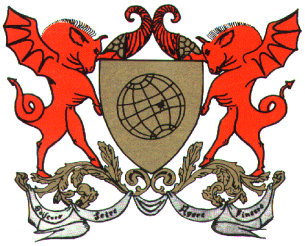


Universidade Federal de Viçosa

Campus Florestal

# TRABALHO PRÁTICO 03 - ROBOCODE

# PROGRAMAÇÃO ORIENTADA A OBJETOS – CCF 313

Samuel Jhonata S. Tavares – 2282

Professor Fabrício Aguiar Silva

Florestal, 2016

# 1. INTRODUÇÃO

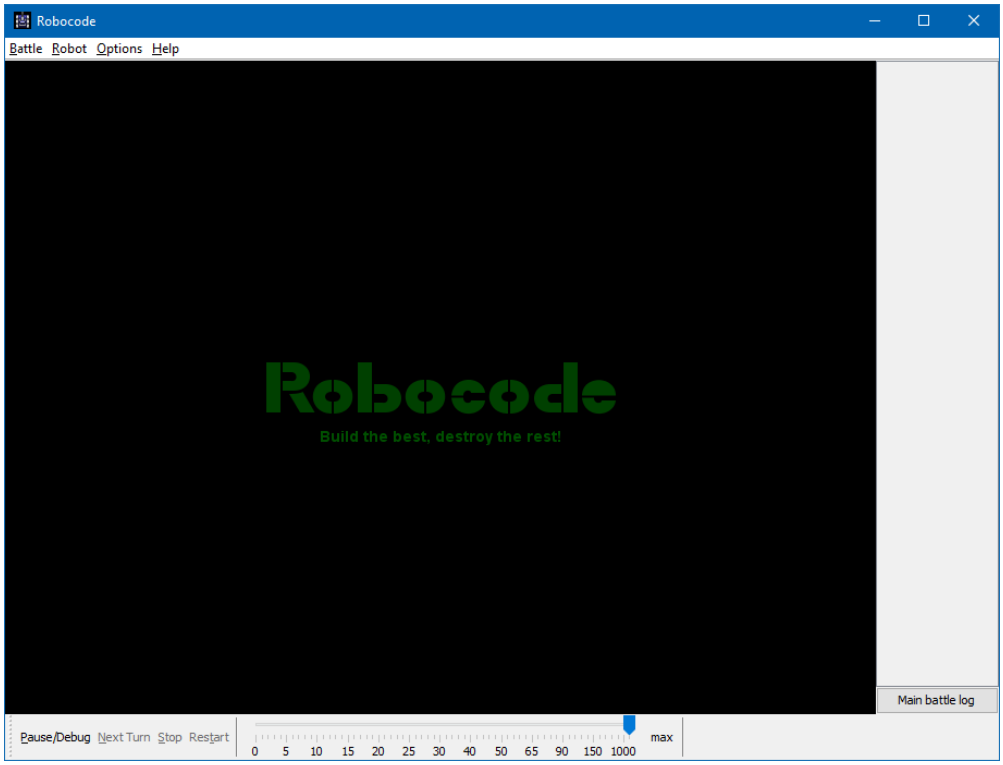
Este trabalho tem por finalidade a implementação de uma classe de um robô, herdade da superclasse Robot da API Robocode, disponibilizada em http://robocode.sourceforge.net .

Foi necessário fazer um estudo da API para descobrir suas funcionalidades já implementadas na classe Robot, para ser criado um novo modelo robô que foi nomeado SJRobot, que será apresentado a seguir.

