Testing Virtual Reality
(And other complex software environments!)

Dr. Emil Alégroth
Auqtus | Blekinge Institute of Technology | Chalmers University of Technology
• We delivering **state-of-art testing** solutions to industry with focus on Visual GUI Testing (VGT).

• Founded by academics and technical experts in the area of automated testing.
  – Combined; over 50 years of academic experience
  – 100s of academic publications
  – World-leading researchers on VGT (Founded the term)

Contributors to the talk:

Dr. Emil Alégroth  
Prof. Robert Feldt  
Dr. Simon Poulding
What is Virtual Reality?

- Oculus rift and Google Cardboard
- Leisure applications
  - Games
  - Communication
- Learning applications
Our application: VRHangout
Unity

• Game engine
• All in-game fixtures treated as objects
  – Makes scripting easy
• C# scripting
  – Quick and powerful
  – Same syntax as java
• Many plug-ins
  – Google cardboard!
• Main limitation
  – Standard components are generally slow!

Unity game: Hearthstone
My experiences with Unity
(5h vs 5 days)
VR is only the beginning!
Augmented reality

- Leisure applications (e.g. Minecraft)
- Life-critical applications
  - Medical systems

Safety requirements will require stringent quality assurance!
How do we test in this environment?

• Low-level:
  – Automated unit tests and integration tests
  – C# components (in Unity)

• High-level:
  – Manual testing 😞

• Why?
  – The real-time, virtual environment, is (often) non-deterministic and complex to test
  – Complexity: Computationally heavy to test, difficult to find oracles, etc.
A complex environment requires complex testing

• Search-based software testing (SBST)
  – Test case generation, based on a (*)fitness function, that can automatically determine if the output of the generated test case is (**)correct/faulty/interesting to/for the tester.

• What is a (*)?
  – Domain model that defines rules for how to, for instance, generate suitable test cases (A heuristic). Can be modified to learn during execution (Machine learning).

• How do we determine (**) in a VR environment?
  – For instance: Kolmogorov complexity!
Test case generation with Meta-Heuristics

- **Meta-heuristic**
  - A description of a possible solution to generate test cases
  - *For instance:* Genetic algorithms

![Diagram of test case generation]

- **Gen 1 seed**
- **Gen 1**
- **Gen 2 seed**
- **Determine best child with a fitness function (best fit)**

- T1
- T2

- **Gen 1 seed**
- **Gen 1**
- **Gen 2 seed**
- **Determine best child with a fitness function (best fit)**

- T1
- T2
Oracle?

- Oracle
  - A means of determining test case outcome
  - Examples: Value, image, strings or other output
- Kolmogorov complexity allows you to determine a distance between any two objects!

- The measure gives a distance between two objects
  - Determine if a state is potentially correct/faulty/interesting
Kolmogorov complexity example usage
SBST + Kolmogorov

• Generate exploratory test cases, verify with the automated oracle and learn by testing!

• Small problem…
  – Kolmogorov complexity cannot be computed!
  – Theoretical concept...

  …BUT! It can be approximated with compression!

\[
\text{Compress}(1,[1,3,4,5]), (2,[1,4,5]) = ([1,4,5], (1,[3]))
\]

\[
\text{Compress}(\text{"Distance"},) = (\text{"Distance"},(255,0,0),(0,0,255))
\]

No tools do this yet but they are on the way!
VR and AR are tomorrow. We need tools/techniques today!

### CHALLENGES

- System model
  - GUI
  - GUI model
  - System architecture
  - System components / Source code

- Speed
- Robustness
- Incompleteness
- Interoperability
- Usability
- Distributed system

- Multiple languages
- Missing interfaces
- Too much code
- Etc.

- Encapsulated design
- Custom components
- Limited usability
- Maintenance costs

- More support

- Stronger need
  - (e.g. Continuous development)

- Failures
- Defects
Visual GUI Testing (VGT): JAutomate

- Seamless testing of different applications
- End-to-end testing
- Advanced testing through the GUI
- Non-functional testing (e.g. Performance)
- Etc.
VGT: Primarily System- and Acceptance Regression testing

• VGT-GUITAR
  – Automated Exploratory, model-based, GUI testing
  – State space explosion problems

• What if we could provide the tester with a heads-up display to improve exploratory testing based on user domain knowledge?

• What if we could automatically record exploratory test cases for regression testing?
Augmented exploratory testing

Circles: covered states

Color: Likelyhood

Expand:

Fantastic!!!

but is it a Fantasy???
Current state
Neural networks combined with advanced search algorithms

Neural net for Version X+1

If advanced model, automatic exploration!

COMING SOON!
Conclusions

- VR and AR require more complex test techniques
- Search-based testing with Kolmogorov complexity
  - Not available yet!
- Visual GUI Testing
- Augmented Exploratory Testing


