

Puncte uniform distribuite

Didactica Specialității
Niculae Ștefan

Enunt

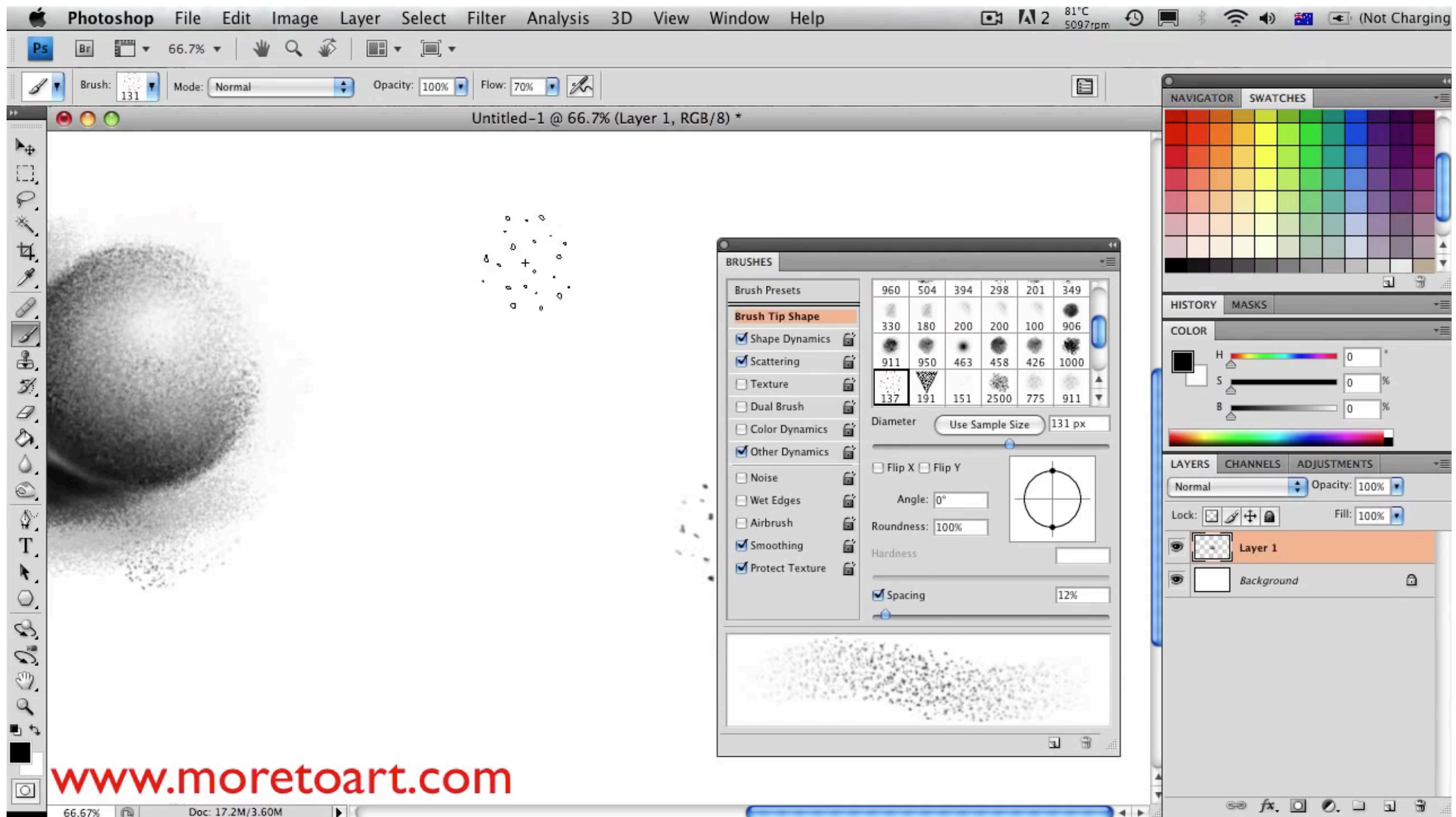
- Generarea unor puncte uniform distribuite
- In interiorul unui poligon
 - dreptunghi
 - regulat
 - neregulat

Aplicatii

- Jocuri video - elemente **mediu/decor**
- Prelucrare foto - **pensule**
- Grafica computerizata - **textura**
- Editare video - efecte cu **particule**



Generare automata de elemente decor
in jocuri video (*ModNation Racers*)



Pensula particularizata in prelucrare foto (*Photoshop*)

Mr. Peabody & Sherman (2014)



Efecte cu particule
in editarea video (*Dreamworks*)



Textura
in grafica computerizzata (*Toy Story 3*)

Motivatie

- Plasare elemente **decor** (copaci) in zona desemnata (padurea)
- Plasare elemente **destructibile** (butoaie explozibile) in zona desemnata (campul de batalie)
- **Regenerarea** dupa distrugere in locuri diferite

Prezentarea algoritmului

- Constructie **incrementala**
 1. unidimensional
 2. dreptunghi
 3. poligon regulat
 4. poligon neregulat
- **Exemplu** practic la final