# 2. DESENVOLVIMENTO

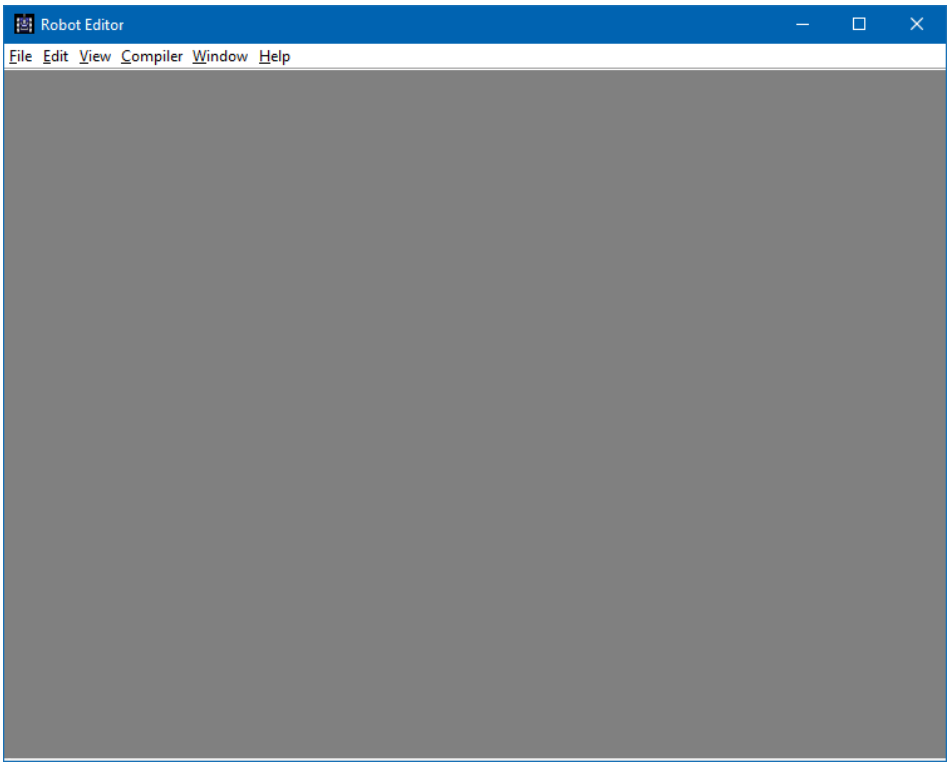
## 2.1 – Informações gerais sobre a API Robocode

A API Robocode e seu software foram baixados e instalados, sendo possível se criar e editar robôs e posteriormente criar batalhas, com os robôs desenvolvidos e/ou com os já implementados.




*Tela Principal do Robocode*

O software é de fácil uso, e para criar um novo robô ou editar um já existente, é necessário abrir o RoboEditor, como mostra na figura a seguir:




*Editor de Robôs*

A classe de qualquer robô deve estender uma das três superclasses existentes na API: JuniorRobot, Robot ou AdvancedRobot. A primeira tem seus métodos restritos, havendo apenas as funcionalidades mais simples do robô. A segunda apresenta mais possibilidades, porém ainda um pouco limitado. Já a terceira apresenta métodos mais complexos que podem ajudar a se ter um melhor desempenho ao robô implementado.

Os principais métodos, da classe AdvancedRobot, a serem implementados:

**run():** responsável pela execução do robô, é nela que ficam os métodos que serão executados a todo tempo, enquanto o robô ainda estiver na batalha.

**onScannedRobot(ScannedRobotEvent e)**: esta é ativada toda vez que o radar detecta um adversário.

**onHitByBullet(HitByBulletEvent**: ativada quando o robô é atingido por um tiro.

**onHitWall(HitWallEvent e)**: ativada quando o robô bate em uma parede.

**onHitRobot()**: ativada quando o robô bate em um adversário.

**onHitByBullet()**: ativada quando o robô é atingido por uma bala.

**onBulletHit():** ativada quando a bala disparada atinge um inimigo.

**onBulletMissed():** ativada quando a bala disparada atinge a parede.

**onBulletHitBullet():** ativada quando a bala disparada atinge outra bala.

Há diversos métodos que devem ser utilizados para fazer o robô andar pra frente [**setAhead()**], andar para trás [**setBack()**], girar para a direita [**setTurnRight()**] e para a esquerda [**setTurnLeft()**], fazer o canhão girar para a direita [**setTurnGunRight()**], ou para a esquerda [**setTurnGunRight()**], atirar [**fire()**], fazer movimentos em simultâneo [**execute()**], etc.

### 2.2 – Definição do Robô

Já com os conhecimentos fundamentais da API, agora é hora de se definir uma estratégia para o robô criado, antes de sua implementação.

