ami a koordináták méretétől függ.

Szerves része a programnak a különböző callback-ek. A callback-ek olyan függvények, amelyeket adatként eltárolunk későbbi felhasználásra és egy másik függvény hívására fut le. Matlab-ban, illetve Javascript-ben is rengeteg callback-et használunk állapotváltozások figyelésére vagy kezelésére. A Geomatplot-ban több előre definiált callback van. Ezek például szögfelezőket, kör középpontokat számolnak. A felhasználási lehetőségek tárháza végtelen. A Matlab világában kifejezetten jól használhatók alakzatok megjelenítésénél, ahol több elem egymástól függ. És szerencsénkre a Javascript-ben is előszeretettel használják. Ezért tökéletes lehetőséget kínálnak ezek a függvények arra, hogy Javascript-el tegyük interaktívvá az exportált SVG-t.

Ha már alakzatok figyeléséről esett szó //event listener, eventek, mutation observer, DOM, matematikai dolgok talán,

# Összefoglalás és további fejlesztési lehetőségek

# Irodalomjegyzék

# Melléklet