Software Engineering Principles for Complex Systems - Project Workshop -

Ricardo Caldas, Mazen Mohamad, Wardah Mahmood,
Malik Hannan, Kevin Ayad

CHALMERS | UNIVERSITY OF GOTHENBURG

Agenda (105’)

- Robocode
  - *Robocode Introduction and Demo (20’)*
  - *Hands on! (DIY) (20’~30’)*
- FeatureIDE
  - *Install FeatureIDE (DIY) (20’~30’)*
  - *Guided example in Antenna (20’)*
- Gitlab
  - *Try to access your gitlab repo (DIY) (5’)*

Join Mentimeter!

Web: www.mentimeter.com/

Code: 7740 1922

OR

www.menti.com/ale9u173d5qs

OR scan me!

Robocode Introduction

Tobias Schwarz, Mazen Mohamad, Thorsten Berger, Wardah Mahmood

CHALMERS | UNIVERSITY OF GOTHENBURG

What is Robocode?

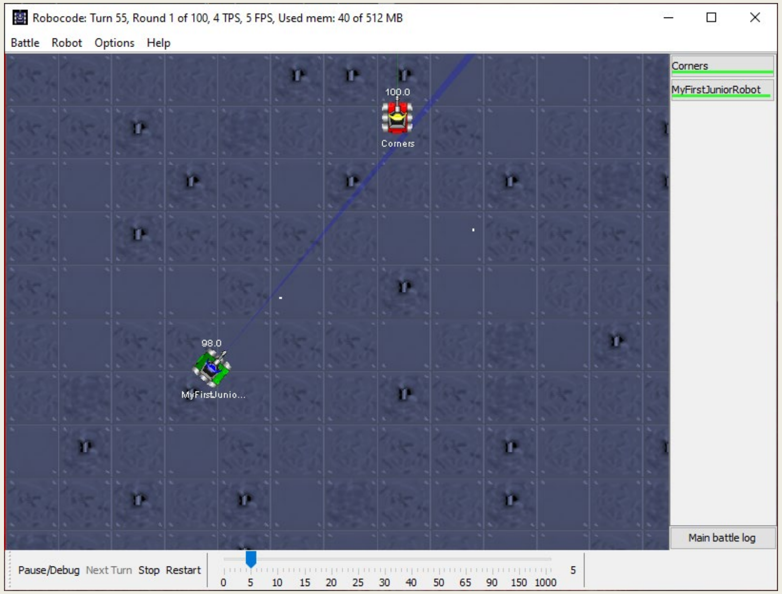
- Robocode is a programming game
  - *Provides a game engine to simulate robot competitions (battles)*
- You’re not directly controlling the robot, but programming it to manage the battle by its own
- Implement one or many Java classes how the robot behaves and reacts to its environment
- Robot competitions takes places

