Epilogue

Alexandre Bergel
http://bergel.eu
21/12/2020

Intense semester

We have covered many things during the semester

The Java programming Language: classes, methods, message, this & super, method lookup, interface, generics

Design patterns: creational, structural, behavioral

Unit testing: structuring tests, writing tests, coverage

Programming methodology: test-driven design, test coverage, object-oriented design

Artifacts

Few examples

- TicTacToe & other

Many libraries

- JUnit (3, 4, 5),
- JavaFX,
- Files,
- Math,
- …

Development tools

- IntelliJ
- GitHub
- Debugger

What we have not seen

How to work in a team

- This has impact on the code versioning
- Tasks and requirements management
- Module dependencies (e.g., using Maven)

Java SE vs Java EE

- Java EE is a superset of Java SE
- Useful to make web applications
- Java EE is frequently used (particularly large industries)

What we have not seen

Generics

- We have seen just the easy part, how to use it
- Generics is important to fully exploit JavaFX

JavaFX

- Again, we have seen just the basic functionalities
- JavaFX allows a UI to be exposed in a web browser
- Lambdas for callbacks

Distributed Computing

- Restful API
- Remote Method Invocation (RMI)

What we have not seen

Persistency

- Databases, JSON, XML, binary serialization are extremely important

Concurrency (Threads…)

Measuring and improving performance

- Effect of different caches are important to efficiently

CC3002 in practice

The *techniques and methodologies* we have seen during the semesters are used in large IT companies

Unfortunately, the Chilean IT industry *is still not using* much testing and design patterns

You can see this positively, as *you will have value* when you will enter a company

What will the future be?

Nobody knows…

15 years ago, the content of this class was very relevant

It is still relevant today

It will probably be relevant in 15 years

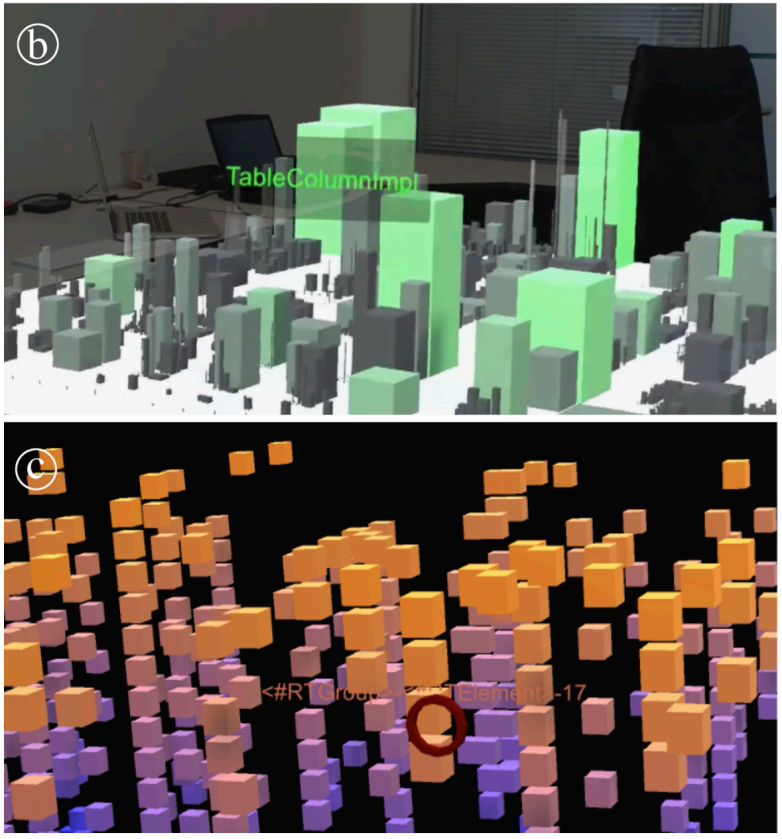
What will the future be?

But there are actives research communities

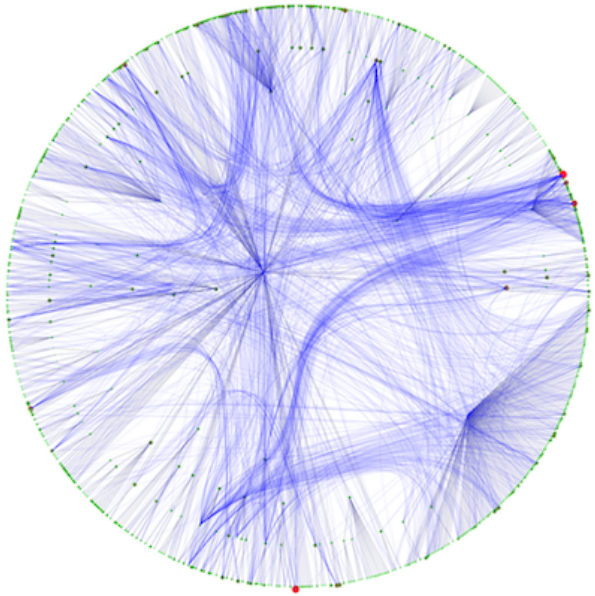
- embedding programming language in another (C# and SQL)
- code generation (e.g., Ruby on Rails, extensions of JavaScript)
- automatically generating tests (e.g., random testing, mutation testing)
- cloud and multi-core machines may change completely the way we design software
- Applying artificial intelligence techniques to software engineering

Glimpse of the research we do

Augmented Reality for software engineering



Software visualization



- Chatbots
- Genetic programming to fix bugs

...

That’s all folk!

Memorable semester

Big thanks to yourself! 👏👏

Big thanks to the auxiliaries and ayudantes! 👏👏

Programming is fantastic way to express and concretize ideas

Keep learning!

Maybe we will have opportunities to work together

Trabajo dirigido

Curso de Neural Networks y Programación Genética (CC5114)