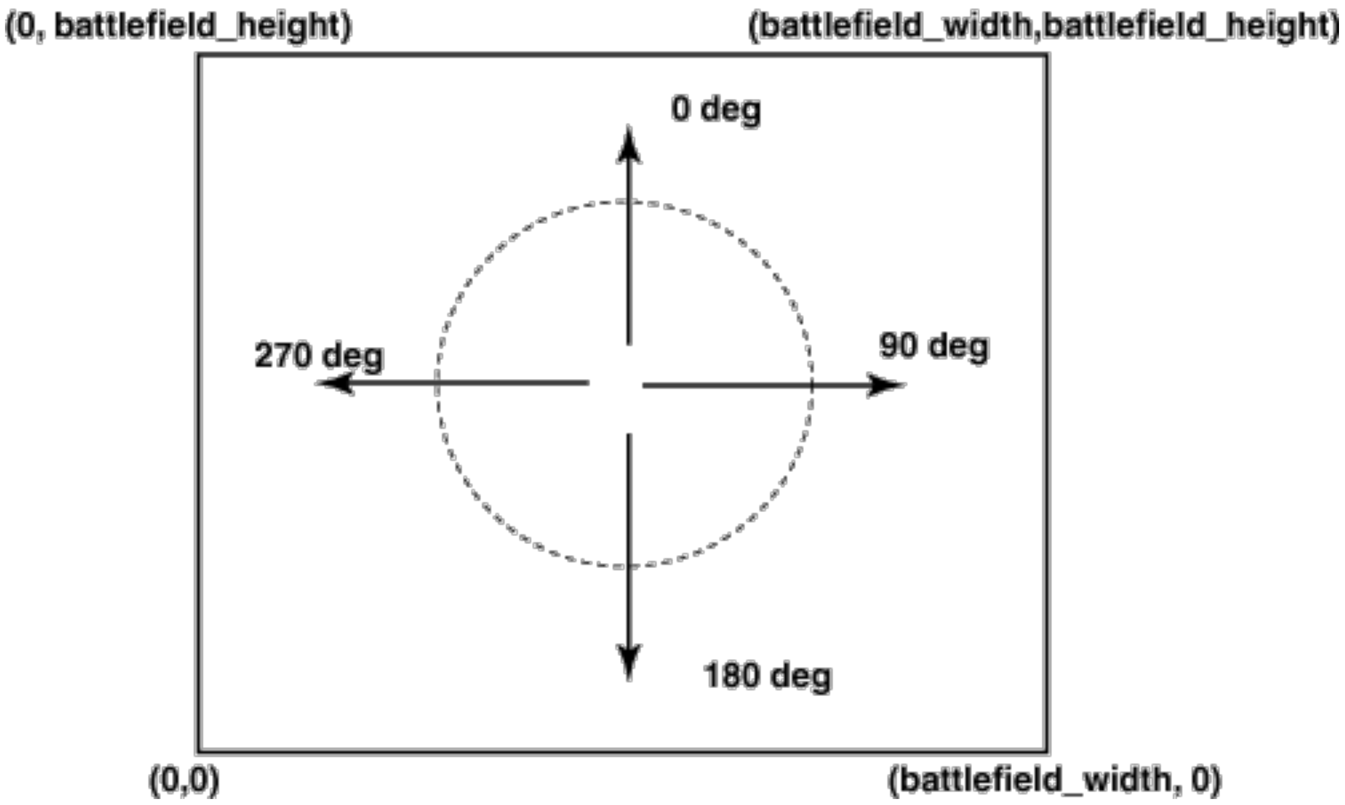
METRO

HACKATHON

FALL 2016 - ROBOCODE COMPETITION

Coordination and Directions

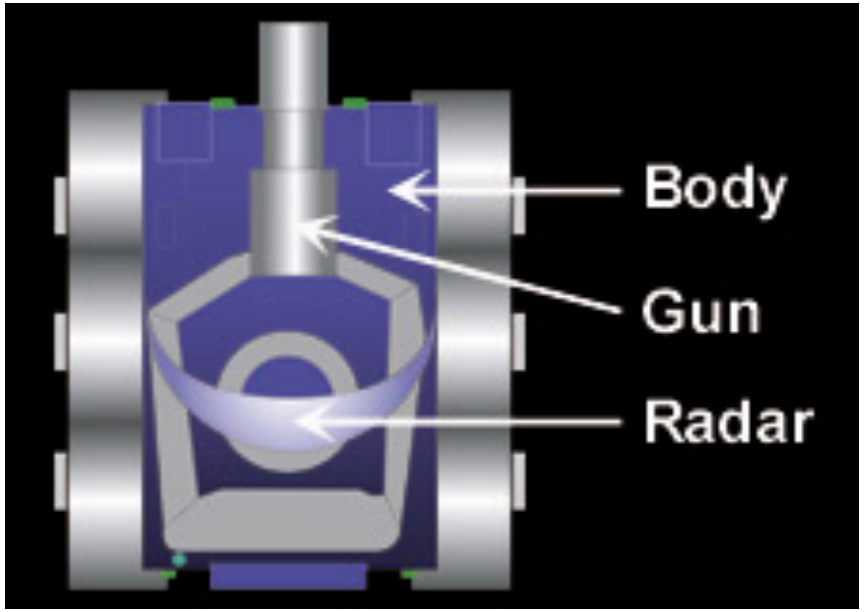


- Cartesian Coordinate System
- Clockwise Direction Convention from 0 (North) and 180 (South)