  on a simulated battlefield

- **Purpose of assignment:**

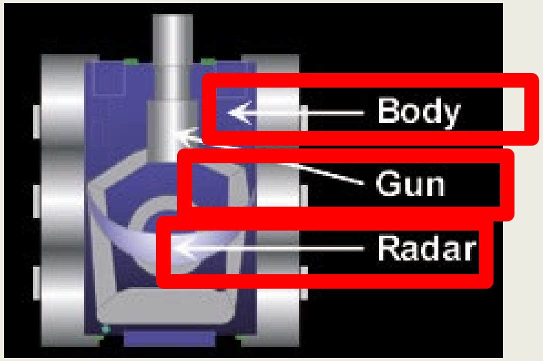
  Learn creating an SPL in a fun

  and interactive environment

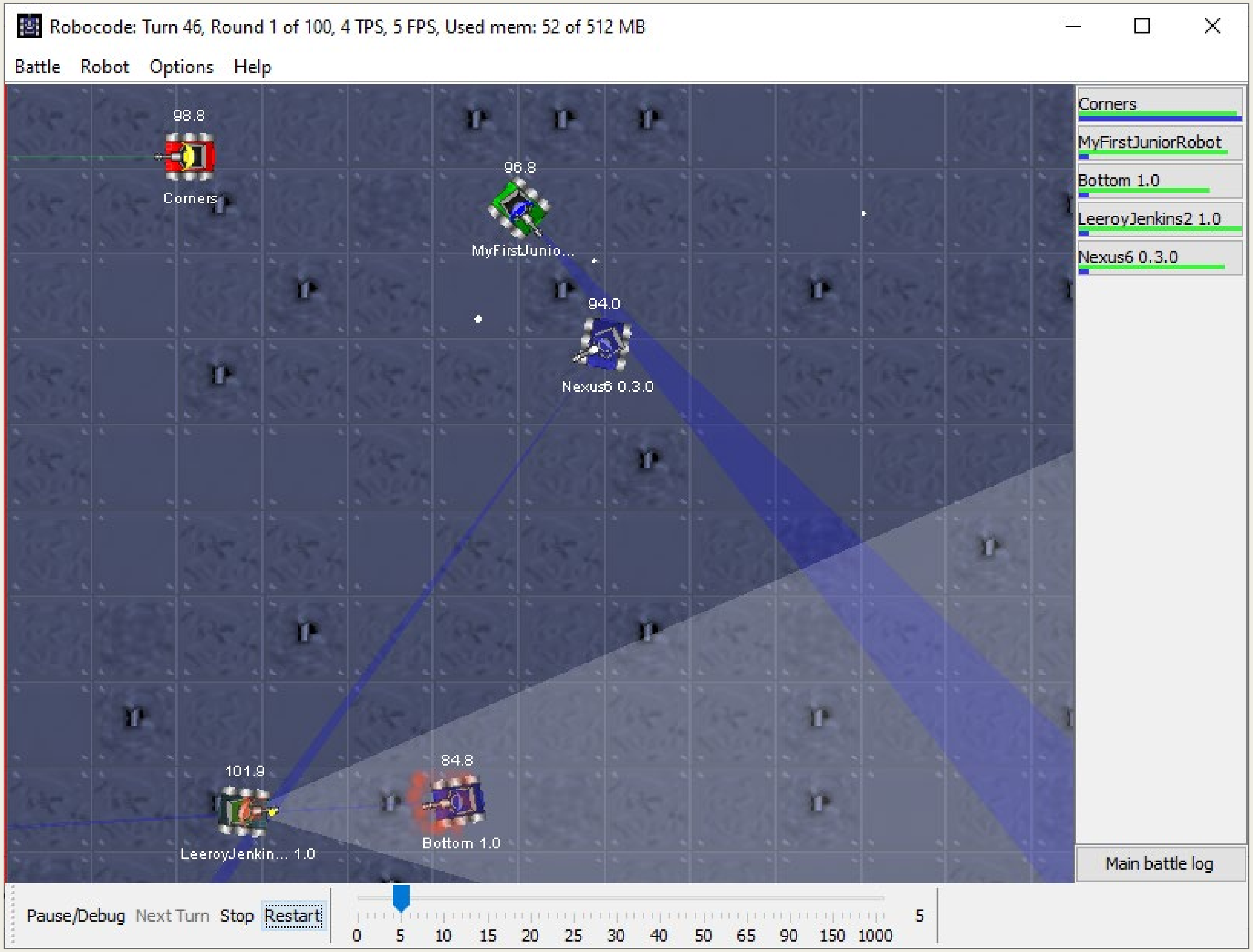


Robot Anatomy

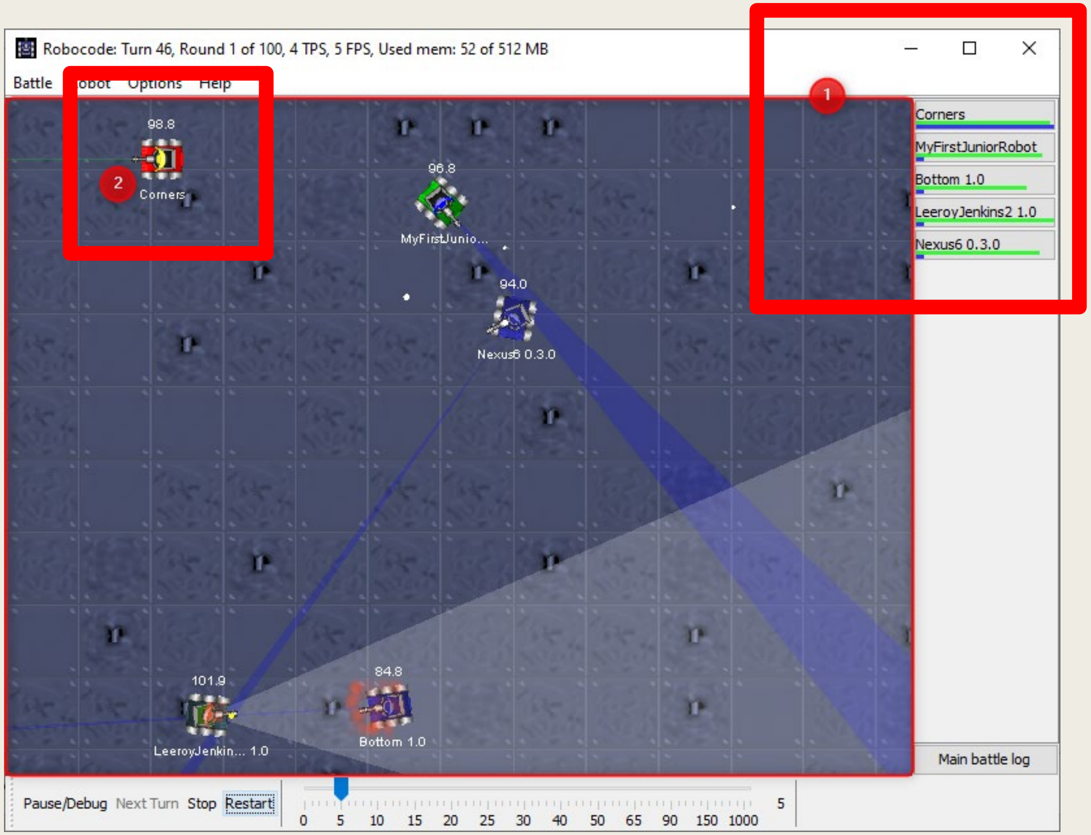
- **Body** – Carries the gun with the radar on top. The body is used for moving the robot ahead and back, as well as turning left or right.
- **Gun** – Mounted on the body and is used for firing energy bullets. The gun can turn left or right. Carries the radar on top.
- **Radar** – Mounted on the gun and is used to scan for other robots when moved. The radar can turn left or right.
- Each component has a set of associated strategies