O robô deverá fazer movimentos circulares (em sentido horário), que permitirão desviar das balas mais facilmente. Enquanto o robô gira em sentido horário, o seu radar gira em sentido oposto, para detectar o inimigo.

Quando o inimigo é encontrado, deve ser disparado um tiro, dependendo da distância que o inimigo se encontra, será definido a sua intensidade.

Caso acerte o tiro, então o canhão deverá ficar parado, pois há um alvo na sua frente que está sendo atingido.

Se o robô bater na parede, é preciso que ele consiga sair, pois ficar encurralado poderá custar a sua vida.

Quando o robô é atingido por um tiro, ele deverá retornar um pouco para trás, para tentar sair da visão do radar do inimigo.

Se ele bater em um inimigo, deverá se afastar do mesmo e depois continuar sua rota.

### 2.3 – Implementação do Robô

Já com o robô definido, foi implementado o seguinte código:

O método que é executado sempre, enquanto o robô não sofrer alguma intervenção:

```
public class SJRobo extends AdvancedRobot {

    /**
     * run: SJRobo's default behavior
     */
    public void run() {
        setBodyColor(Color.BLUE); //cor do corpo
        setGunColor(Color.white); //cor da arma
        setRadarColor(Color.white); //cor do radar
        setBulletColor(Color.blue); //cor da bala

        // Robot main loop
        while (true) {
            setAhead(40); //andar pra frente
            setTurnRight(180); //girar pra direita
            setBack(50); //andar pra trás
            execute(); //executar conjuntamente
            setBack(10); //andar pra trás
            for (int i = 0; i < 2; i++) {
                setAhead(0); //andar pra frente
                setTurnRight(45); //girar pra direita
                setTurnGunRight(80); //girar o canhão para a direita
            }
            for (int i = 0; i < 4; i++) {
                setBack(60); //andar pra tras
                setTurnRight(40); //girar pra direita
                setTurnGunLeft(30); //girar o canhão para a esquerda
                execute();//executar conjuntamente
            }

        }
    }
```

*Método run()*

Método disparado quando detecta um adversário no radar:

```
public void onScannedRobot(ScannedRobotEvent e) { //detecta um adversário no radar

    if(e.getDistance()<15){
        fire(3); //atirar
    }else if(e.getDistance() < 30){
        fire(2.5); //atirar
    }else if(e.getDistance() < 50){
        fire(2); //atirar
    }else{
        fire(1.5); //atirar
    }
}
```

*Método onScannedRobot()*

Método disparado quando o robô é atingido por um tiro:

```
public void onHitByBullet(HitByBulletEvent e) { //atingido por um tiro
    setBack(40); //andar pra trás
}
```

*Método onHitByBullet()*

Método disparado quando o robô bate na parede:

```
public void onHitWall(HitWallEvent e) { //bate em uma parede
    setTurnLeft(180 - Math.abs(e.getBearing())); //girar para a esquerda
    setAhead(80); //andar pra frente
    execute();//executar conjuntamente
}
```

*Método onHitWall*

Método disparado quando o robô bate em um adversário:

```
public void onHitRobot(){ //bate em um adversário
    setTurnLeft(90); //girar para a esquerda
    setAhead(150); //andar pra frente
    fire(3); //atirar
    execute();//executar conjuntamente
}
```

*Método onHitRobot*

Método disparado quando o robô é atingido por uma bala:

```
public void onHitByBullet(){ //atingido por uma bala
    setTurnRight(10); //girar pra direita
    setAhead(50); //andar pra frente
    execute();//executar conjuntamente
}
```

*Método onHitByBullet()*

Método disparado quando a bala disparada atinge o inimigo:

```
public void onBulletHit(){//bala disparada atingiu inimigo
    setAhead(30); //andar pra frente
    setTurnLeft(0); //girar para a esquerda
    execute();//executar conjuntamente
}
```

*Método onBulletHit()*

## 2.4 – Realizando batalhas de teste

Com o código pronto, foram executadas três batalhas de teste com 18 oponentes:


*Batalha realizada com 19 robôs*

Primeiro resultado:

