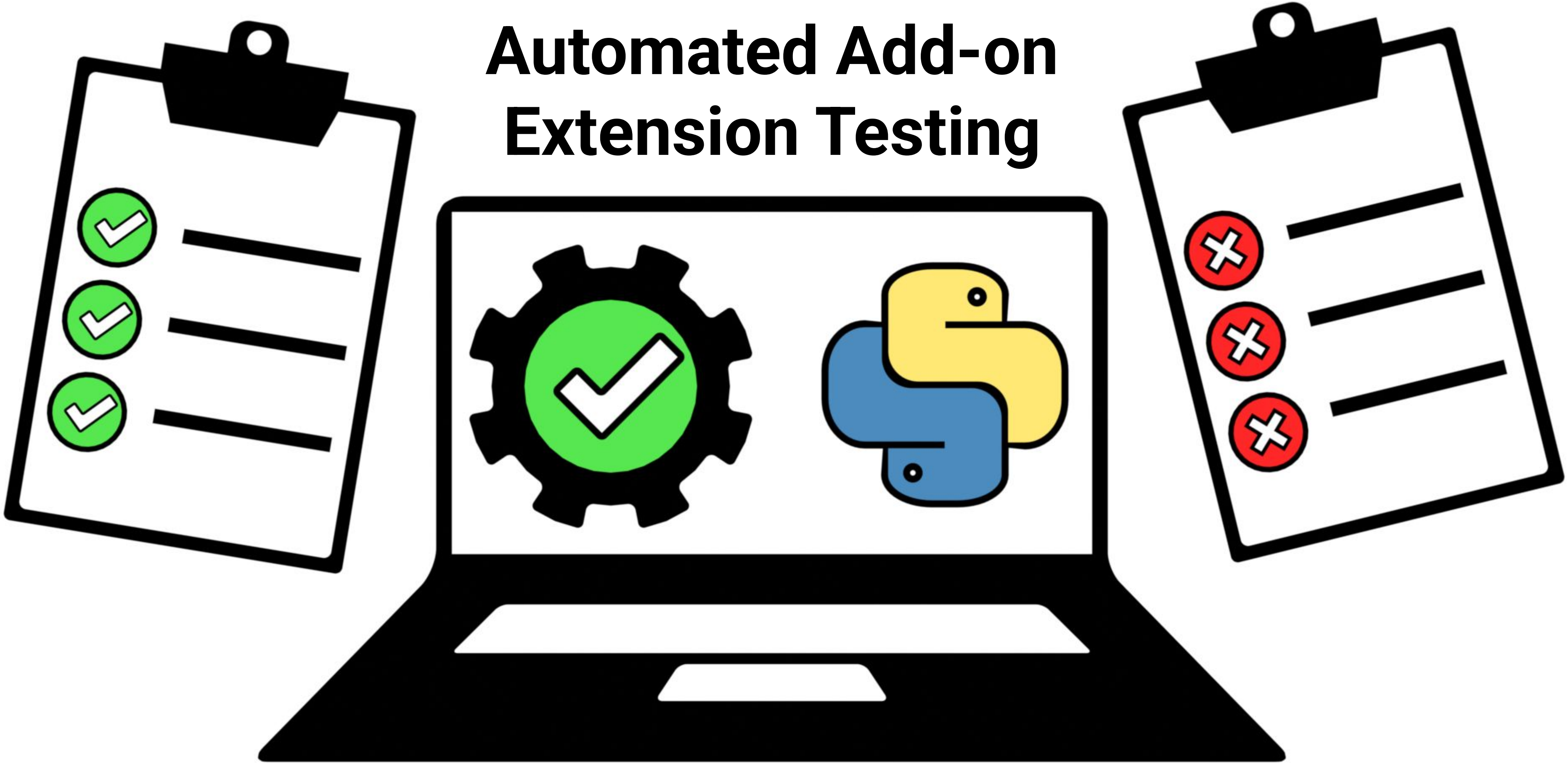
Automated Add-on Extension Testing



Spencer Magnusson

Overview

1. Benefits and the "why"
2. Designing Tests
3. Improve your Testing Workflow
4. Testing Blender without (and with) UI
5. Code demo/example (with Light Painter)

Ever had to do a hotfix?

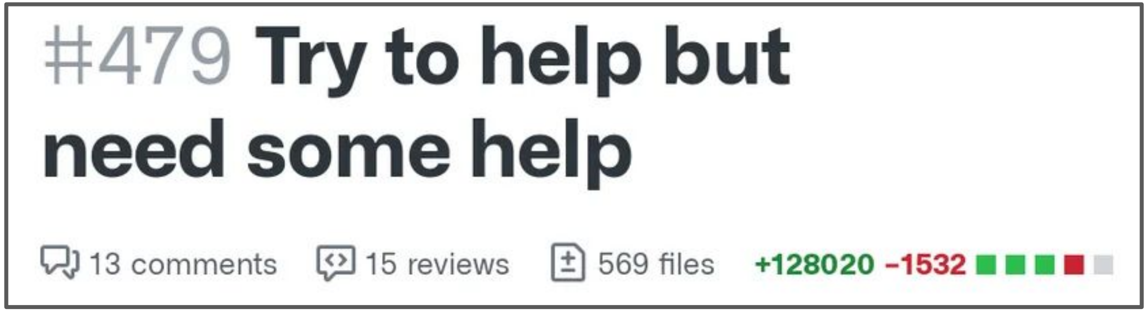
… only to reintroduce an old bug?

Ever added a feature

… changing over a dozen files?

Or refactor all your add-on code

… without knowing it will work?



#479 **Try to help but need some help**

13 comments 15 reviews 569 files +128020 −1532 🟩🟩🟩🟥⬜

What are your options?