The battlefield



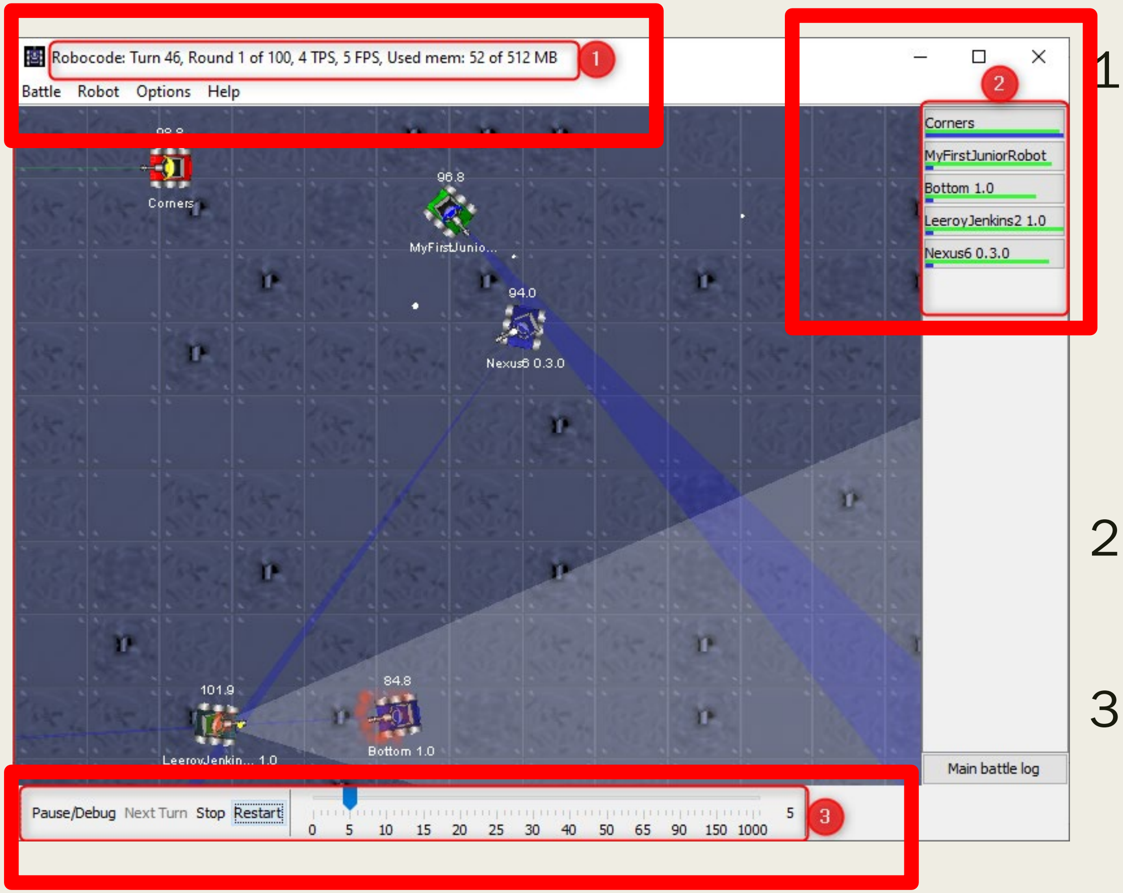
The battlefield



Robocode: Turn 46, Round 1 of 100, 4 TPS, 5 FPS, Used mem: 52 of 512 MB

1. Simulated battle
   - *Different robots*
   - *Radar*
   - *Bullets*
   - *Hit by bullet*
2. Individual robot
   1. *Health points (=98.8)*
   2. *Name (=Corners)*

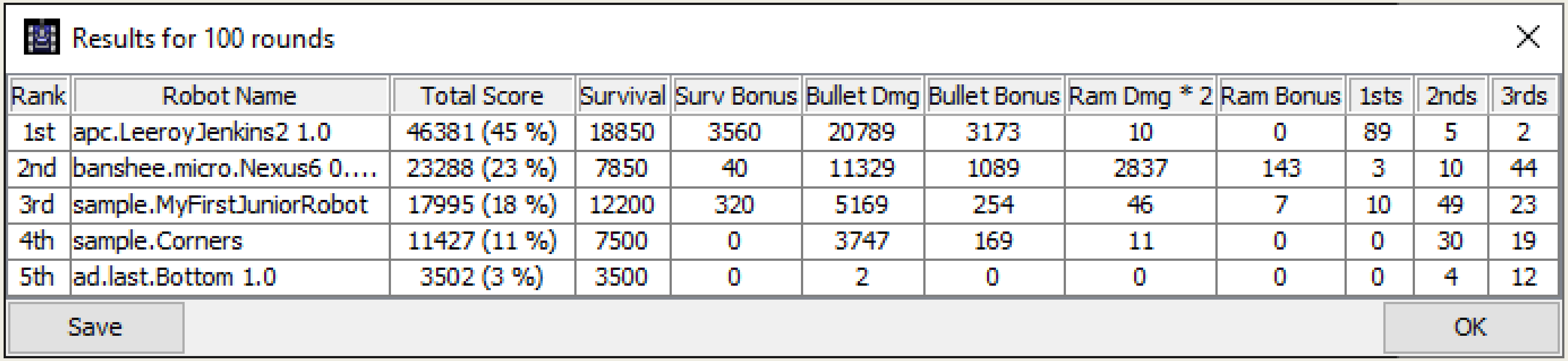
The battlefield



Robocode: Turn 46, Round 1 of 100, 4 TPS, 5 FPS, Used mem: 52 of 512 MB

1. Simulator information
   - *Turns/Ticks (time measure)*
   - *Round*
   - *FPS (1Tick per FPS)*
   - *Memory*
2. Robot information
   - *Name and health*
3. Simulation settings
   - *Pause/stop/restart*
   - *FPS setting*

