BuqSquashing:

Usual & Unusual Bugs

Bastian Graser
28th Sept. 2011

GERMAN
CANCER RESEARCH CENTER
IN THE HELMHOLTZ ASSOCIATION

Does it compile? Does it work?

```
..
mitk::Image::Pointer outputImage;
makeImage(outputImage);
..
// Put Image in Datamanager
```

```
void makeImage(mitk::Image::Pointer image)
{
 image = mitk::Image::New();
 // Fill Image
}
```

Does it compile? Does it work?

```
..
mitk::Image::Pointer outputImage;
makeImage(outputImage);
..
// Put Image in Datamanager
```

Yes, it compiles!
No, it is not working!

```
void makeIn
{
 image = mitk::Image::New();
 // Fill Image
}
```

Solution: Smartpointers are no pointers

```
mitk::Image::Pointer outputImage = mitk::Image::New();
makeImage(outputImage);
```

```
void makeImage(mitk::Image::Pointer image) {
  // Fill Image
}
```

or

```
mitk::Image::Pointer outputImage;
makeImage(outputImage);
```

```
void makeImage(mitk::Image::Pointer & image) {
 image = mitk::Image::New();
 // Fill Image
}
```

Does it compile? Does it work?

```
class MyFilter
{
public:
   MyFilter(float p);
   setThreshold(float p);
private:
   float threshold;
};
```

```
MyFilter::setThreshold(float p)
{
   if (abs(p-threshold) < 0.01)
      return;
   else
      threshold = p;
}

MyFilter::MyFilter(float p)
{
   this->setThreshold(p);
}
```

```
void main()
{
   myFilter(0.91);
}
```

Does it compile? Does it work?

```
class MyFilter
{
public:
   MyFilter(float p);
   setThreshold(float p);
private:
   floa
};
```

```
MyFilter::setThreshold(float p)
{
   if (abs(p-threshold) < 0.01)
      return;
   else
      threshold = p;

                          loat p)
{
   this->setThreshold(p);
}
```

```
void main()
{
   myFilter(0.91);
}
```

Yes, it compiles!
Yes, it works! (in Debug)

Heisenbug

- A bug that disappears when you try to examine it (named after Heisenberg Uncertainty Principle)
- e.g. Bug occurs in Release but not in Debug. Only on Windows, not on Linux.
- Problems:
  - Might be recognized very late, since it occurs only under certain conditions
  - Cannot use debugger (Since it works in debug)

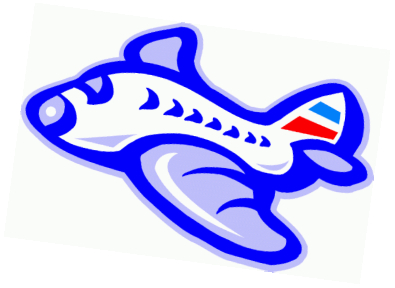
Heisenbug

- Explanation:
  - General Rule: Memory is usually UNINITIALIZED
  - In this example, as long as memory is not initialized as „NAN“, it will work, but not always as expected
  - Here, when running a release under windows, memory is filled with NAN which causes a crash

- How To Deal:
  - Reason is **ALWAYS** a variable which is used before it is initialized!
  - Put COUT‘s everywhere to determine code section where program freezes
  - Use external tools: CHESS (windows), VALGRIND (linux)

Alpha Particle Bug

- You are on the plane to MICCAI and do some last fixes in your code



- Suddenly program freezes
- You cannot find a bug somewhere in the code

Alpha Particle Bug

- **Cosmic Rays** can cause Bit-Flips in RAM
- Can cause program freeze, build errors, memory corruption..
- Chance depends on altitude and chip density
  A cave would be most secure!
- ~1 error per month per 256 MB RAM for Desktop PCs
  - Checksum should cover it

Conclusion

- Also seemingly undeterministic bugs have deterministic reasons