1. Make your add-on smaller
2. Users = testers
3. Manually test
4. Automate your testing!

nVIEW

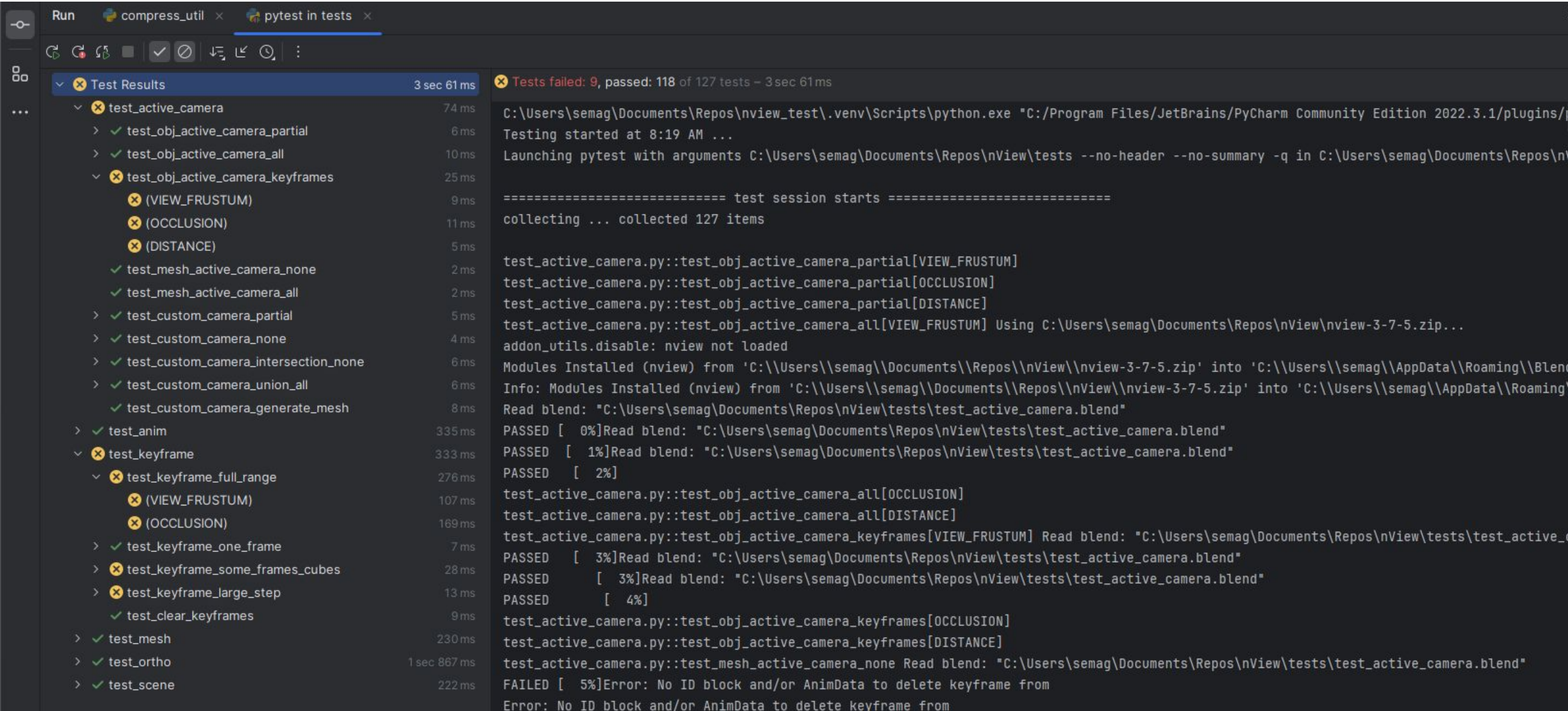
LIGHT PAINTER

Benefits of Scripted Testing

1. **Consistency**

(less human error in testing)

2. Fast feedback



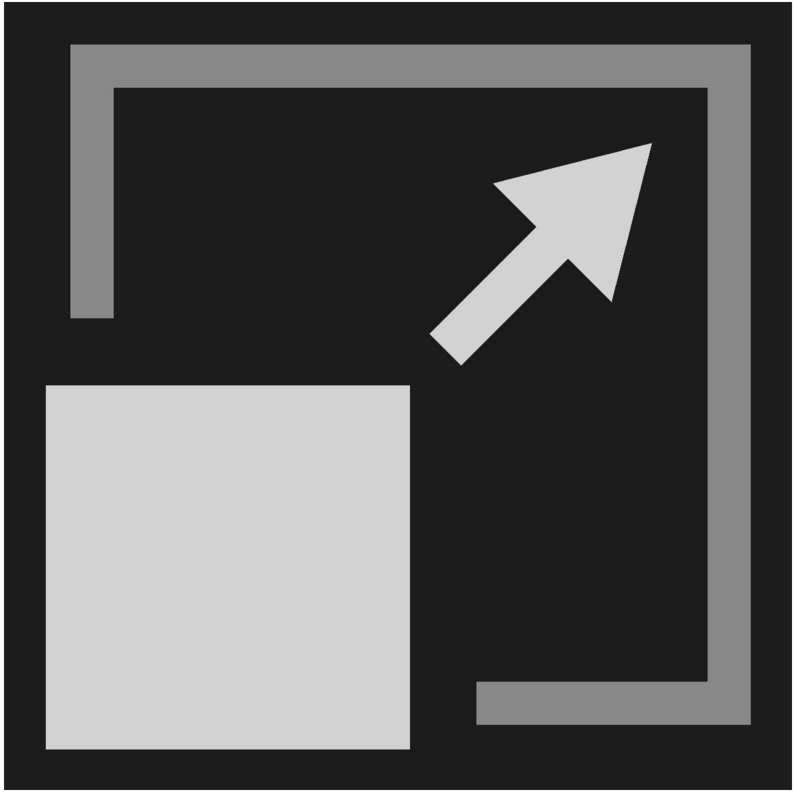
```
⊗ Test Results                                   3 sec 61 ms
  ⊗ test_active_camera                           74 ms
    > ✓ test_obj_active_camera_partial           6 ms
    > ✓ test_obj_active_camera_all               10 ms
    ⊗ test_obj_active_camera_keyframes           25 ms
        ⊗ (VIEW_FRUSTUM)                         9 ms
        ⊗ (OCCLUSION)                            11 ms
        ⊗ (DISTANCE)                             5 ms
      ✓ test_mesh_active_camera_none             2 ms
      ✓ test_mesh_active_camera_all              2 ms
    > ✓ test_custom_camera_partial               5 ms
    > ✓ test_custom_camera_none                  4 ms
    > ✓ test_custom_camera_intersection_none     6 ms
    > ✓ test_custom_camera_union_all             6 ms
      ✓ test_custom_camera_generate_mesh         8 ms
  > ✓ test_anim                                  335 ms
  ⊗ test_keyframe                                333 ms
    ⊗ test_keyframe_full_range                   276 ms
        ⊗ (VIEW_FRUSTUM)                         107 ms
        ⊗ (OCCLUSION)                            169 ms
    > ✓ test_keyframe_one_frame                  7 ms
    > ⊗ test_keyframe_some_frames_cubes          28 ms
    > ⊗ test_keyframe_large_step                 13 ms
      ✓ test_clear_keyframes                     9 ms
  > ✓ test_mesh                                  230 ms
  > ✓ test_ortho                                 1 sec 867 ms
  > ✓ test_scene                                 222 ms
```

