



OSTRAVSKÁ UNIVERZITA
PŘÍRODOVĚDECKÁ FAKULTA

SOFCO

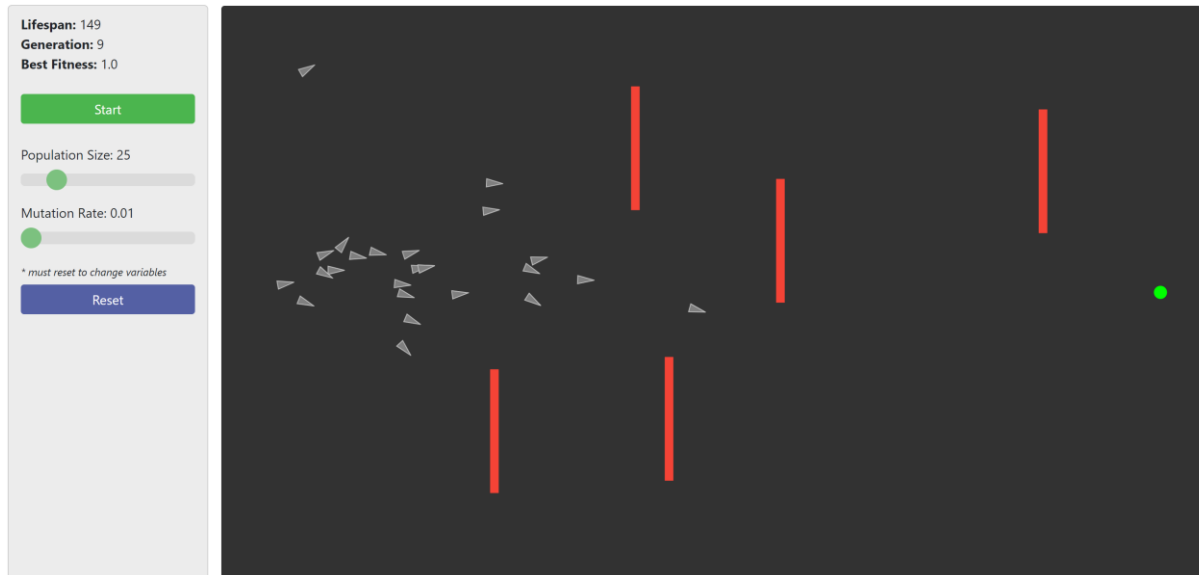
Projekt Genetické Algoritmy

KREJČÍ DAVID, R16439



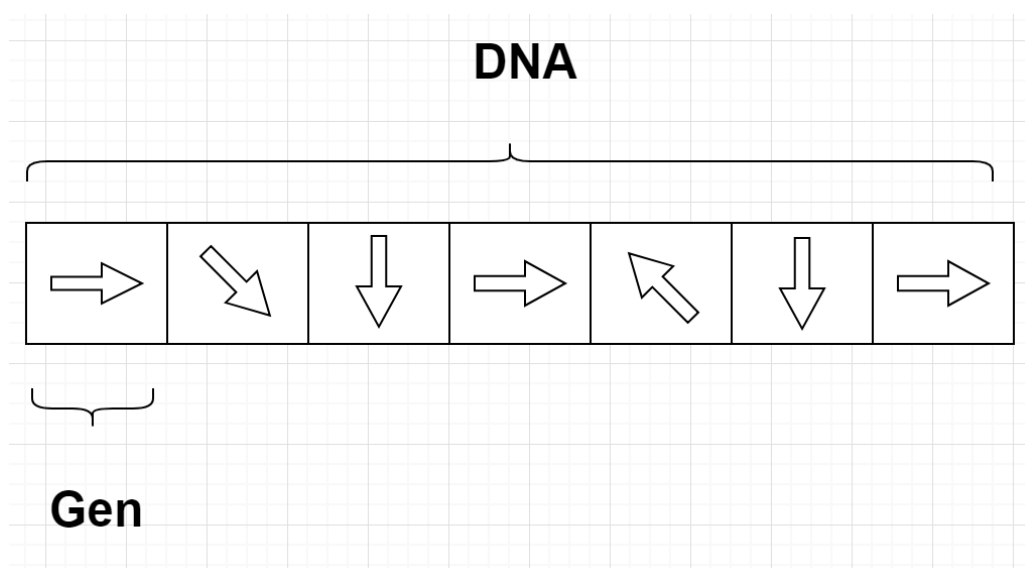
Vymezení problému

Cílem práce je implementování genetického algoritmu, který má za úkol najít cestu rakety do cíle, aniž by raketa narazila do překážky.