The battlefield


Results for 100 rounds

Rank	Robot Name	Total Score	Survival	Surv Bonus	Bullet Dmg	Bullet Bonus	Ram Dmg * 2	Ram Bonus	1sts	2nds	3rds
1st	apc.LeeroyJenkins2 1.0	46381 (45 %)	18850	3560	20789	3173	10	0	89	5	2
2nd	banshee.micro.Nexus6 0....	23288 (23 %)	7850	40	11329	1089	2837	143	3	10	44
3rd	sample.MyFirstJuniorRobot	17995 (18 %)	12200	320	5169	254	46	7	10	49	23
4th	sample.Corners	11427 (11 %)	7500	0	3747	169	11	0	0	30	19
5th	ad.last.Bottom 1.0	3502 (3 %)	3500	0	2	0	0	0	0	4	12

- Rank
- Robot Name
- Total Score

- Points for
  - *Survival*
  - *Survival Bonus*
  - *Bullet Damage*
  - *Bullet Bonus*
  - *Ram Damage*
  - *Ram Bonus*
- Times won on position X

Live Demo

- With robots
  - *Sample.MyFirstRobot*
  - *Sample.Corners*
  - *Sample.RamFire*
  - *Sample.Crazy*

MyFirstRobot - Code explained

```
package pkg;

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);
    }
}

```

- **“Package”** for organizational purposes of robots in RoboCode Simulator
- **Import RoboCode library** to receive access to its functions

MyFirstRobot - Code explained

```
package pkg;

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);
    }
}

```

- **“Main” class** of your robot
- **Extends Robot** / AdvancedRobot
  - *Robot = blocking calls*
  - *Ad.Robot = non-blocking calls*

MyFirstRobot - Code explained

```
package pkg;

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);
    }
}
```

- **Run()** for robot configuration
- **While-loop** contains basic behavior; always executed when no on-events, e.g. onScannedRobot
- In this example:
  - *Continue loop until health points run out or onScannedRobot is called*

MyFirstRobot - Code explained

```
package pkg;

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);
    }
}
```

- **Ahead** – Pixels to move
- **turnGunRight** – Degree to turn gun attached radar
- In this example:
  1. *Move ahead 100 pixels.*
  2. *Turn the gun right by 360 degrees.*
  3. *Move back 100 pixels.*
  4. *Turn the gun right by 360 degrees again.*

MyFirstRobot - Code explained

```
package pkg;

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);
    }
}

```

- **Event handling code** on certain event and implementation about action to take
  - ***onScannedRobot()***
  - ***onHitByBullet()***
  - ***onHitWall()***
  - ***...***
- Callout contains information about event, such as scanned enemy robot

Further reading

- Learn more about movement, targeting and firing

  Highly recommended:

  - *https://www.ibm.com/developerworks/java/library/j-robocode/*
  - *https://robocode.sourceforge.io/docs/robocode/*

  Recommended:

  - *https://www.ibm.com/developerworks/java/library/j-robocode2/j-robocode2-pdf.pdf*
  - *http://robowiki.net/ (Chalmers Mirror) -> Radar, Targeting, Movement, Tutorials*

Further reading

- RoboCode FAQ
- Basic knowledge in trigonometry (used to targeting, movement and avoid getting hit): https://www2.clarku.edu/faculty/djoyce/trig/
- Secrets from the Robocode masters https://robocode.sourceforge.io/developerWorks.php
- http://mark.random-article.com/robocode/
- Interests of research
  - *Applying Machine Learning to Robocode*
  - *Deep Q-Learning for Robocode*

Hands on Robocode!!
DIY/20’

Hands on Robocode!! (DIY/20’~30’)

- Download and install required robocode (https://robocode.sourceforge.io)
- Build your first own robot
- Run a competition with your own robot

<u>Any questions?</u>

1st ask your colleagues
2nd raise your hand
3rd look for your TA

<u>Are you done already?</u>

1st answer mentimeter
2nd help your teammates
3rd wait till next session.

mentimeter: 7740 1922

Install FeatureIDE + Eclipse DIY/20’~30’

Install FeatureIDE + Eclipse (DIY/20’~30’)

- Download and install FeatureIDE (https://featureide.github.io/)
  Versioning: Java 17, FeatureIDE 3.9, and Eclipse 4.20
  - *Install Java 17 in your computer*
  - *Enter the FeatureIDE website and download (newest version)*
  - *Open Eclipse, create a new FeatureIDE Project (fiddle around)*

Any questions?

1st ask your colleagues
2nd raise your hand
3rd look for your TA

Are you done already?

