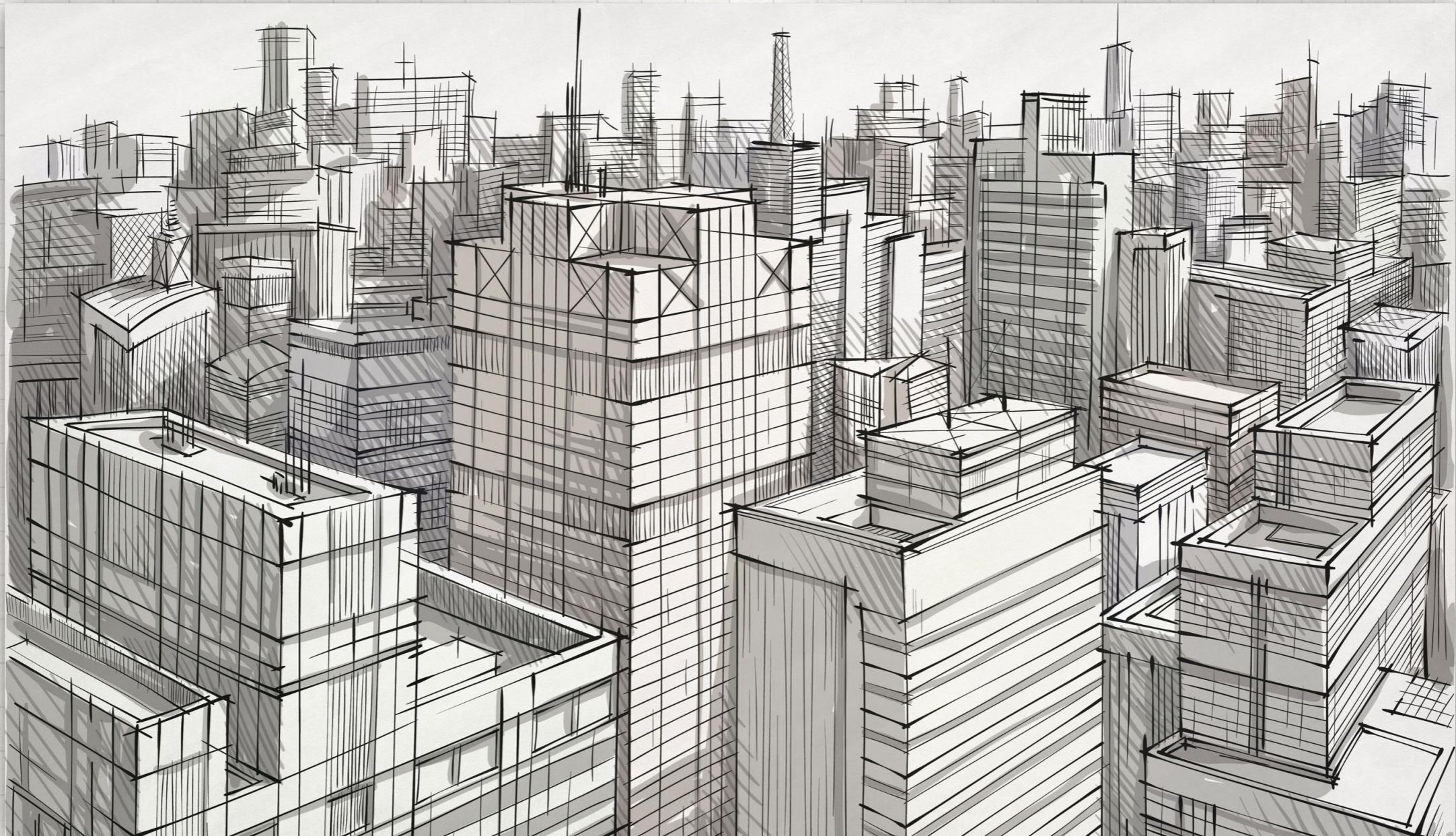


UNIT, INTEGRATION AND NSUBSTITUTE

TEST DRIVEN DEVELOPMENT IN UNITY3D FOR CHIMPS



by Valentin Burov

YOU'RE EITHER ON IT...OR YOU HAVEN'T TRIED IT

TESTING WITH ADDICTING EVENT HOOKS

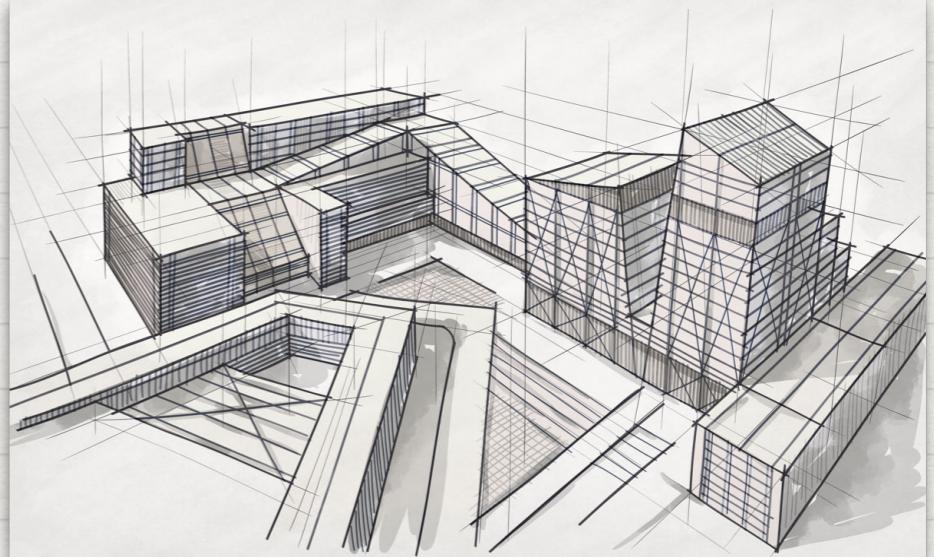
- Eloquence not required
- Single responsibility pattern is
- Press button, get banana user behaviour simulation

