Clean Unit Tests

XP Days Benelux 2010

Lars Vonk, lvonk@zilverline.com
Rob Westgeest, rob@qwan.it

Contents

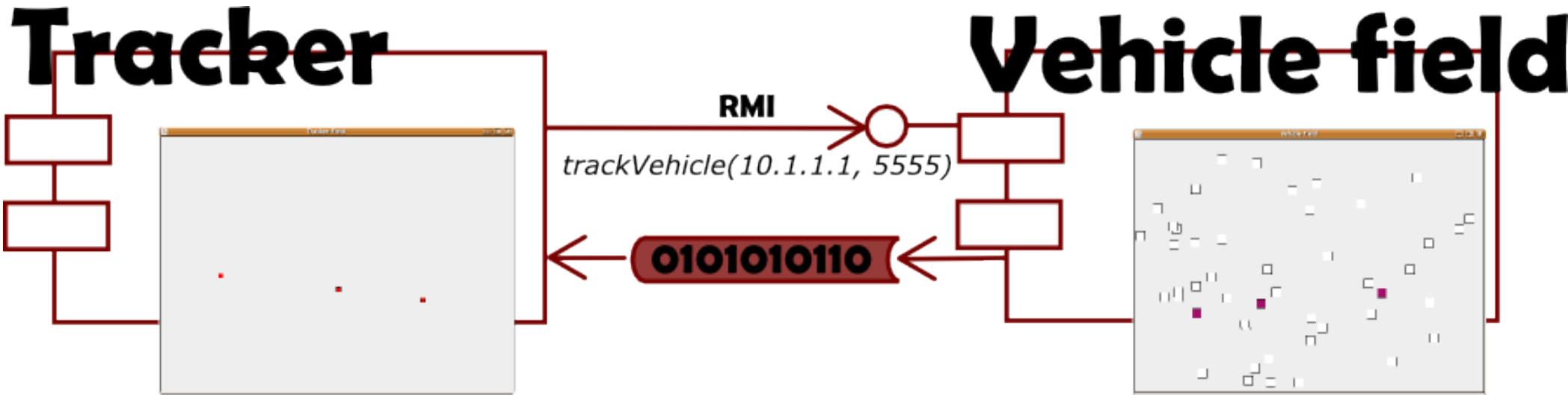
- (5) Introduction
- (20) Gallery session
- (10) Gathering unit testing qualities
- (20) Test improvement kata
- (5) Micro retro

Gallery

Gathering Unit Test Qualities

Test Improvement Kata

Tracking Protocol



Compression In protocol:

F | P | P | P | F | P | P | P | F | P | P | P | F | P | P | P |

```
public class SpeedFrameEncodingTest extends AbstractDeltaFrameEncodingTest {
    private int speed = 0;

    protected byte[] createFrame(byte identification, int timestamp) {
        return new SpeedFrame(identification, timestamp, speed).encode();
    }

    @Test
    public void itContainsSpeedData() {
        assertEquals(speed, CodecUtils.decodeInt16(getFrame(), Protocol.SpeedIndexInDelta));
    }

    @Test
    public void itIsASpeedFrame() {
        assertEquals(Protocol.PartialSpeedFrameLength, getFrame().length);
        assertEquals(Protocol.PartialSpeedFrameLength, getFrame()[Protocol.FrameLengthIndex]);
        assertEquals(Protocol.PartialSpeedFrame, getFrame()[Protocol.FrameTypeIndex]);
    }
}
```

```
public class PositionFrameEncodingTest extends AbstractDeltaFrameEncodingTest {
    private int xPos = 1;
    private int yPos = 1;

    protected byte[] createFrame(byte identification, int timestamp) {
        return new PositionFrame(identification, timestamp, xPos, yPos).encode();
    }

    @Test
    public void ittContainsPositionData() {
        assertEquals(xPos, CodecUtils.decodeInt16(getFrame(), Protocol.XPosIndexInDelta));
        assertEquals(yPos, CodecUtils.decodeInt16(getFrame(), Protocol.YPosIndexInDelta));
    }
    @Test
    public void itIsAPositionFrame() {
        assertEquals(Protocol.PartialPositionFrameLength, getFrame().length);
        assertEquals(Protocol.PartialPositionFrameLength, getFrame()[Protocol.FrameLengthIndex]);
        assertEquals(Protocol.PartialPositionFrame, getFrame()[Protocol.FrameTypeIndex]);
    }

}
```

```
public abstract class AbstractDeltaFrameEncodingTest {

    private int timestamp;
    private byte identification;
    private byte[] frame;

    @Before
    public void setup(){
        identification = 0x45;
        timestamp = 500;
        frame = createFrame(identification, timestamp);
    }

    protected abstract byte[] createFrame(byte identification, int timestamp);

    @Test
    public void itIsAFrame() {
        assertEquals(crc(),frame[Protocol.CrcIndex]);
        assertEquals(identification, frame[Protocol.IdentificationIndex]);
        assertEquals(timestamp, CodecUtils.decodeInt32(frame, Protocol.TimeStampIndex));
    }

    private int crc() {
        return Crc8.checksum(frame, Protocol.PayloadIndex, frame.length - Protocol.PayloadIndex);
    }

    public byte[] getFrame() {
        return frame;
    }

}
```

- When your test smells, it's probably the code that stinks.
- Test code need same level of attention as production code.
  - Method names may differ
  - Sometimes a little duplication helps readability

Micro Retro