Unidimensional

- Generarea unui **numar in intervalul** dat
- Consideram functia Random implementata

Unidimensional

```
1  function GenereazaInInterval(a, b)
2      return Random(a, b)
```

Dreptunghi

- Sistem coordonate xOy
- Dreptunghiul este definit de:
 - coltul **stanga sus** A
 - coltul **dreapta jos** B

Dreptunghi

```
1  function GenereazaInDreptunghi(A, B)
2      x = Random(A.x, B.x)
3      y = Random(B.x, B.y)
4      return (x, y)
```

Poligon

- Retinut ca **lista** de n colturi
- **Bounding Box**: maxim si minim pt x si y
- **Orientare**: aria definita de vectorii colturilor

Poligon

```
1  function Orientare(P, Q, R)
2      val = (Q.y-P.y)(R.x-Q.x)-(Q.x-P.x)(R.y-Q.y)
3      return signum(val)
4
5  function Contine(P, X)
6      parte = Orientare(P[0], P[1], P[2])
7      for i = 2, P.n
8          if Orientare (P[i-1], P[i], P[i+1]) != parte
9              return false
10     return true
```

Poligon regulat

- Pastram doar punctele din **interior**
- Algoritmul se **termina** (aleator nu, improbabil)

Poligon regulat

```
1  function GenereazaInPoligonRegulat(P)
2      BB = BoundingBox(P)
3
4      do
5          X = GenereazaInDreptunghi(BB)
6      while P contine X
7
8      return X
```

