

Deep Space Orbiter

odyssey in code



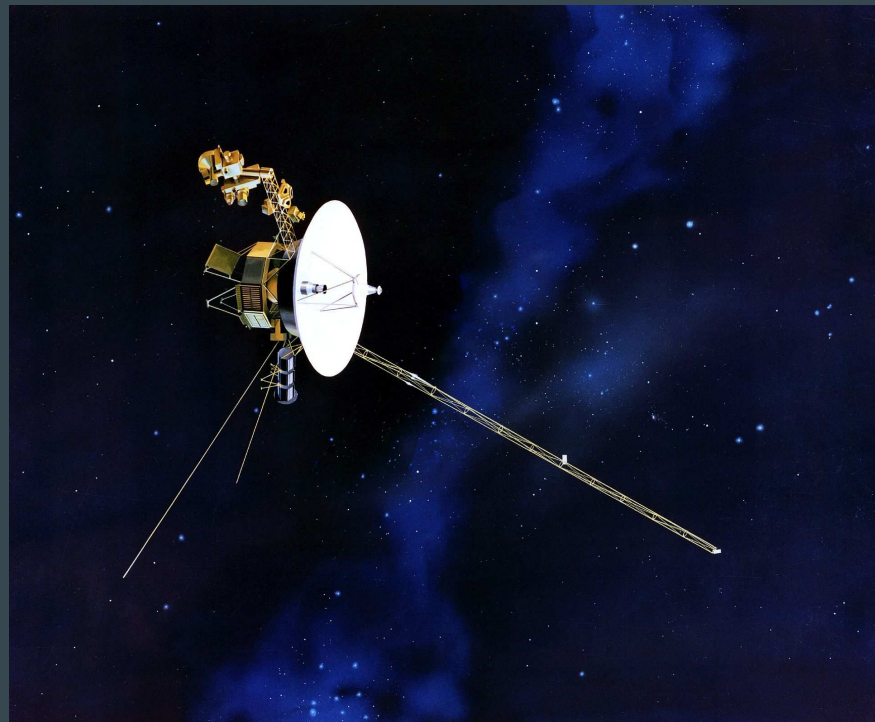
Fabian

Idea

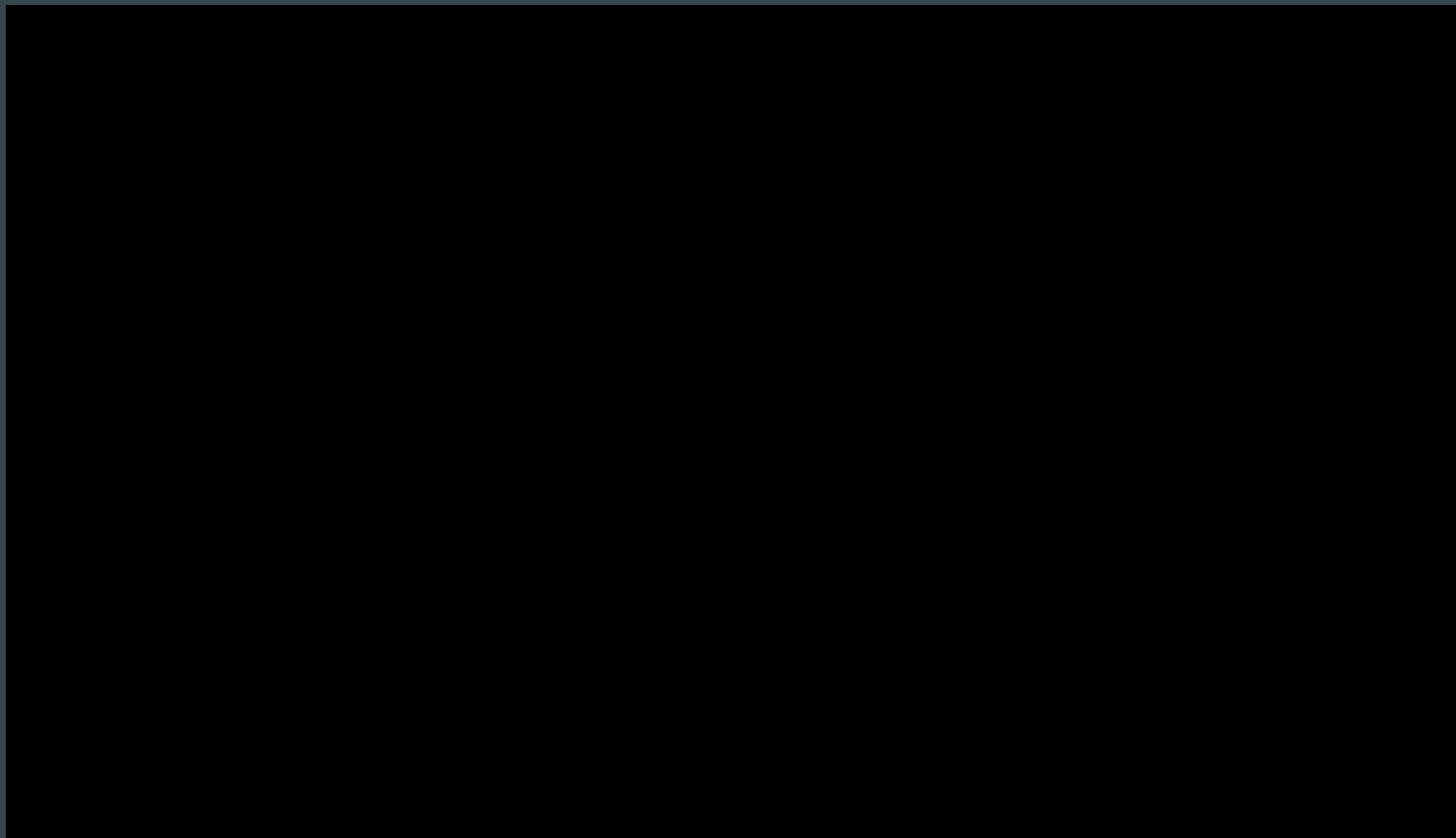
- Implement something with interwoven / complicated “business logic”
- Try Solo development
- Experience evolving requirements
- Experience “the obvious”

