# Vizualizace elektrostatického pole

Cílem této práce je vytvořit vizualizace intenzity elektrostatického pole v bodě s proměnlivým časem a pozicí. Na začátku se projekt vytvoří pomocí build.cmd, který vytvoří složku bin s kompilovaným projektem. Následně pomocí příkazové řádky zavoláme run.cmd s parametrem a aplikace se spustí. Podle zvoleného parametru se spustí konkrétní scénář, který bude následně vizualizován.

# Program.cs

Jedná se o vstupní bod aplikace, nastavuje prostředí, načítá vizuální styly a vytváří instanci MainForm, které předává argument z terminálu.

# MainForm.cs

Vytváří instanci DrawingPanel, která umožňuje kreslení. Předává parametr do konstanty vytvořené v drawingPanel

# DrawingPanel.cs

Atributy třídy:

* String parameter – předaný parametr, který rozhoduje který scénář se spustí
* Int m\_Start – ukládá aktuální čas aplikace ve chvíli spuštění
* Float elapsed – proměná, který se s časem mění, slouží pro animaci
* Double power – ukládá intenzitu el. Pole
* Int[,] position – dvojrozměrné pole, ve kterém se ukládají pozice nábojů
* Int[] charge – pole, do kterého se ukládá síla jednotlivých nábojů

## Konstruktor DrawingPanel()

Vytváří instanci Timer, který se stará o sledování času aplikace, díky tomu je možné vytvářet animaci.

## Metoda Timer\_Tick(object sender, EventArgs e)

Při každém zavolání překreslí plátno.

## Metoda OnPaint(PaintEventArgs e)

Hlavní funkce, ve které se vykresluje vše co je na plátně. Proměnná scale slouží k upravení velikosti objektů vzhledem k velikosti okna, je zde zvolena konstanta, ze které scale následně vypočítán. V metodě jsou vytvářeny dvě instance, Charge a Probe. Jednotlivé scénáře jsou řešeny pomocí switch funkce, kde parametr rozhoduje o tom, který scénář se spustí. V jednotlivých scénářích jsou pak vytvořené objekty, a do polí positions a charges jsou přiřazené příslušné hodnoty, do power je pak přiřazena hodnota intenzity pole.

## Metoda OnResize(EventArgs eventargs)

Po změnění velikosti okna překreslí plátno.

# Charge.cs

Tato třída slouží pro vykreslení náboje.

## Metoda add(Graphics g, int charge, int[] pos, float scale)

Metodě je předán grafický kontext, aby bylo možné v metodě volat funkce pro vykreslení, dále síla náboje, která pomáhá určit vizuální velikost náboje. Pole pos předává informaci o pozici náboje.

Nejdříve se vytvoří Cesta, do které je přidána Elipsa. Následně je vytvořen region, do kterého je cesta poté přidána. Tento přístup byl zvolen, aby se mohl využít PathGradientBrush pro výplň náboje. Barva brushe se přiřazuje podle toho jestli je náboj nabitý pozitivně či negativně. V prostředku náboje je taktéž vypsána informace o jaký naboj se jedná, tj. jakou velikost má.

# Probe.cs

Tato třída slouží k výpočtu intenzity elektrostat. Pole a vykreslení vektoru směru intenzity a vypsání hodnoty intenzity v konkrétním bodě, který se pohybuje po kružnici.

Atributy:

* Float[] vectorPosition – Nese v sobě x1,y1,x2,y2 souřadnice pro vykreslení grafu.

## Metoda calculate\_Intensity(int[] charges, int[,] positions, float elapsed)

Počítá aktuální hodnotu v bodě X, Y. Následně se vytvoří pole pro uložení x2, y2 bez přenásobení size a scale.

Pomocí for cyklu, který projíždí přes všechny náboje, jsou pak počítání Coulombovým zákonem souřadnice, které jsou průběžně přičítány do pole. Proměnná nMagnitude nese hodnotu intenzity, kterou pak celá metoda vrací. Do pole vectorPosition[] jsou předány souřadnice vektoru.

## Metoda addVector(Graphics g, float scale, double result)

Metoda přebírá grafický kontext, aby bylo možné volat funkce pro vykreslení, dále scale, aby byla možná reakce na resize okna a result, který nese hodnotu intenzity pole.

Přenásobí souřadnice vektoru velikostí a scale, aby souřadnice seděly do velikosti světa. Následně je spočten směr a velikost vektoru. Na konci se vykresluje vektor, sonda a hodnota intenzity.