⊗ Tests failed: 9, passed: 118 of 127 tests – 3 sec 61 ms

```
C:\Users\semag\Documents\Repos\nview_test\.venv\Scripts\python.exe "C:/Program Files/JetBrains/PyCharm Community Edition 2022.3.1/plugins/p
Testing started at 8:19 AM ...
Launching pytest with arguments C:\Users\semag\Documents\Repos\nView\tests --no-header --no-summary -q in C:\Users\semag\Documents\Repos\n

============================= test session starts ==============================
collecting ... collected 127 items

test_active_camera.py::test_obj_active_camera_partial[VIEW_FRUSTUM]
test_active_camera.py::test_obj_active_camera_partial[OCCLUSION]
test_active_camera.py::test_obj_active_camera_partial[DISTANCE]
test_active_camera.py::test_obj_active_camera_all[VIEW_FRUSTUM] Using C:\Users\semag\Documents\Repos\nView\nview-3-7-5.zip...
addon_utils.disable: nview not loaded
Modules Installed (nview) from 'C:\\Users\\semag\\Documents\\Repos\\nView\\nview-3-7-5.zip' into 'C:\\Users\\semag\\AppData\\Roaming\\Blend
Info: Modules Installed (nview) from 'C:\\Users\\semag\\Documents\\Repos\\nView\\nview-3-7-5.zip' into 'C:\\Users\\semag\\AppData\\Roaming
Read blend: "C:\Users\semag\Documents\Repos\nView\tests\test_active_camera.blend"
PASSED [  0%]Read blend: "C:\Users\semag\Documents\Repos\nView\tests\test_active_camera.blend"
PASSED  [  1%]Read blend: "C:\Users\semag\Documents\Repos\nView\tests\test_active_camera.blend"
PASSED   [  2%]
test_active_camera.py::test_obj_active_camera_all[OCCLUSION]
test_active_camera.py::test_obj_active_camera_all[DISTANCE]
test_active_camera.py::test_obj_active_camera_keyframes[VIEW_FRUSTUM] Read blend: "C:\Users\semag\Documents\Repos\nView\tests\test_active_c
PASSED   [  3%]Read blend: "C:\Users\semag\Documents\Repos\nView\tests\test_active_camera.blend"
PASSED      [  3%]Read blend: "C:\Users\semag\Documents\Repos\nView\tests\test_active_camera.blend"
PASSED       [  4%]
test_active_camera.py::test_obj_active_camera_keyframes[OCCLUSION]
test_active_camera.py::test_obj_active_camera_keyframes[DISTANCE]
test_active_camera.py::test_mesh_active_camera_none Read blend: "C:\Users\semag\Documents\Repos\nView\tests\test_active_camera.blend"
FAILED [  5%]Error: No ID block and/or AnimData to delete keyframe from
Error: No ID block and/or AnimData to delete keyframe from
```

3. Higher Quality Code

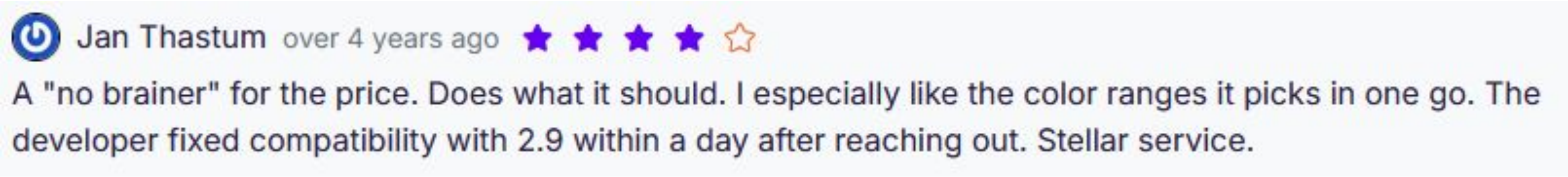
(less human error in coding)



4. Scalable*

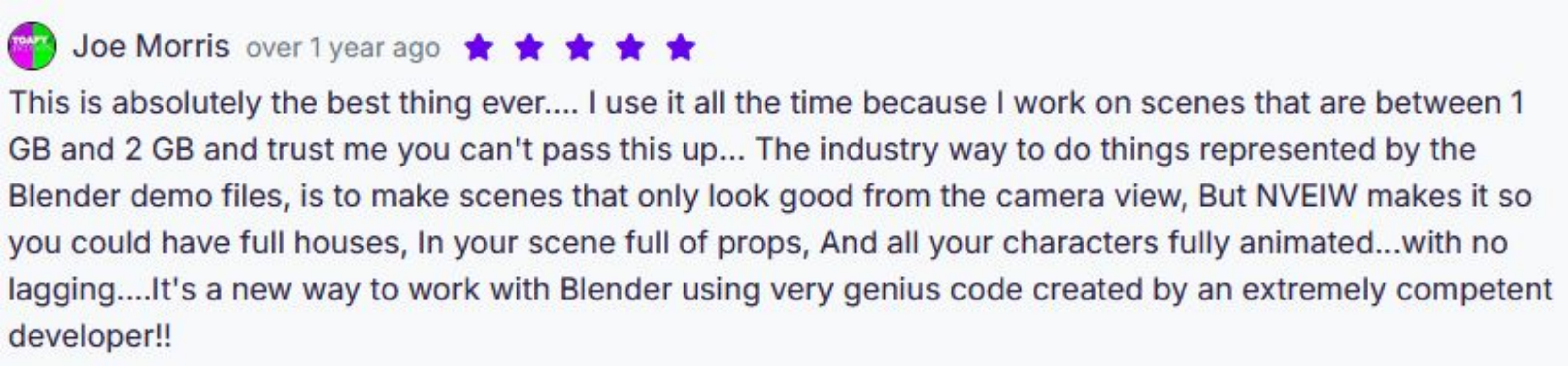
(if you maintain your tests)

5. User Satisfaction



Jan Thastum over 4 years ago ★★★★☆

A "no brainer" for the price. Does what it should. I especially like the color ranges it picks in one go. The developer fixed compatibility with 2.9 within a day after reaching out. Stellar service.



Joe Morris over 1 year ago ★★★★★

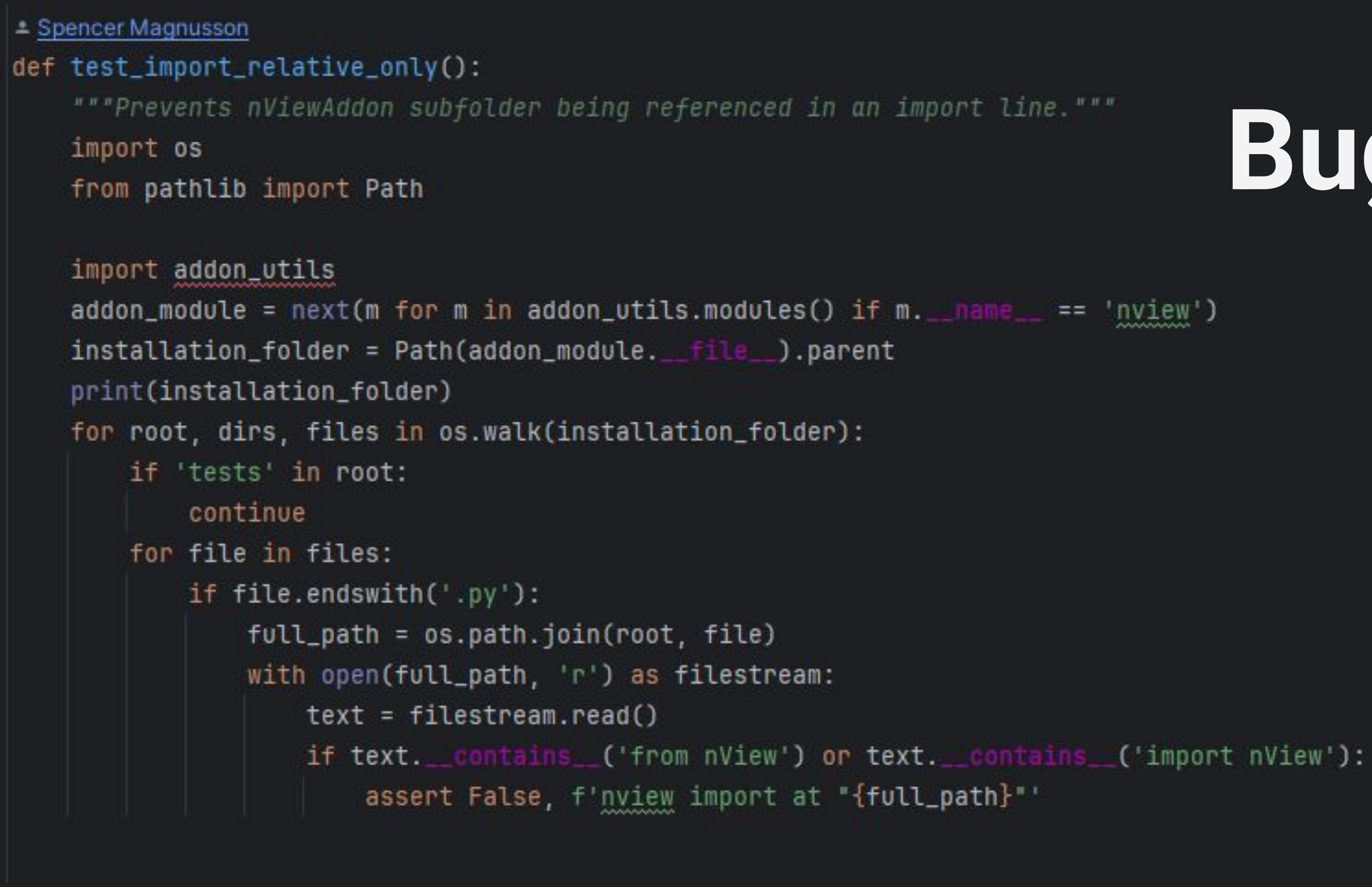
This is absolutely the best thing ever.... I use it all the time because I work on scenes that are between 1 GB and 2 GB and trust me you can't pass this up... The industry way to do things represented by the Blender demo files, is to make scenes that only look good from the camera view, But NVEIW makes it so you could have full houses, In your scene full of props, And all your characters fully animated...with no lagging....It's a new way to work with Blender using very genius code created by an extremely competent developer!!