Robot Movement Physics

- Robocode time is measured in "ticks". Each robot gets one turn per tick. 1 tick = 1 turn.
- Distance is measured in pixels, even when you can move by a fraction.
- Velocity has a maximum of 8 pixels / turn.
- Acceleration is at a rate of 1 pixel / turn / turn. Deceleration is at a rate of 2 pixels / turn / turn.
- Distance Equation: distance = velocity * time

Robot, Gun and Radar Rotation



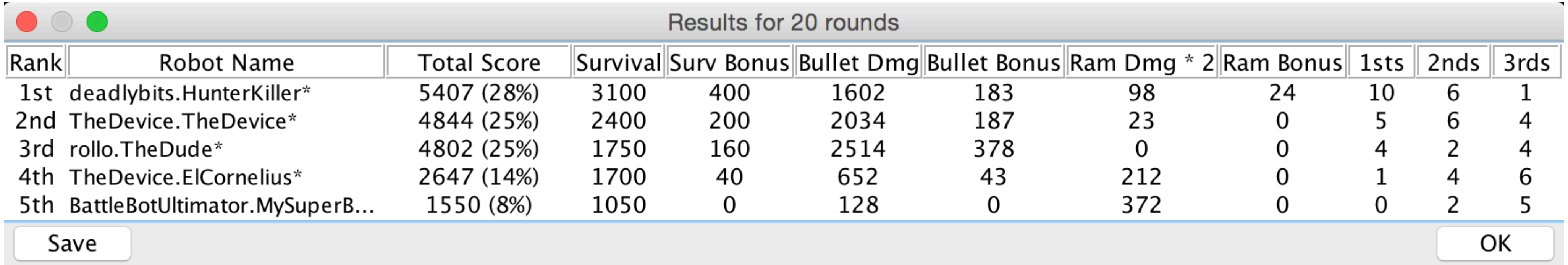
- Robot, Gun and Radar rotate independently.
- Max rate of rotation of robot: (10 - 0.75 * abs(velocity)) deg / turn. The faster you're moving, the slower you turn.
- Max rate of rotation of gun: 20 deg / turn. This is added to the current rate of rotation of the robot.
- Max rate of rotation of radar: 45 deg / turn. This is added to the current rate of rotation of the gun.

Bullets and Collision

- Damage: 4 * firepower. If firepower > 1, it does an additional damage = 2 * (power - 1).
- Velocity: 20 - 3 * firepower
- Gun Heat generated: 1 + firepower / 5. You cannot fire if gunHeat > 0. All guns are hot at the start of each round.
- Power returned on hit: 3 * firepower.
- Collision of robots: Each robot takes 0.6 damage. If a robot is moving away from the collision, it will not be stopped.

Battle and Scoring

- Survival Score – Each robot that's alive scores 50 points every time another robot dies
- Last Survivor Bonus – The last robot alive scores 10 additional points for each robot that died
- Bullet Damage – Robots score 1 point for each point of damage they do to enemies
- Total Score – All above ones added Up
- Bullet Damage Bonus – When a robot kills an enemy, it scores an additional 20% of all the damage it did to that enemy.
- Ram Damage – Robots score 2 points for each point of damage they cause by ramming enemies
- Ram Damage Bonus – When a robot kills an enemy by ramming, it scores an additional 30% of all the damage it did to that enemy.



Results for 20 rounds

Rank	Robot Name	Total Score	Survival	Surv Bonus	Bullet Dmg	Bullet Bonus	Ram Dmg * 2	Ram Bonus	1sts	2nds	3rds
1st	deadlybits.HunterKiller*	5407 (28%)	3100	400	1602	183	98	24	10	6	1
2nd	TheDevice.TheDevice*	4844 (25%)	2400	200	2034	187	23	0	5	6	4
3rd	rollo.TheDude*	4802 (25%)	1750	160	2514	378	0	0	4	2	4
4th	TheDevice.ElCornelius*	2647 (14%)	1700	40	652	43	212	0	1	4	6
5th	BattleBotUltimator.MySuperB...	1550 (8%)	1050	0	128	0	372	0	0	2	5

Battlefield and Competition

- Battlefield size: 1200 x 1200 pixel
- Gun cooling rate: 0.1
- Rounds 20 (Test Battles 10 Rounds)

- Dates:
- Test Battles at 13:00 and 15:00
- Final Commits at 17:00 and Final Battle at 17:30

