# Simulace elektrostatického pole

## Popis řešení

Tato aplikace vizualizuje elektrostatické pole vytvořené různými konfiguracemi bodových nábojů.

Uživatel si může vybrat jeden ze 4 přednastavených scénářů, které se liší rozmístěním a velikostí

nábojů.

**Hlavní funkcionalita zahrnuje:**

* Vykreslení elektrostatického pole
* Zobrazení bodových nábojů jako barevných kruhů, jejichž velikost odpovídá velikosti náboje
* Animaci pohybu testovací sondy po kružnici
* Interaktivitu s náboji
* Vykreslení mapy intenzity pole

**Omezení a zjednodušení:**

* Je k dispozici pouze 5 přednastavených scénářů bez možnosti úprav.
* Jedná se pouze o 2D vizualizaci.

## Popis klíčových algoritmů

Výpočet síly elektrostatického pole v daném bodě vychází z Coulombova zákona:

function CalculateForceOnProbe(probePosition, charges):

electricField = (0, 0)

for charge in charges:

r = probePosition - charge.position

rMagnitude = length(r)

if rMagnitude == 0:

continue

fieldContribution = r / rMagnitude^3

electricField += fieldContribution \* (charge.charge \* k)

return electricField

Kde k je Coulombova konstanta a charge.charge reprezentuje velikost náboje v daném bodě.

Animace pohybu testovacího náboje po kružnici je realizována pomocí aktualizace jeho pozice v

závislosti na čase:

function UpdateProbePosition():

angle = angularVelocity \* velocityMultiplier \* time

probePosition = (cos(angle), sin(angle))

Kde angularVelocity a velocityMultiplier určují rychlost pohybu náboje.

Color color = IntensityToColor(fieldVector.Magnitude());

for (int y = yStart; y < yEnd; y++) {

int rowOffset = y \* stride;

for (int x = xStart; x < xEnd; x++) {

int index = rowOffset + x \* 4;

if (index < pixelData.Length - 4) {

pixelData[index] = color.B;

pixelData[index + 1] = color.G;

pixelData[index + 2] = color.R;

pixelData[index + 3] = 255;

} } }

Tento kód převede velikost vektoru intenzity elektrického pole na barvu, kterou poté použije k vyplnění bloku pixelů v bitmapě, přičemž nastavuje jednotlivé RGB složky a alfa kanál pro každý pixel v rámci definovaných hranic.

Spuštění a ovládání aplikace  
Aplikace se spustí z konzole příkazem:  
dotnet bin/UPG\_semestralka.dll <cislo\_scenare> -g<X>x<Y>

Kde -g<X>x<Y> je volitelný parametr kde za *<X>* a *<Y>* jsou celočíselné hodnoty udávající rozteč vzorků mřížky v ose x a y v pixlech

**Ovládání:**

* Tlačítka " 0" až " 4" umožňují jednoduchý výběr jednoho z přednastavených scénářů.  
  Zaškrtávací tlačítka od „0x“ do „2x“ nastavují rychlost sondy vzhledem k dané hodnotě.
* Kolečkem myši lze zvětšovat/zmenšovat velikost náboje
* Držením náboje a posunem myši se náboj posouvá po poli.
* Dvojklikem myši se přidá sonda měřicí intenzitu. Otevře se druhé okno, ve kterém je vykreslován graf závislosti velikosti vektorového pole v místě sondy na čase od okamžiku umístění sondy.

## Známé nedostatky

* Mohlo by být užitečné přidat možnost ručního umístění nábojů a jejich editaci.
* Vizualizace by mohla být rozšířena o zobrazení ekvipotenciálních čar.
* Aplikace by mohla být rozšířena o možnost exportu vykresleného stavu do obrázku nebo  
  PDF.