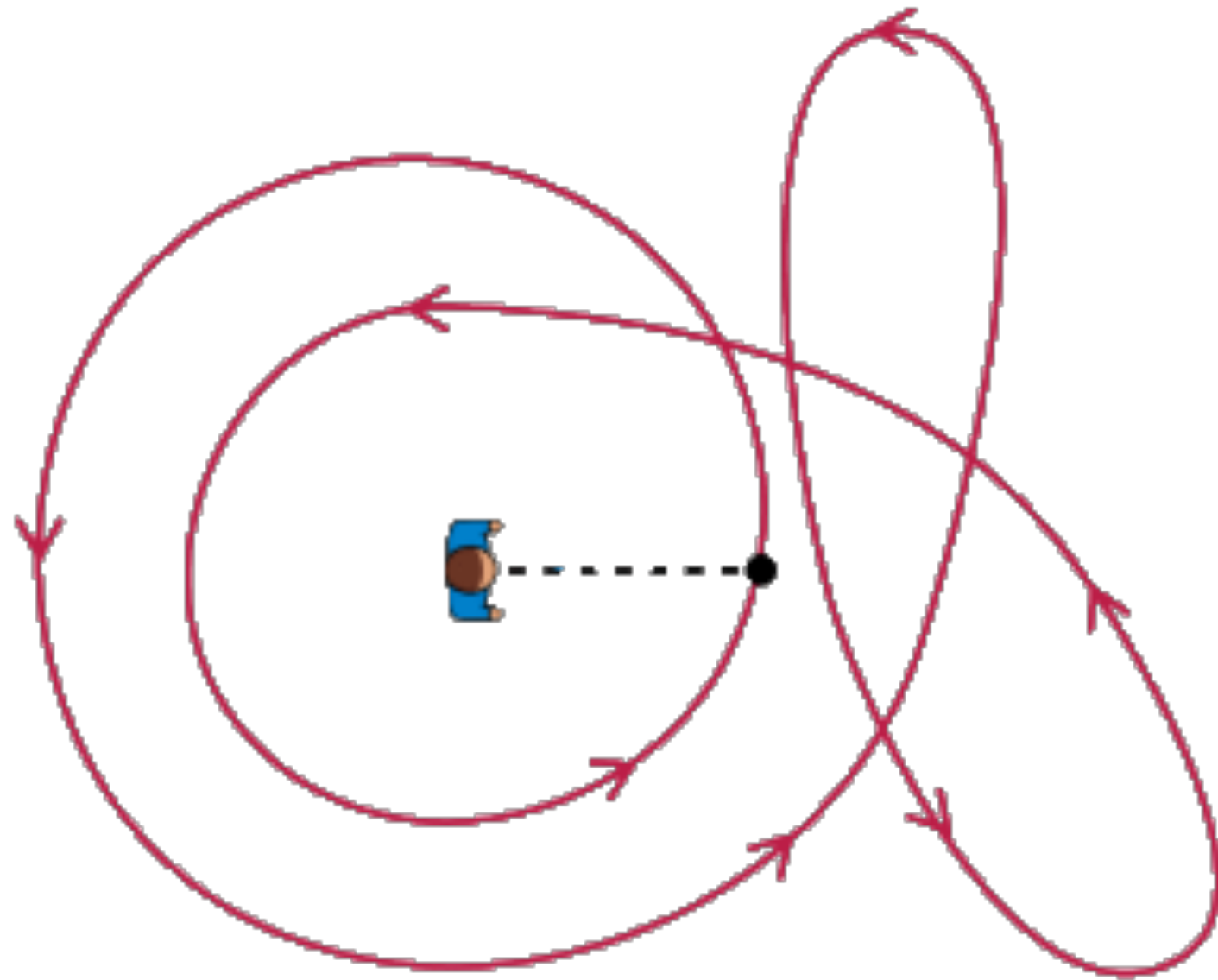
Poligon neregulat

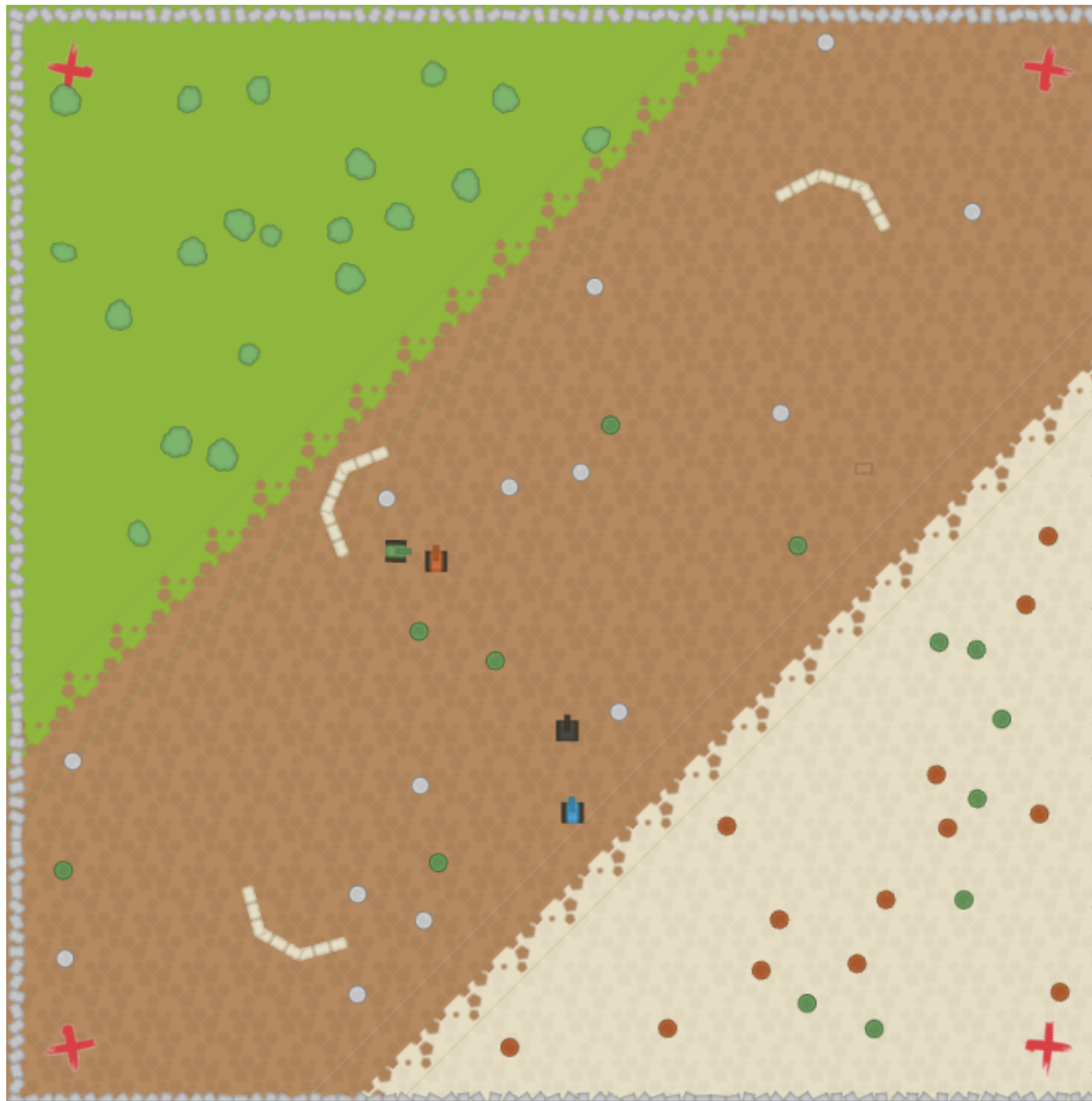
- Test **apartenenta** diferit
- Numarul de **bobinari** in jurul axei

Poligon neregulat

```
1  function Contine(P, X)
2      wn = 0
3      for i = 0, P.n
4          orient = Orientare(P[i], P[i+1], X)
5          if P[i].y <= X.y
6              if P[i+1].y > X.y AND orient > 0
7                  wn++
8          else
9              if P[i+1].y <= X.y AND orient < 0
10                 wn--
11      return wn != 0
```

Numar de bobinari





Copaci si butoaie
Senile de Foc

Intrebari...

Multumesc!

Bibliografie

- youtu.be/t8r8DWL9gj8
- youtu.be/7DGb_ZAqD5Q
- youtu.be/xPpy8mYHQps
- youtu.be/JcpWXaA2qeg
- Computational Geometry Algorithms and Applications
- en.wikipedia.org/wiki/Random_number_generation
- geomalgorithms.com/a03-_inclusion.html