Designing Tests

1. Installation (non-extensions and extensions)
2. Accessing properties
3. Operators
4. Non-Blender packages (executables, libraries)
5. Don’t forget uninstallation!

Other Types of Tests

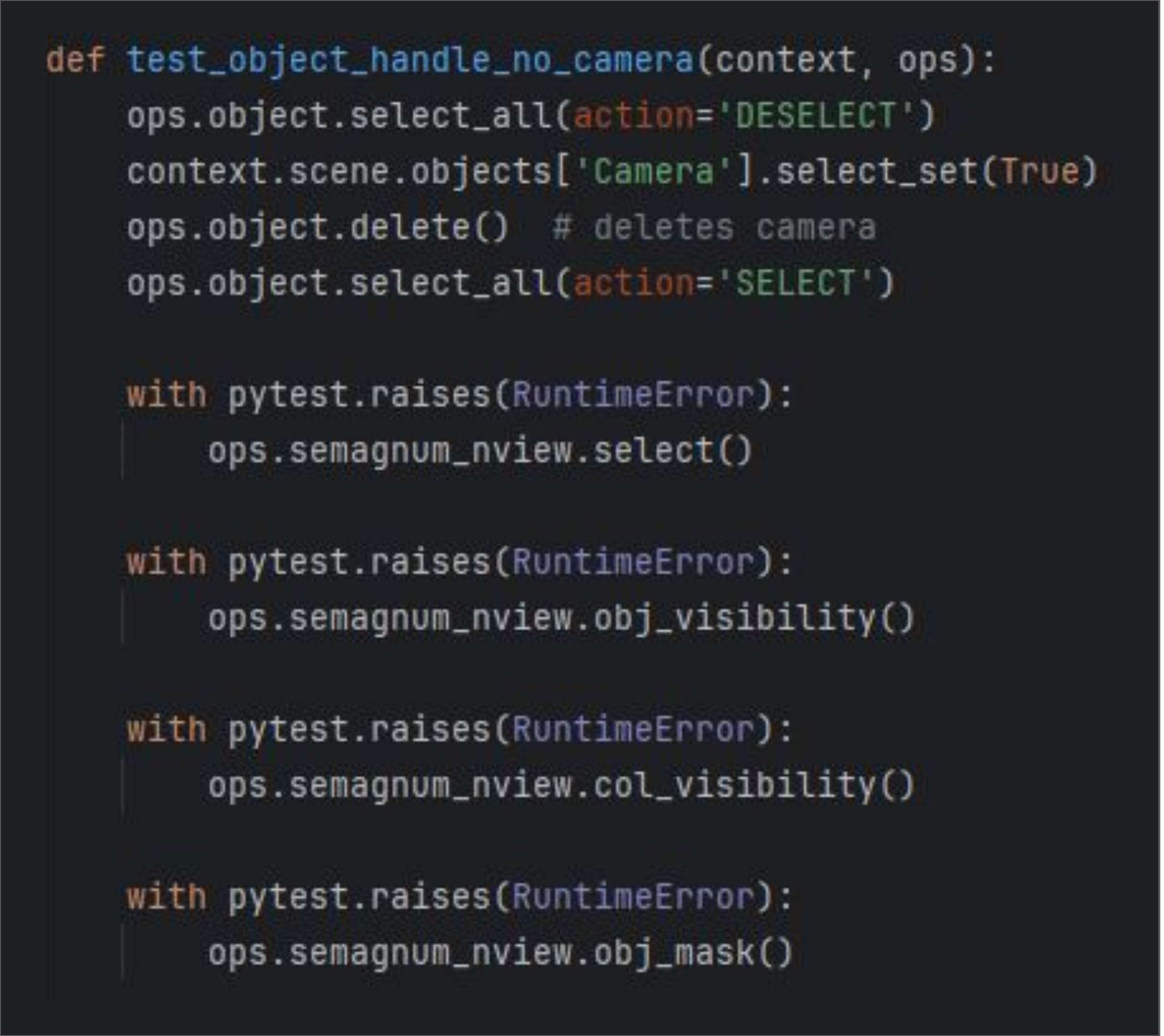
Bugs



Spencer Magnusson

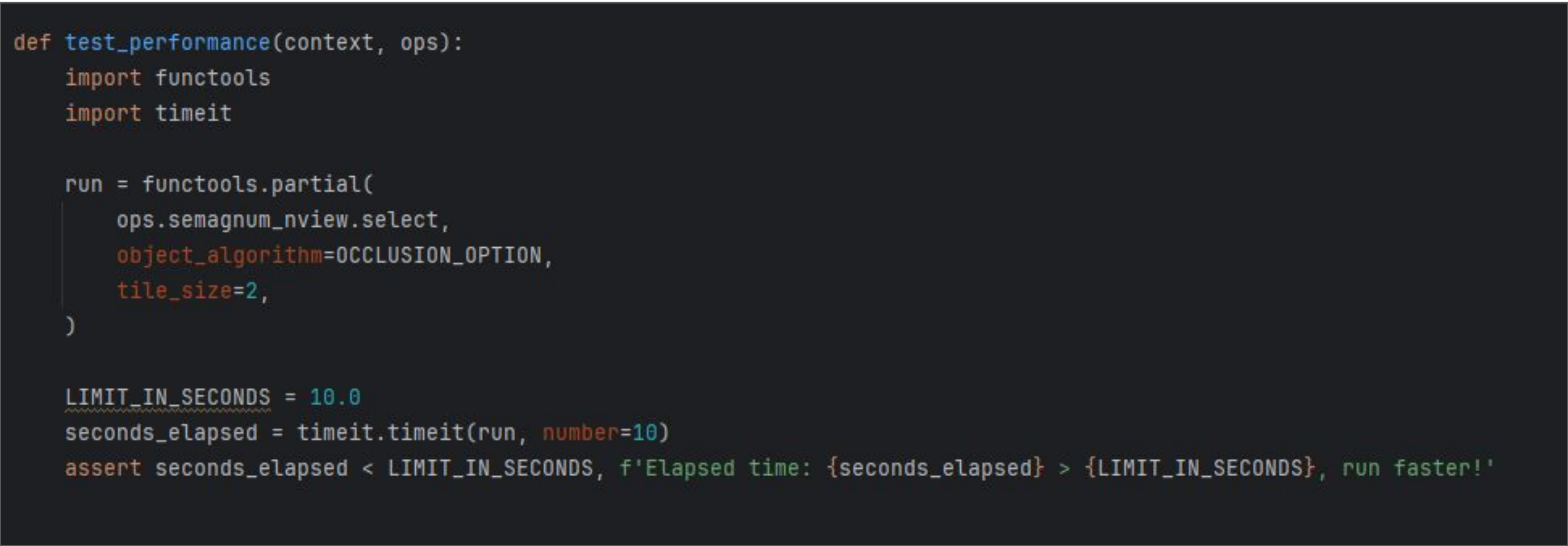
```python
def test_import_relative_only():
    """Prevents nViewAddon subfolder being referenced in an import line."""
    import os
    from pathlib import Path

    import addon_utils
    addon_module = next(m for m in addon_utils.modules() if m.__name__ == 'nview')
    installation_folder = Path(addon_module.__file__).parent
    print(installation_folder)
    for root, dirs, files in os.walk(installation_folder):
        if 'tests' in root:
            continue
        for file in files:
            if file.endswith('.py'):
                full_path = os.path.join(root, file)
                with open(full_path, 'r') as filestream:
                    text = filestream.read()
                    if text.__contains__('from nView') or text.__contains__('import nView'):
                        assert False, f'nview import at "{full_path}"'
```

Input validation



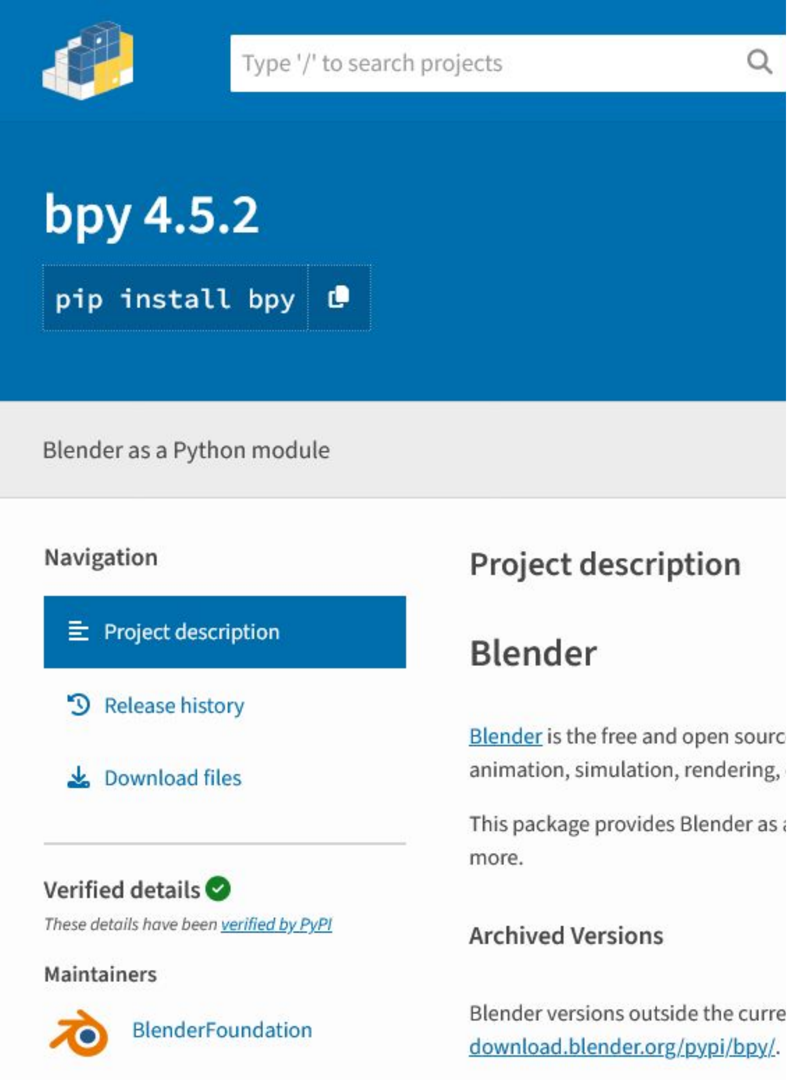
```
def test_object_handle_no_camera(context, ops):
    ops.object.select_all(action='DESELECT')