Results for 10 rounds

| Rank | Robot Name | Total Score | Survival | Surv Bonus | Bullet Dmg | Bullet Bonus | Ram Dmg * 2 | Ram Bonus | 1sts | 2nds | 3rds |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1st | sample.SpinBot | 9740 (9%) | 6600 | 180 | 2617 | 154 | 190 | 0 | 1 | 1 | 1 |
| 2nd | sample.Tracker | 9558 (8%) | 6250 | 360 | 2527 | 311 | 110 | 0 | 2 | 3 | 0 |
| 3rd | SJ.SJRobo* | 9357 (8%) | 6350 | 540 | 2076 | 107 | 283 | 0 | 3 | 0 | 1 |
| 4th | sample.Walls | 9096 (8%) | 5950 | 720 | 2147 | 208 | 65 | 6 | 4 | 0 | 1 |
| 5th | sample.TrackFire | 7160 (6%) | 4350 | 0 | 2579 | 230 | 0 | 0 | 0 | 0 | 0 |
| 6th | sample.Fire | 6960 (6%) | 5650 | 0 | 1260 | 39 | 11 | 0 | 0 | 1 | 2 |
| 7th | sample.Crazy | 6796 (6%) | 5750 | 0 | 898 | 6 | 143 | 0 | 0 | 2 | 1 |
| 8th | sample.RamFire | 6776 (6%) | 3850 | 0 | 1964 | 100 | 631 | 232 | 0 | 1 | 0 |
| 9th | sample.Corners | 6421 (6%) | 4900 | 0 | 1476 | 40 | 5 | 0 | 0 | 0 | 1 |
| 10th | sample.MyFirstJuniorRobot | 5592 (5%) | 4750 | 0 | 755 | 34 | 53 | 0 | 0 | 0 | 1 |
| 11th | sample.VelociRobot | 5502 (5%) | 3900 | 0 | 882 | 17 | 679 | 24 | 0 | 1 | 0 |
| 12th | sample.PaintingRobot | 4984 (4%) | 4350 | 0 | 598 | 8 | 28 | 0 | 0 | 1 | 0 |
| 13th | sampleex.MasterAndSlave | 4905 (4%) | 4350 | 0 | 525 | 3 | 28 | 0 | 0 | 0 | 0 |
| 14th | sampleex.ProxyOfGreyEmine... | 4241 (4%) | 3700 | 0 | 512 | 4 | 25 | 0 | 0 | 0 | 0 |
| 15th | sampleex.Alien | 3619 (3%) | 3100 | 0 | 487 | 3 | 29 | 0 | 0 | 0 | 1 |
| 16th | sample.Target | 3318 (3%) | 3300 | 0 | 0 | 0 | 18 | 0 | 0 | 0 | 1 |
| 17th | sample.MyFirstRobot | 3310 (3%) | 2850 | 0 | 425 | 1 | 34 | 0 | 0 | 0 | 0 |
| 18th | sampleex.AlienComposition | 3303 (3%) | 2850 | 0 | 428 | 0 | 25 | 0 | 0 | 0 | 0 |
| 19th | sample.SittingDuck | 2500 (2%) | 2500 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Save OK

*Resultados de teste 01*

Segundo Resultado:

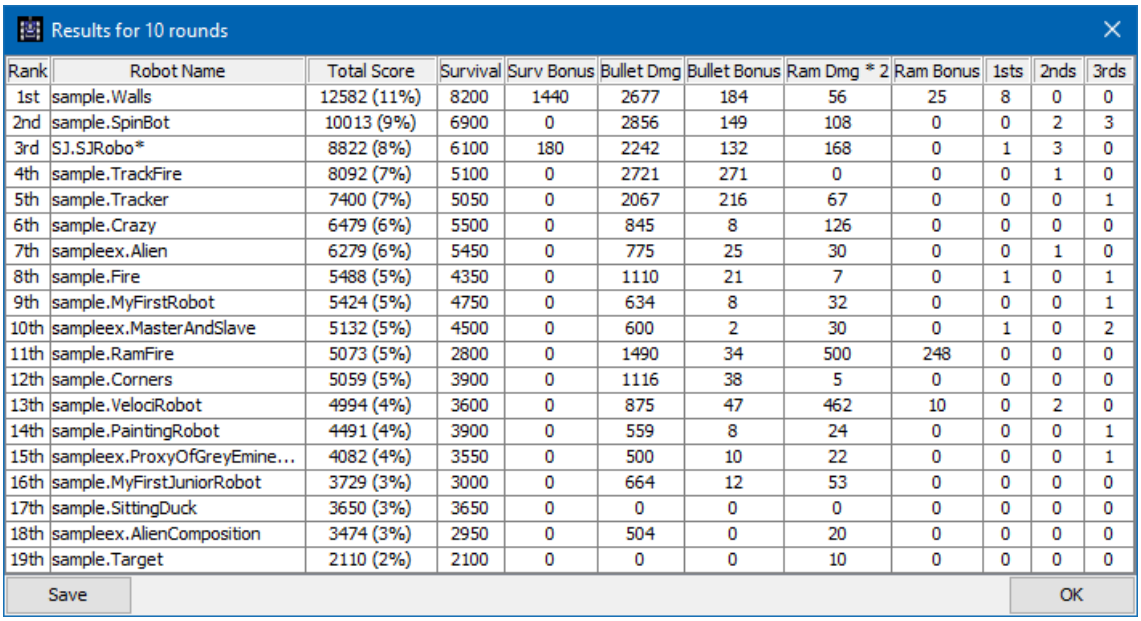
Results for 10 rounds

| Rank | Robot Name | Total Score | Survival | Surv Bonus | Bullet Dmg | Bullet Bonus | Ram Dmg * 2 | Ram Bonus | 1sts | 2nds | 3rds |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1st | sample.Walls | 13045 (12%) | 8300 | 1620 | 2765 | 283 | 77 | 0 | 9 | 0 | 0 |
| 2nd | sample.SpinBot | 9557 (8%) | 6550 | 0 | 2680 | 172 | 155 | 0 | 0 | 3 | 2 |
| 3rd | SJ.SJRobo* | 9282 (8%) | 6800 | 0 | 2161 | 88 | 218 | 14 | 0 | 2 | 2 |
| 4th | sample.Tracker | 8544 (8%) | 5750 | 0 | 2356 | 226 | 187 | 25 | 0 | 3 | 1 |
| 5th | sample.RamFire | 7379 (7%) | 4600 | 0 | 1904 | 54 | 604 | 217 | 0 | 1 | 1 |
| 6th | sample.Crazy | 6952 (6%) | 5800 | 180 | 832 | 6 | 133 | 0 | 1 | 0 | 0 |
| 7th | sampleex.AlienComposition | 5958 (5%) | 5150 | 0 | 755 | 8 | 46 | 0 | 0 | 0 | 0 |
| 8th | sample.TrackFire | 5829 (5%) | 3600 | 0 | 1998 | 230 | 0 | 0 | 0 | 1 | 0 |
| 9th | sample.Fire | 5661 (5%) | 4500 | 0 | 1116 | 31 | 14 | 0 | 0 | 0 | 1 |
| 10th | sample.Target | 4714 (4%) | 4700 | 0 | 0 | 0 | 14 | 0 | 0 | 0 | 0 |
| 11th | sampleex.ProxyOfGreyEmine... | 4663 (4%) | 4050 | 0 | 576 | 8 | 29 | 0 | 0 | 0 | 0 |
| 12th | sampleex.Alien | 4396 (4%) | 3750 | 0 | 599 | 3 | 43 | 0 | 0 | 0 | 0 |
| 13th | sample.VelociRobot | 4337 (4%) | 2950 | 0 | 760 | 12 | 589 | 27 | 0 | 0 | 1 |
| 14th | sample.Corners | 4254 (4%) | 2900 | 0 | 1277 | 70 | 6 | 0 | 0 | 0 | 0 |
| 15th | sample.MyFirstRobot | 4197 (4%) | 3650 | 0 | 515 | 4 | 28 | 0 | 0 | 0 | 0 |
| 16th | sampleex.MasterAndSlave | 4117 (4%) | 3650 | 0 | 432 | 2 | 34 | 0 | 0 | 0 | 0 |
| 17th | sample.MyFirstJuniorRobot | 3933 (3%) | 3250 | 0 | 590 | 16 | 67 | 9 | 0 | 0 | 1 |
| 18th | sample.PaintingRobot | 3802 (3%) | 3250 | 0 | 509 | 2 | 41 | 0 | 0 | 0 | 1 |
| 19th | sample.SittingDuck | 2300 (2%) | 2300 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Save OK

