# Scenario

Třída Scenario představuje správce objektů, které implementují rozhraní INaboj, a IProbe a umožňuje jejich přidávání, odebírání, získávání jejich pozic a kreslení celého scénáře na grafickém plátně.

## Metody:

* **GetCharges**
  + Vrací aktuální pole charges obsahující objekty typu INaboj.
* **Save**
  + Vrátí řetězec pro uložení
* **EmptyCharges**
  + Vyprázdní seznam nábojů
* **CreateProbe(PointF center, float radius, float anglePerSecond)**
  + Vytvoří sondu podle zadaných parametrů
* **CreateProbe(PointF center, float radius, string anglePerSecond)**
  + Vytvoří sondu podle zadaných parametrů
* **AddCharge(INaboj naboj)**
  + Přidává nový náboj do pole charges
* **Load(string[] lines, float startTime)**
  + Načte se seznamu řádek objekty to scénáře
* **RemoveCharge(INaboj naboj)**
  + Odebere náboj z pole na základě předaného objektu naboj.
* **RemoveCharge(int id)**
  + Odebere náboj z pole na základě jeho ID.
* **RemoveProbe(int id)**
  + Odebere sondu se zadaným ID.
* **GetCharge(int id)**
* Vrátí instanci náboje se zadaným ID
* **GetProbe(int id)**
  + Vrátí instanci sondy se zadaným ID
* **EmptyProbes**
  + Vyprázdní seznam sond
* **Move(float x, float y)**
  + Posune výřez o x a y
* **ZoomIn a ZoomOut**
  + Přibližuje a oddaluje
* **Draw(Graphics g, float width, float height, int startTime)**
  + Vypočítá výřez k zobrazení tak, že vezme nejmenší obdélník obsahující všechny náboje, rozšíří ho ve všech směrech o 25 %, zvětší ho a roztáhne, aby v jednom směru zaplnil okno, nakonec doplní výřez ve druhém směru
  + Postupně vykreslí pozadí, osy, náboje a nakonec sondu

Třída poskytuje základní funkce pro správu a vizualizaci nabitých částic (INaboj), přičemž dynamicky upravuje jejich zobrazení na základě jejich vlastností a pozic.

# Grid

Třída Grid implementuje rozhraní IGrid a zajišťuje vykreslení mřížky s osami v grafickém prostředí. Tato třída je navržena tak, aby kreslila mřížku s šipkami na kladných koncích os.

## Metody:

* **Draw(Graphics g, float tipLength, float scale)**
  + Veřejná metoda, která vykreslí mřížku
  + Používá interní metodu DrawGrid pro kreslení samotných os a šipek na konci os.

Třída je zodpovědná za kreslení základní mřížky s osami a šipkami, ale funkce pro práci s počtem čar v mřížce zatím není implementována. Rozměry souřadných os (tloušťka a velikost šipky) si zachovávají pevnou reálnou velikost i při změně velikosti okna.

# Naboj

Třída Naboj implementuje rozhraní INaboj a reprezentuje nabitou částici s určitým nábojem, polohou, poloměrem a unikátním ID. Tato třída také obsahuje metodu pro vykreslení nabité částice na grafickém plátně.

## Metody:

* **float GetCharge()**
  + Vrací aktuální hodnotu náboje částice.
* **string GetChargeStr()**
  + Vrátí řetězec pro výpočet náboje
* **void SetChargeStr(string charge)**
  + Nastaví řetězec pro výpočet náboje částice.
* **string Save()**
  + Vrátí řetězec pro uložení náboje do souboru
* **bool IsHit(PointF point)**
  + Je bod v grafické reprezentaci náboje
* **Drag(Vector2 v)**
  + Posune náboj o daný vektor
* **PointF GetPosition()**
  + Vrací aktuální pozici (střed) částice.
* **void SetPosition(float X, float Y)**
  + Nastaví pozici částice.
* **float GetRadius()**
  + Vrací aktuální hodnotu poloměru částice.
* **void SetRadius(float radius)**
  + Nastaví nový poloměr částice.
* **int GetID()**
  + Vrací unikátní identifikátor částice.
* **void Draw(Graphics g, PointF panelCenter, float scale)**
  + Vykreslí částici na plátno Graphics g.
  + Vytvoří kruh.
  + Zobrazí popisek s hodnotou náboje částice uprostřed kruhu.
  + Pro vykreslení popisku je použito písmo, které se mění podle zadaného měřítka (scale).

Třída Naboj poskytuje základní funkce pro práci s nabitou částicí, jako je získávání a nastavování hodnot náboje, pozice, poloměru a unikátního ID, a také možnost vykreslení částice na grafické plátno.