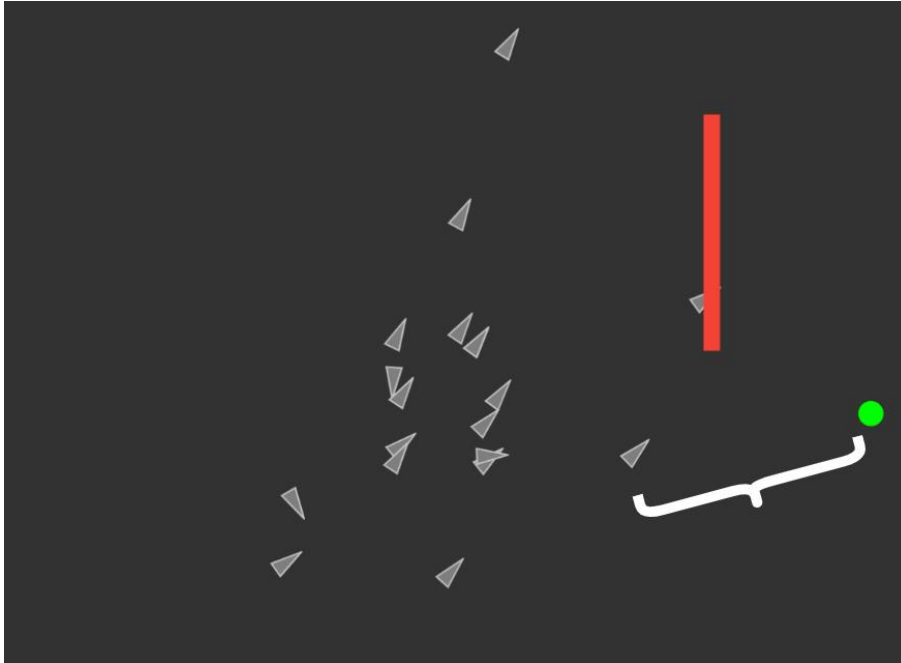
Navržení genetické struktury

Na vstupu máme proměnlivý počet raket, které mají DNA/chromozom vytvořené z vektorů. Tyto vektory určují směr pohybu každé rakety v každém vykresleném snímku aplikace.



Fitness funkce

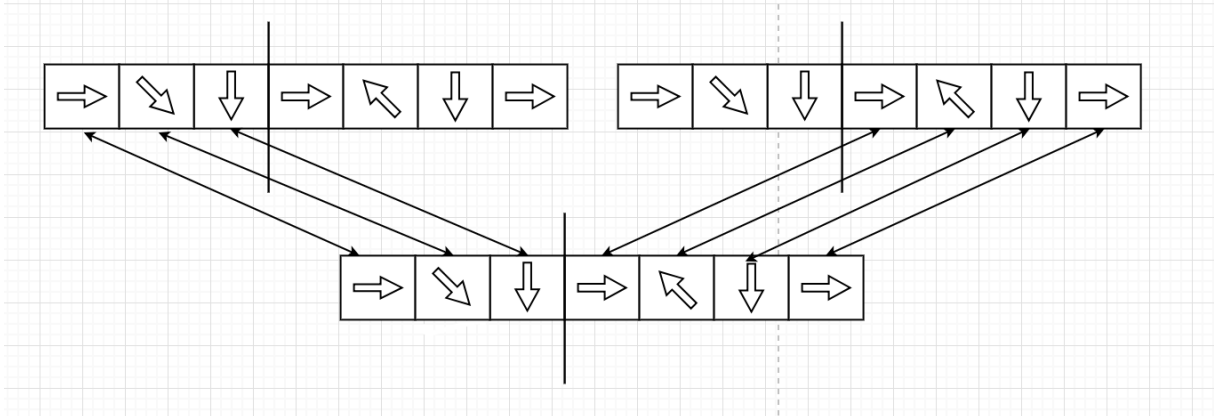
Pro každou z raket se při konci každé generace vypočte fitness na základě vzdálenosti od cíle. Navíc pokud raketa trefí cíl, její fitness se vynásobí desetkrát. Pokud naopak raketa narazí do překážky, její fitness se zmenší o 80%.



```
this.calculateFitness = function(){  
  var distance = dist(this.position.x,this.position.y,target.x,target.y);  
  this.fitness = map(distance, 0, width, width, 0);  
  if (this.completed) {  
    this.fitness *= 10;  
  }  
  if (this.crashed) {  
    this.fitness = this.fitness - (this.fitness*0.8);  
  }  
}
```

Křížení

Křížení probíhá jednobodové. Vybereme náhodný bod v DNA a vytvoříme nové DNA s první částí genů jednoho rodiče a druhou částí druhého rodiče.



```
this.crossover = function(partner){
  var newgenes = [];
  var mid = floor(random(this.genes.length))
  for (let i = 0; i < this.genes.length; i++) {
    if(i > mid){
      newgenes[i] = this.genes[i];
    } else{
      newgenes[i] = partner.genes[i];
    }
  }
  return new DNA(newgenes);
}
```

Mutace

V každém členovi nové populace aplikujeme mutaci s pravděpodobností `mutationRate`. Mutace spočívá ve vygenerování nového náhodného genu (vektoru).

```
this.mutation = function () {
  for (let i = 0; i < this.genes.length; i++) {
    if (random(1) < mutationRate) {
      this.genes[i] = p5.Vector.random2D();
      this.genes[i].setMag(maxForce);
    }
  }
}
```

Selekce

Vytvoříme nové pole `matingPool`, kde přidáváme jednotlivé rakety podle fitness. Čím větší fitness, tím vícekrát je přidáme.

```
this.matingPool = [];  
for (let i = 0; i < this.populationSize; i++) {  
  var n = this.population[i].fitness * 100;  
  for (let j = 0; j < n; j++) {  
    this.matingPool.push(this.population[i])  
  }  
}
```

Poté z tohoto „pářícího pole“ náhodně vybíráme jedince pro křížení. Jedinci s nejvyšší fitness a tudíž nejpočetněji zastoupení, mají největší pravděpodobnost výběru pro křížení.

Je zaveden elitismus. Raketa s nejlepší fitness je předána do další generace.

```
this.selection = function () {  
  var newPopulation = [];  
  
  // Elitism - pass best rocket from previous generation to new generation  
  var bestDNA = this.bestDNA();  
  bestDNA.mutation();  
  newPopulation.push(new Rocket(bestDNA));  
  
  for (let i = 0; i < this.population.length - 1; i++) {  
    var parentA = random(this.matingPool).dna;  
    var parentB = random(this.matingPool).dna;  
    var child = parentA.crossover(parentB);  
    child.mutation();  
    newPopulation.push(new Rocket(child));  
  }  
  this.population = newPopulation;  
}
```