And

- Share my experience



So it had to be complicated



Call Getters in Tests? -> Smell?

- `getCameraDirection()` and `getAntennaDirection()` in Testcode
- “Forward-” and “Backward Facing” are internal concepts
- So is `Direction Earth` and `Mars`
- Find another way to test it
 - We can observe the received recordings instead
- -> Write new tests

```
@Test
void recordEarthAfterLaunch() {
    orbiter.updateDistance(newDistance 1);
    assertThat(orbiter.getRecord(), hasItemInArray(EARTH));
}
```

And the nice part

- Delete a lot of old code

```
Git: Local Changes Log History: DeepSpaceOrbiterAntennaShould.java
Fabian Mächler 9/11/21, 12:25 PM r turn antenna earth when recordings full
Fabian Mächler 9/11/21, 12:17 PM f exception if unable to transmit
Fabian Mächler 9/11/21, 11:59 AM f delete getCameraDirection getAntennaDirection
Fabian Mächler 9/11/21, 11:57 AM f only transmit if antenna faces earth
Fabian Mächler 9/11/21, 11:45 AM f make multiple records
Fabian Mächler 9/11/21, 11:36 AM f for simplification, distance = time
Fabian Mächler 9/11/21, 11:30 AM f records
Fabian Mächler 9/5/21, 12:40 PM l Test naming
Fabian Mächler 9/5/21, 12:39 PM f Record Earth at Launch
Fabian Mächler 9/5/21, 12:36 PM f Antenna point in any direction
Fabian Mächler 9/5/21, 12:31 PM f Antenna point in any direction
Fabian Mächler 9/5/21, 12:22 PM r test setup
Fabian Mächler 9/5/21, 12:21 PM f Camera should not face the same direction as antenna in the beginning
Fabian Mächler 9/5/21, 12:09 PM f Camera until halfway to earth then mars
Fabian Mächler 9/5/21, 12:04 PM f Point camera to mars
Fabian Mächler 9/5/21, 11:53 AM r Naming
Fabian Mächler 9/5/21, 11:49 AM f Start recording after Launch
Fabian Mächler 9/5/21, 11:41 AM e Requirements again
Fabian Mächler 9/5/21, 11:40 AM r naming
Fabian Mächler 9/5/21, 11:38 AM f Change requirements to FORWARD/BACKWARD instead of degrees
Fabian Mächler 9/5/21, 11:20 AM f Requirements and first test camera 180

657e488 (DeepSpaceOrbiterAntennaShould.java)
29:06 void setUp() {
31:51 orbiter = new Orbiter();
32:52 }
51:51
54 @Test
55 void startWithCameraFacingEarth() {
56 Direction cameraAngle = orbiter.getCameraDirection();
57
58 assertEquals(cameraAngle, equalTo(EARTH));
59 }
61
62 @ParameterizedTest
63 @CsvSource({
64 "0", EARTH",
65 "45", EARTH",
66 "90", EARTH",
67 "135", MARS",
68 "180", MARS",
69 })
70 void pointCameraTowardsEarthUntilHalfway(int kms, Direction cameraDirection) {
71 orbiter.updateDistance(kms);
72
73 assertEquals(orbiter.getCameraDirection(), equalTo(cameraDirection));
74 }
75
76 @Test
77 void startWithAntennaAndCameraNotFacingTheSameDirection() {
78 assertEquals(orbiter.getCameraDirection(), not(equalTo(orbiter.getAntennaDirection())));
79 }
80
81 @Test
82 void startWithAntennaPointingInADirection() {
83 assertEquals(orbiter.getAntennaDirection(), notNullValue());
84 }
85
86 @Test
87 void recordEarthAfterLaunch() {
88 orbiter.updateDistance(1);
89 assertEquals(orbiter.getRecord(), hasItemInArray(EARTH));
```