# Probe

Třída Probe implementuje rozhraní IProbe a slouží k vizualizaci sondy a jejího chování v elektrickém poli. Třída simuluje pohyb částice a vykresluje šipku znázorňující sílu elektrického pole v daném bodě.

## Metody:

* **void Draw(Graphics g, int startTime, INaboj[] charges, float scale)**
  + Vykresluje sondu na grafické plátno g.
  + Vypočítá aktuální polohu sondy podle času (startTime) a rychlosti rotace (anglePerSecond).
  + Simuluje vliv okolních nábojů (charges) na elektrické pole v daném bodě.
  + Vypočítá sílu elektrického pole a vykreslí vektor síly jako šipku.
  + Šipka je doplněna popiskem ukazujícím velikost síly elektrického pole v jednotkách TN/C.
* **float GetRadius()**
  + Vrátí poloměr oběhu
* **SetRadius(float newRadius)**
  + Nastaví poloměr oběhu
* **PointF GetCenter()**
  + Vrátí střed oběhu
* **SetCenter(PointF newCenter)**
  + Nastaví střed oběhu
* **float GetAnglePerSecond()**
  + Vrátí úhlovou rychlost oběhu
* **SetAnglePerSecond(float newAngle)**
  + Nastaví úhlovou rychlost oběhu
* **AddTimeHeld(float t)**
  + Přičte hodnotu t ke stávajícímu času, po který byla sonda držena
* **int GetID()**
  + Vrátí ID sondy
* **bool IsHit(PointF point)**
  + Je bod v grafické reprezentaci sondy
* **Drag(Vector2 v)**
  + Posune sondu o daný vektor
* **string Save()**
  + Vrátí řetězec pro uložení náboje do souboru
* **Tick()**
  + Obsluha nového snímku
* **Calc(Vector2 start, INaboj[] charges)**
  + Vypočte a uloží si vektor intenzity na aktuální poloze
* **Calc(int startTime, INaboj[] charges)**
  + Vypočte a uloží si vektor intenzity na aktuální poloze

## Další informace:

* **Vykreslování textu a sondy**: Při vykreslování sondy se zobrazí také kruh v počátku vektoru síly a text znázorňující sílu elektrického pole v daném bodě.
* **Transformace souřadnic**: Pro vykreslení se využívá transformace souřadnic (g.TranslateTransform), aby se pozice a měřítko přizpůsobily grafickému prostředí.
* **Je použita i jako šipka v mřížce**

Tato třída umožňuje grafickou vizualizaci pohybu sondy a změny elektrického pole způsobené okolními náboji.

# Sestavení a spuštění

Pro sestavení se spustí skript Build.cmd v domovské složce projektu.

Ke spuštění slouží skript Run.cmd spuštěný v domovské složce projektu. Skript bere jeden parametr od 0 do 4, který značí číslo scénáře k vykreslení. Při neuvedení parametru nebo při neplatném parametru se použije 0.

Dále je možné nastavit rozteč mřížky pomocí přepínače -g<X>x<Y>, při neuvedení se bere hodnota 50x50

Případně lze sestavit a spustit pomocí exec.ps1

# Další části vizualizace

* **Ovládací panel**
  + Slouží k ovládání vizualizace
  + Zapínání barevného podkreslení a mřížky a úprava rozteče mřížky
  + Výběr scénáře, uložení a otevření souboru se scénářem
  + Otevírání oken pro úpravy objektů v simulaci a okna s grafy
  + Legenda k barevné mapě
* **Grafy**
  + Záložka 0 – všechny sondy
  + Ostatní záložky – jednotlivé sondy
* **Tabulka nábojů a sond**
  + Slouží k tvorbě, úpravám a mazání nábojů, resp. sond
* **Ukládání do souboru**
  + Textový soubor .upg
  + Formát:
    - naboj:{vzorec pro vypocet naboje};{pozice x};{pozice y}
    - sonda:{stred X};{stred Y};{polomer};{uhlova rychlost}
* **U úprav souborů, nábojů a sond nutno dodržet:**
  + Matematické funkce vždy s velkým písmenem (viz dokumentace knihovny NCalc)
  + Desetinná čísla vždy s tečkou
  + Parametr času vždy jako „[t]“, pokud je úprava vůbec přípustná – pouze u velikosti náboje
  + Při nedodržení není zaručena funkčnost
* **Pohyb**
  + Posun – chytit pravým tlačítkem myši a táhnout, případně pomocí WASD
  + Přiblížení – kolečkem myši nebo klávesy I a O

# Dokončená rozšíření

* Načtení scénáře ze souboru
* Editor dynamických scénářů
* Zvětšení a posun
* Piktogramy nábojů
* Více sond (až 7 – více už nebylo příliš přehledné)