

# Áramkör szimulátor

## 1. Projekt leírása

Az áramkör szimulátor egy didaktikai eszköznek készülő program. A felhasználó egy könnyen kezelhető grafikus felhasználói felület segítségével kommunikál a programmal, ahol pár egérgattintással összeállíthatja a vizsgálni kívánt áramkört. Ezután egy virtuális kapcsoló átbillentésével elindíthatja az áramkört. Virtuális mérőműszerek “képernyőjén” olvashatja le a tudni kívánt értékeket. A program a Kirchoff törvények alapján számol, s mivel az alkalmazás didaktikai célú, ha a felhasználó látni kívánja, megjeleníthetők a Kirchoff egyenletek is.

## 2. Csapattagok

Az Áramkör szimulátor (a továbbiakban az ÁSZ rövidítést használjuk) nevű alkalmazás tervezésén és implementálásán a következő csapat dolgozik:

Mag Csaba
Mátis Márta-Timea
Szabó László
Szotyori Alpár
Tamás Levente
Vrencsán Szilárd

## Projektterv

Az ASZ felhasználó, grafikus felület segítségével, összeállít egy áramkört. A következő áramköri elemeket fogja szimulálni:

- Ellenállás (egyszerű és változtatható).
- Kondenzátor (polaritásfüggetlen) (egyszerű és változtatható).
- Áramforrás (csakis egyenáramú).
- Kapcsoló (egyállású).
- Fogyasztó (égő).

- Mérőműszerek (A, V,  $\Omega$ ).
- A komponenseket összekötő huzal.

A fentiekből kiderül, hogy az ASZ csak és kizárólag passzív elemekből álló, egyenáramú áramkört szimulál. A továbbiakban az áramköri elemeket komponensnek fogjuk nevezni.

## 1.1 Felhasznált technológia:

C++, grafikus felület Qt.

## 1.2 Struktúra:

A program két nagy részből fog állni: grafikus illetve a matematikai háttér. Úgy a grafikus, mint a matematikai részben minden, a felhasználó által beszúrt komponensre létrejön egy objektum. Ezeknek az objektumoknak egy közös ősük van, egy absztrakt osztály, azonban a komponenseknek ebből az osztályból származtatott különböző osztályok felelnek meg.

A matematikai részben létrejött objektum tartalmaz minden olyan információt, ami a számoláshoz, és a gráf felépítéséhez szükséges, míg a grafikus részben létrejött objektum a megjelenítéshez, illetve a manipulációhoz szükséges információkat “tudja”.

Tehát, a **matematikai objektumok** komponensenként a következőket tudják:

- Ellenállás: érték(változtatható esetében a maximális illetve az aktuális értéket is).
- Kondenzátor: kapacitás(aktuális- és maximális érték).
- Áramforrás: feszültség, belső ellenállás, maximális teljesítmény.
- Kapcsoló: állapot(be- vagy kikapcsolt).
- Fogyasztó: teljesítmény, ellenállás, működési feszültség.
- Mérőműszer: ellenállás.

- Komponenseket összekötő huzal: tulajdonságainak értéke elhanyagolható, tehát erre a komponensre nem jön létre külön objektum. Ehelyett minden eddig leírt komponens tudja kivezetéseire (végpontjaihoz) kapcsolt más komponenseket.

A **grafikus részben** létrejött objektumok ismerik a megjelenítéshez, ábrázoláshoz szükséges koordinátákat, illetve a komponensek kivezetéseinek koordinátáit.

Ezekből következik, hogy a grafikus objektumok tudják a nekik megfelelő matematikai objektumokat, a matematikai objektumoknak viszont nem szükséges tudniuk a nekik megfelelő grafikus objektumot. Az értékek megjelenítéshez illetve beállításához a grafikus objektumok csak meghívják a matematikai objektumok getter és setter függvényeit.

A menüsorban lesz még megnyitás, és mentés menüpont. A mentés esetében az áramkör rajza lementhető belső- illetve kép (BMP) formátumban.

## **1.3 Fejlesztési stratégia**

Először létrehozuk a fent leírt alapot, és a programot frissítési lehetőségekkel látjuk el. Így, amikor megszületik az alkalmazás azon verziója (amit gyorsan a főprogram után meg is írunk), amely már dolgozik félvezetőkkel és a váltakozó áramot is tudja kezelni könnyedén lehet majd hozzáadni a régi verzióhoz.