    context.scene.objects['Camera'].select_set(True)
    ops.object.delete()  # deletes camera
    ops.object.select_all(action='SELECT')

    with pytest.raises(RuntimeError):
        ops.semagnum_nview.select()

    with pytest.raises(RuntimeError):
        ops.semagnum_nview.obj_visibility()

    with pytest.raises(RuntimeError):
        ops.semagnum_nview.col_visibility()

    with pytest.raises(RuntimeError):
        ops.semagnum_nview.obj_mask()
```

Load/Performance Testing



```python
def test_performance(context, ops):
    import functools
    import timeit

    run = functools.partial(
        ops.semagnum_nview.select,
        object_algorithm=OCCLUSION_OPTION,
        tile_size=2,
    )

    LIMIT_IN_SECONDS = 10.0
    seconds_elapsed = timeit.timeit(run, number=10)
    assert seconds_elapsed < LIMIT_IN_SECONDS, f'Elapsed time: {seconds_elapsed} > {LIMIT_IN_SECONDS}, run faster!'
```

Testing with Blender



bpy 4.5.2

```
pip install bpy
```

Blender as a Python module

Verified details

These details have been verified by PyPI

Maintainers

BlenderFoundation

Project description

Blender

Blender is the free and open sourc
animation, simulation, rendering,

This package provides Blender as a
more.

Archived Versions

Blender versions outside the curre
download.blender.org/pypi/bpy/.