1st answer mentimeter
2nd help your teammates
3rd wait till next session

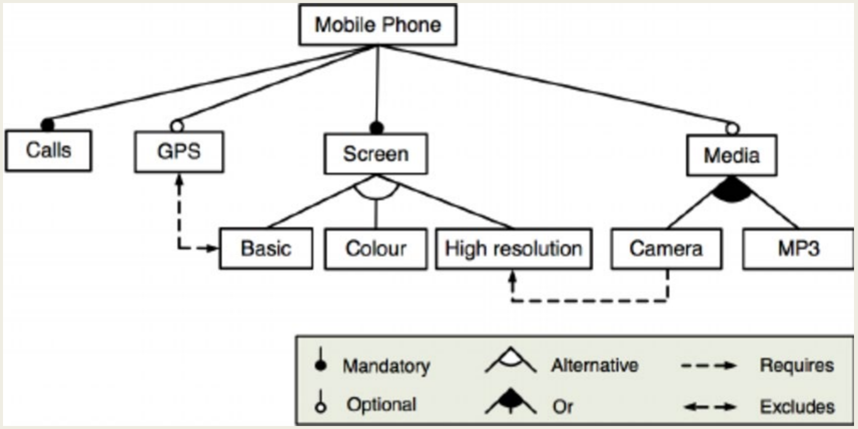
mentimeter: 7740 1922

Guided Example with Antenna 20’

Ricardo Caldas, Mazen Mohamad, Wardah Mahmood,
Malik Hannan, Kevin Ayad

FeatureIDE

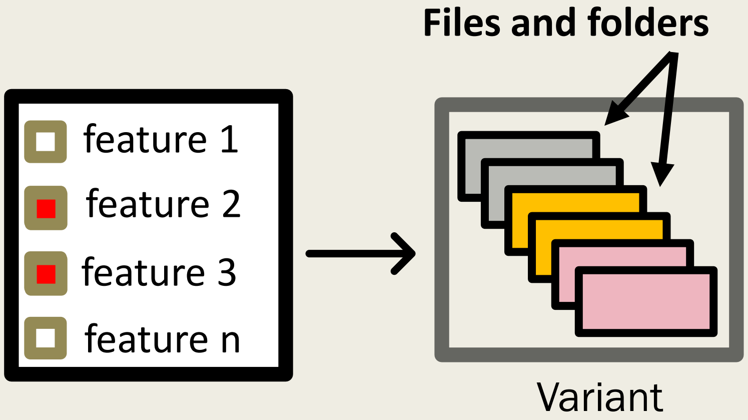
- A Java plugin
- Allows you to:
  - Create feature models
  - Add features in code



- **Feature model:** Tree like structure with features
- **Feature:** Anything meaningful to a user

Why add features into code?

- To allow *configuration*
- **Configuration:** Generating variants with selected features.
- **Sneak peak into assignment 3: Find** features in Robocode codebase, and label them. Generate Robocode variants.



mentimeter: 7740 1922

Guided Example with Antenna (20’) – Follow along!

- [New Project] >> Example… >> FeatureIDE Examples
- Expand Antenna >> Select HelloWorld-Antenna >> Finish
- Change the Feature Model (model.xml)
- Change the Configuration File (BeautifulWorld.xml)
- Insert new feature in the code (HelloWorld.java)
- Observe that changing the configuration also changes the code! Isn’t it cool?!

Are you done already?

1st answer mentimeter
2nd help your teammates
3rd wait till next session

mentimeter: 7740 1922

Gitlab Repo Assessment
(5’)

Can you access the Gitlab repo assigned to your group? (5’)

- Try to enter the following webpage:

https://git.chalmers.se/courses/tda594/2022_groups/group-X

where X is the number or your group

Didn’t work??

Call your TA.

Did it work??

Answer mentimeter and submit your answer.

mentimeter: 7740 1922

Thanks for today!

Ricardo Caldas (ricardo.caldas@chalmers.se)
Mazen Mohamad (mazenm@chalmers.se)
Wardah Mahmood (wardah@chalmers.se)
Malik Hannan (gusmalikah@student.gu.se)
Kevin Ayad (gusayake@student.gu.se)

CHALMERS | UNIVERSITY OF GOTHENBURG