DEMO: RoboCode

- Execute robocode-setup.jar
- Default Homedirectory/robocode
- Depending on OS: Execute ./robocode.command
- Battle -> New -> Choose Robots -> Add
- Options -> Preferences -> Development Options -> Add Folder /target/classes

Team Repository

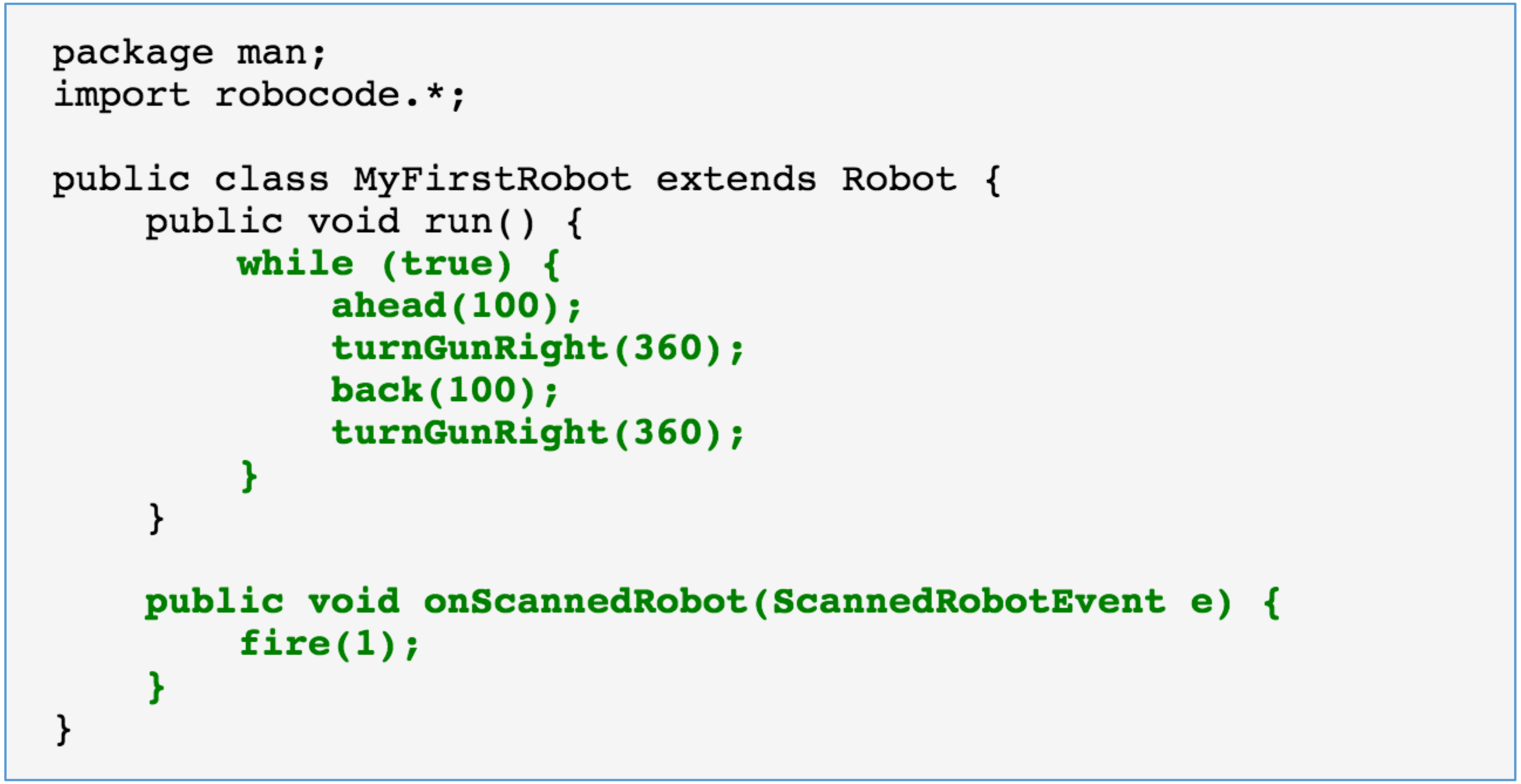
- https://github.com/niklasdelissen/robocode + teamnumber
- git clone https://github.com/niklasdelissen/robocode .git
- push frequently and watch build monitor if anything is broken

Add Dependency to local repository

- On commandline / shell execute the following in the checked out project libs folder :
- mvn install:install-file -Dfile=robocode.jar -DgroupId=net.sf.robocode -DartifactId=robocode.api -Dversion=1.9.2.5 -Dpackaging=JAR

```
<dependency>
   <groupId>net.sf.robocode</groupId>
   <artifactId>robocode.api</artifactId>
   <version>1.9.2.5</version>
</dependency>
```

MyFirstRobot.java



```java
package man;
import robocode.*;

public class MyFirstRobot extends Robot {
    public void run() {
        while (true) {
            ahead(100);
            turnGunRight(360);
            back(100);
            turnGunRight(360);
        }
    }

    public void onScannedRobot(ScannedRobotEvent e) {
        fire(1);
    }
}
```

Ready to code?