Blender Python Module

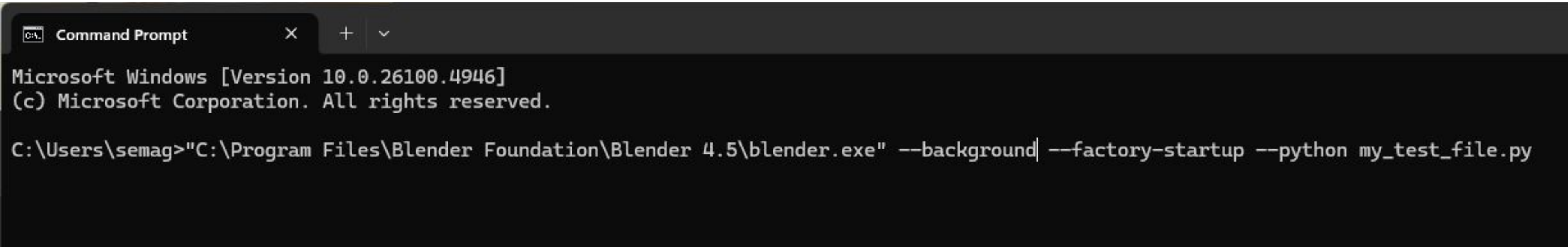
Options:

- Build from source code (using `make bpy`)
- Install with `pip`
- Download from pypi.org

Blender executable

Using the command line:

- `--background` (no UI)
- `--python <python_file>`
- `--factory-startup` (optional)

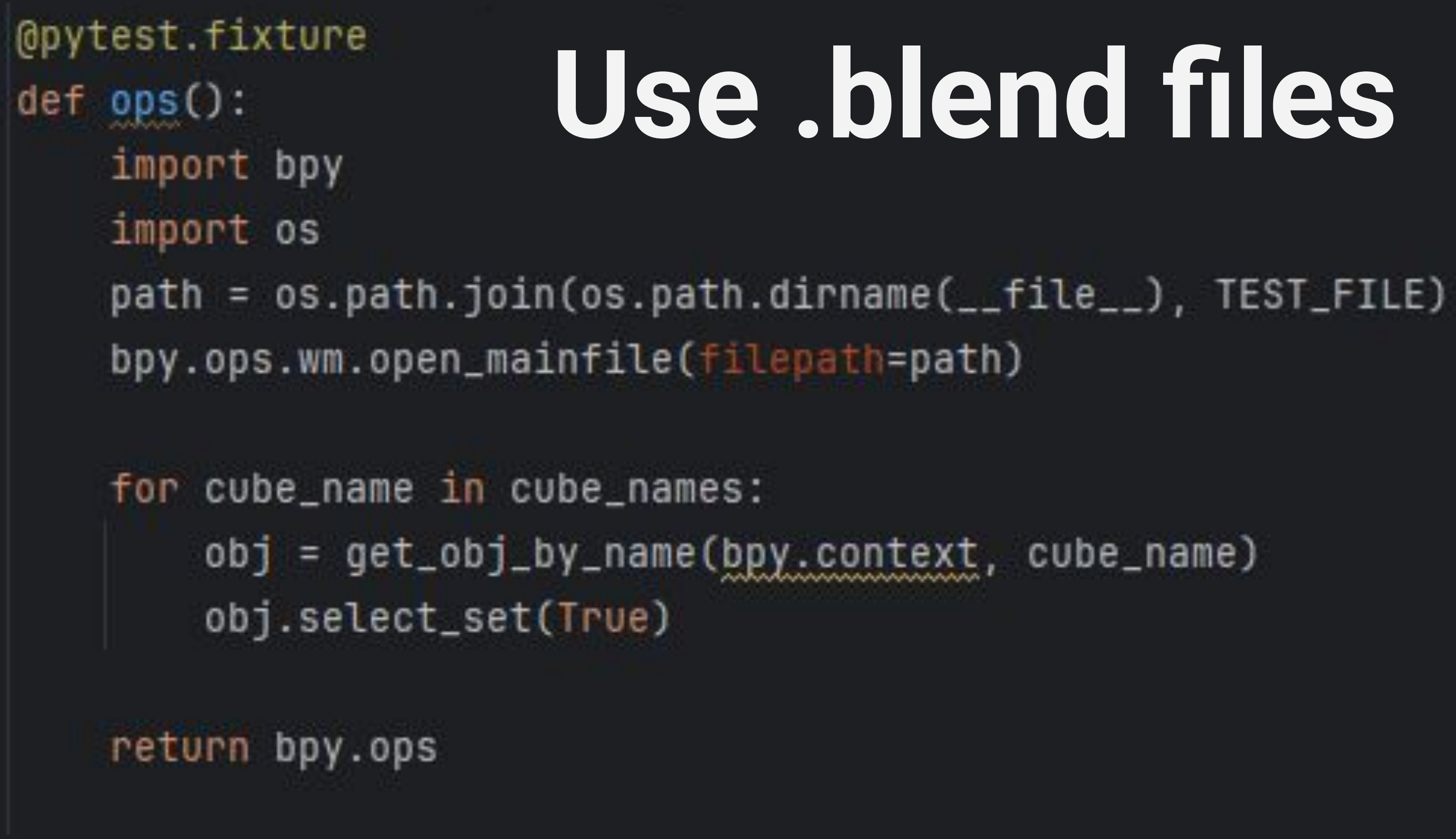


```
Microsoft Windows [Version 10.0.26100.4946]
(c) Microsoft Corporation. All rights reserved.

C:\Users\semag>"C:\Program Files\Blender Foundation\Blender 4.5\blender.exe" --background --factory-startup --python my_test_file.py
```