INTEGRATION TESTING

UI IS A GOOD USE, SO ARE ASYNC SCENARIOS

- Great for coded UI testing
- Not kiddie-scripting
- Complex Scenarios
- Coroutine-friendly:

The tests are executed in the editor, in non-play mode, all in one frame. It is not possible to skip a frame and/or execute an API that requires skipping frames.



INTEGRATION TEST CASE

BUTTON PRESS DEACTIVATES IMAGE COMPONENT

- User waits some time
- User presses button
- Image of banana is shown

TEST STRUCTURE

PSEUDOCODE

- Subscribe to event onEnable
- StartCoroutine, wait variable amount of seconds with a callback
- Invoke a method that enables an image

UNIT TESTING

PURE CODE

- No need to run the entire app
- No UIs
- Eliminates false positives and false negatives
- All in one frame
- Write test first, code second
- Break it to smallest chunks - units
- Uses its own syntax
- Not difficult - different



ORGANIZATION

ARRANGE, ACT, ASSERT

- Set up scenario
- Make a hypothesis
- Validate
- Lean Startup style

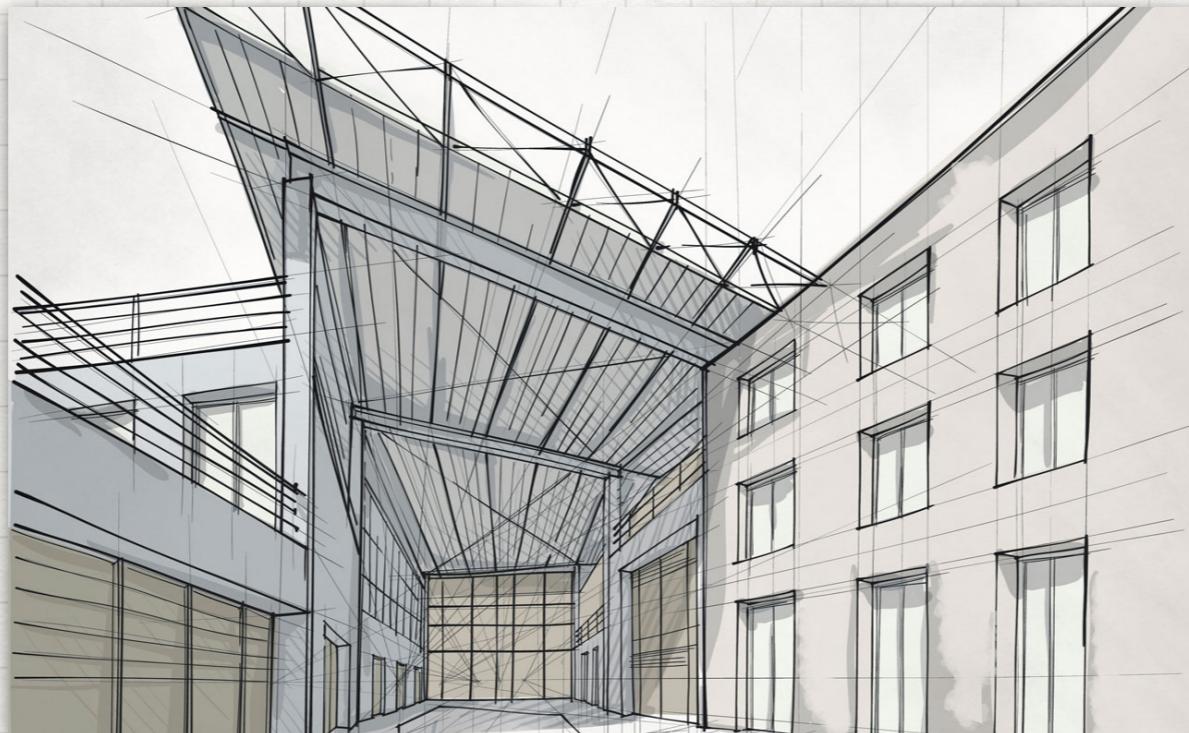
UNIT TEST STRUCTURE

PSEUDOCODE

- Create new GameObject, and add Image Component
- Deactivate Image Component
- Assert that it has been turned off

UNIT TESTING WITH NSUBSTITUTE

MOCKUPS



- Create robust mockups
- Work through interfaces instead of instances
- Combine with IoC Systems

SUBSTITUTIONS AND STUBS

REPLACES DUMMY CLASSES

- Use an interface
- Create substitute object proxy
- Change the behaviour of the proxy
- Assert a condition

CONCLUSION

CLASSICAL BEHAVIOUR



- Procedural banana
- Integration for visual and coroutines
- Unit for pure code
- NSubstitute for Mockups