*Resultados de teste 02*

Terceiro resultado:


Results for 10 rounds

| Rank | Robot Name | Total Score | Survival | Surv Bonus | Bullet Dmg | Bullet Bonus | Ram Dmg * 2 | Ram Bonus | 1sts | 2nds | 3rds |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1st | sample.Walls | 12582 (11%) | 8200 | 1440 | 2677 | 184 | 56 | 25 | 8 | 0 | 0 |
| 2nd | sample.SpinBot | 10013 (9%) | 6900 | 0 | 2856 | 149 | 108 | 0 | 0 | 2 | 3 |
| 3rd | SJ.SJRobo* | 8822 (8%) | 6100 | 180 | 2242 | 132 | 168 | 0 | 1 | 3 | 0 |
| 4th | sample.TrackFire | 8092 (7%) | 5100 | 0 | 2721 | 271 | 0 | 0 | 0 | 1 | 0 |
| 5th | sample.Tracker | 7400 (7%) | 5050 | 0 | 2067 | 216 | 67 | 0 | 0 | 0 | 1 |
| 6th | sample.Crazy | 6479 (6%) | 5500 | 0 | 845 | 8 | 126 | 0 | 0 | 0 | 0 |
| 7th | sampleex.Alien | 6279 (6%) | 5450 | 0 | 775 | 25 | 30 | 0 | 0 | 1 | 0 |
| 8th | sample.Fire | 5488 (5%) | 4350 | 0 | 1110 | 21 | 7 | 0 | 1 | 0 | 1 |
| 9th | sample.MyFirstRobot | 5424 (5%) | 4750 | 0 | 634 | 8 | 32 | 0 | 0 | 0 | 1 |
| 10th | sampleex.MasterAndSlave | 5132 (5%) | 4500 | 0 | 600 | 2 | 30 | 0 | 1 | 0 | 2 |
| 11th | sample.RamFire | 5073 (5%) | 2800 | 0 | 1490 | 34 | 500 | 248 | 0 | 0 | 0 |
| 12th | sample.Corners | 5059 (5%) | 3900 | 0 | 1116 | 38 | 5 | 0 | 0 | 0 | 0 |
| 13th | sample.VelociRobot | 4994 (4%) | 3600 | 0 | 875 | 47 | 462 | 10 | 0 | 2 | 0 |
| 14th | sample.PaintingRobot | 4491 (4%) | 3900 | 0 | 559 | 8 | 24 | 0 | 0 | 0 | 1 |
| 15th | sampleex.ProxyOfGreyEmine... | 4082 (4%) | 3550 | 0 | 500 | 10 | 22 | 0 | 0 | 0 | 1 |
| 16th | sample.MyFirstJuniorRobot | 3729 (3%) | 3000 | 0 | 664 | 12 | 53 | 0 | 0 | 0 | 0 |
| 17th | sample.SittingDuck | 3650 (3%) | 3650 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 18th | sampleex.AlienComposition | 3474 (3%) | 2950 | 0 | 504 | 0 | 20 | 0 | 0 | 0 | 0 |
| 19th | sample.Target | 2110 (2%) | 2100 | 0 | 0 | 0 | 10 | 0 | 0 | 0 | 0 |

Save OK

*Resultado de teste 03*

Nas três execuções, o SJRobot ficou em terceiro colocado, com 8% dos pontos de cada batalha, sendo assim um resultado satisfatório.

## 3. CONCLUSÃO

Portanto, aqui foi apresentado o funcionamento fundamental da API Robocode, com os principais métodos a serem implementados, para se ter uma melhor visão de como fazer a implementação de uma nova classe de um robô.

Também foi apresentada a implementação do SJRobot, detalhadamente e realizadas algumas batalhas de testes, tendo resultados satisfatórios.