Testing Frameworks

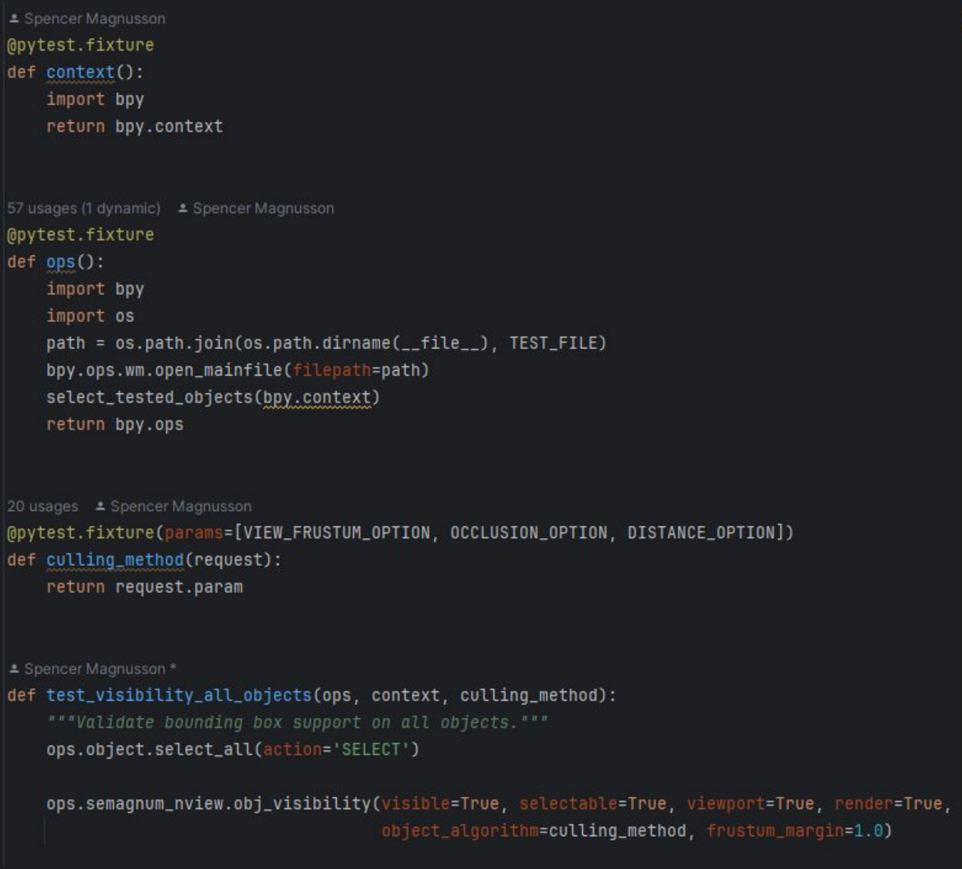
Improve Your Testing Workflow

Use .blend files



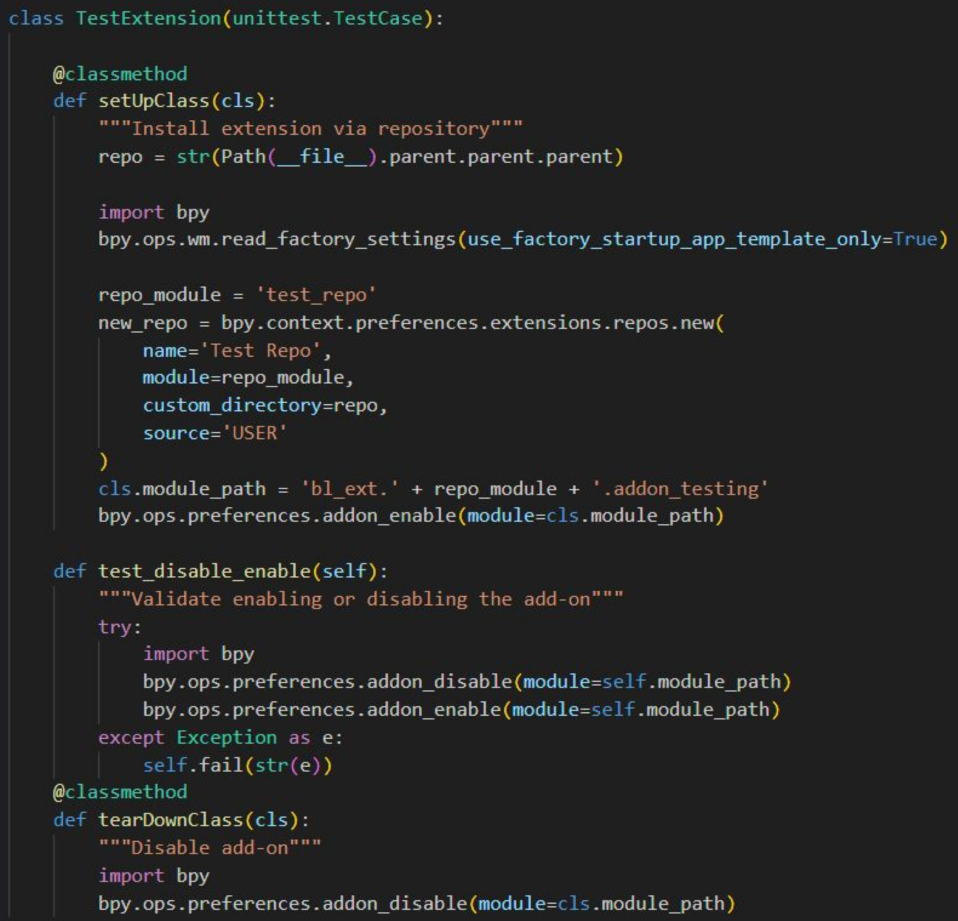
```python
@pytest.fixture
def ops():
    import bpy
    import os
    path = os.path.join(os.path.dirname(__file__), TEST_FILE)
    bpy.ops.wm.open_mainfile(filepath=path)

    for cube_name in cube_names:
        obj = get_obj_by_name(bpy.context, cube_name)
        obj.select_set(True)

    return bpy.ops
```

Fixtures



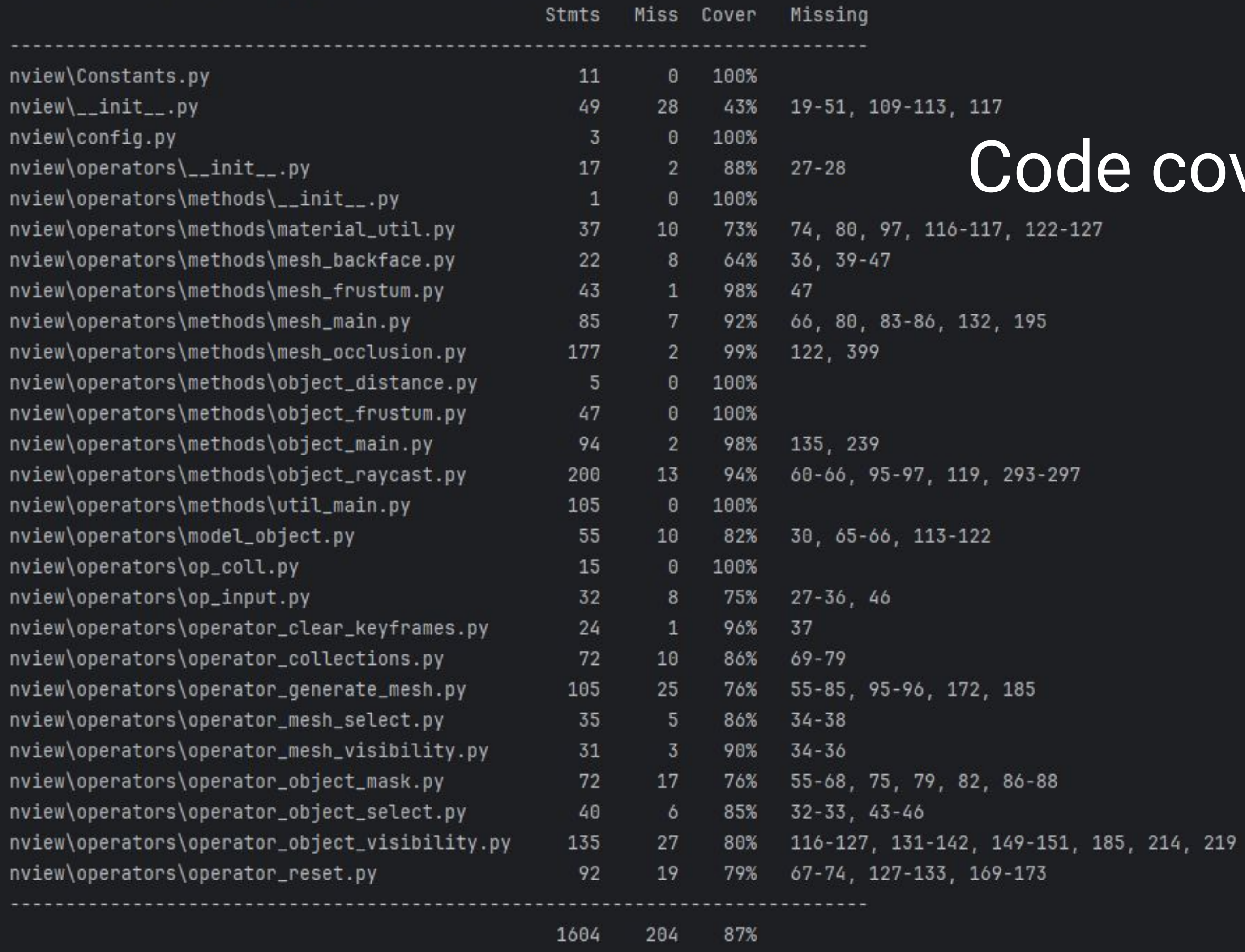
```
Spencer Magnusson
@pytest.fixture
def context():
    import bpy
    return bpy.context


57 usages (1 dynamic)   Spencer Magnusson
@pytest.fixture
def ops():
    import bpy
    import os
    path = os.path.join(os.path.dirname(__file__), TEST_FILE)
    bpy.ops.wm.open_mainfile(filepath=path)
    select_tested_objects(bpy.context)
    return bpy.ops


20 usages   Spencer Magnusson
@pytest.fixture(params=[VIEW_FRUSTUM_OPTION, OCCLUSION_OPTION, DISTANCE_OPTION])
def culling_method(request):
    return request.param


Spencer Magnusson *
def test_visibility_all_objects(ops, context, culling_method):
    """Validate bounding box support on all objects."""
    ops.object.select_all(action='SELECT')

    ops.semagnum_nview.obj_visibility(visible=True, selectable=True, viewport=True, render=True,
                                      object_algorithm=culling_method, frustum_margin=1.0)
```