Drastic Implementation change without changes in the Tests

- I could change from “CameraDirection and AntennaDirection” to “Slots which a Devices occupies” without changes in the tests

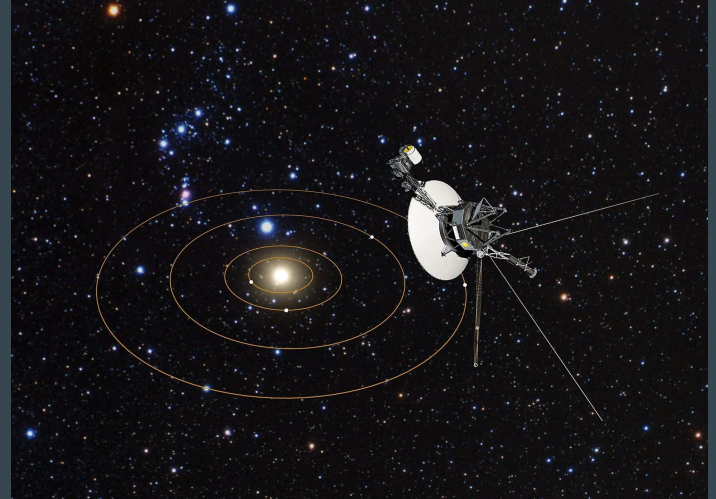
| | | |
|--|---|--|
| <pre>public class Orbiter { public static final int DISTANCE_TO_MARS = 395; private int distance; private Direction cameraDirection; private Direction antennaDirection; private final LinkedList<Direction> records = n public Orbiter() { cameraDirection = EARTH; } }</pre> | <pre>8 9 9 10 10 11 11 12 12 13 13 14 14 15 15 16 16 17 17 18 18 19 19 20 20 21 21 22 22 23</pre> | <pre>public class Orbiter { public static final int DISTANCE_TO_MARS = 395; private Device directionEarthSlot; private int distance; private Direction cameraDirection; private final LinkedList<Direction> records = ne public Orbiter() { directionEarthSlot = CAMERA; } }</pre> |
|--|---|--|

Other observations

- First degree of freedom: Focus on camera direction, not the recording
 - But it's connected (See Getter-Smells)
- Shaping the requirements involved a lot erroring and thus learning
 - Did I make myself to be the only expert in this field?
- Lot of naming is wrong (I didn't notice)
 - Maybe the Mob would help here too
- Change of requirements during implementation
 - Implementing challenges the requirements
- DirectionEartSlot -> I have nothing telling me how to implement this
 - Try and error?
 - Train the production code like a neural network?

State of the Project

- About half of the requirements are implemented
- There are refactorings in the production code open
- But the Tests/Requirements seem to be in good shape to me



Code

You can find the code here: <https://github.com/unSinn/alcor-presentation>