Setup/teardown functions

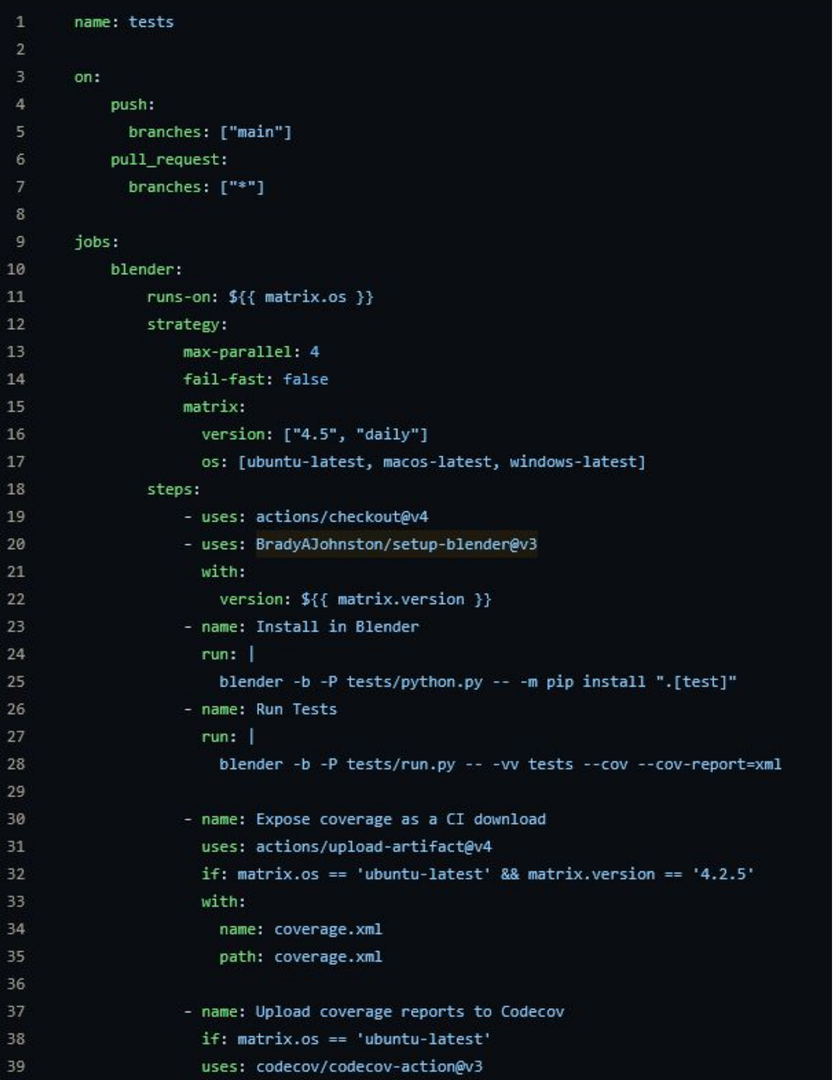


```
class TestExtension(unittest.TestCase):

    @classmethod
    def setUpClass(cls):
        """Install extension via repository"""
        repo = str(Path(__file__).parent.parent.parent)

        import bpy
        bpy.ops.wm.read_factory_settings(use_factory_startup_app_template_only=True)

        repo_module = 'test_repo'
        new_repo = bpy.context.preferences.extensions.repos.new(
            name='Test Repo',
            module=repo_module,
            custom_directory=repo,
            source='USER'
        )
        cls.module_path = 'bl_ext.' + repo_module + '.addon_testing'
        bpy.ops.preferences.addon_enable(module=cls.module_path)

    def test_disable_enable(self):
        """Validate enabling or disabling the add-on"""
        try:
            import bpy
            bpy.ops.preferences.addon_disable(module=self.module_path)
            bpy.ops.preferences.addon_enable(module=self.module_path)
        except Exception as e:
            self.fail(str(e))
    @classmethod
    def tearDownClass(cls):
        """Disable add-on"""
        import bpy
        bpy.ops.preferences.addon_disable(module=cls.module_path)
```

Code coverage



```
                                                  Stmts   Miss  Cover   Missing
----------------------------------------------------------------------------
nview\Constants.py                                   11      0   100%
nview\__init__.py                                    49     28    43%   19-51, 109-113, 117
nview\config.py                                       3      0   100%
nview\operators\__init__.py                          17      2    88%   27-28
nview\operators\methods\__init__.py                   1      0   100%
nview\operators\methods\material_util.py             37     10    73%   74, 80, 97, 116-117, 122-127
nview\operators\methods\mesh_backface.py             22      8    64%   36, 39-47
nview\operators\methods\mesh_frustum.py              43      1    98%   47
nview\operators\methods\mesh_main.py                 85      7    92%   66, 80, 83-86, 132, 195
nview\operators\methods\mesh_occlusion.py           177      2    99%   122, 399
nview\operators\methods\object_distance.py            5      0   100%
nview\operators\methods\object_frustum.py            47      0   100%
nview\operators\methods\object_main.py               94      2    98%   135, 239
nview\operators\methods\object_raycast.py           200     13    94%   60-66, 95-97, 119, 293-297
nview\operators\methods\util_main.py                105      0   100%
nview\operators\model_object.py                      55     10    82%   30, 65-66, 113-122
nview\operators\op_coll.py                           15      0   100%
nview\operators\op_input.py                          32      8    75%   27-36, 46
nview\operators\operator_clear_keyframes.py          24      1    96%   37
nview\operators\operator_collections.py              72     10    86%   69-79
nview\operators\operator_generate_mesh.py           105     25    76%   55-85, 95-96, 172, 185
nview\operators\operator_mesh_select.py              35      5    86%   34-38
nview\operators\operator_mesh_visibility.py          31      3    90%   34-36
nview\operators\operator_object_mask.py              72     17    76%   55-68, 75, 79, 82, 86-88
nview\operators\operator_object_select.py            40      6    85%   32-33, 43-46
nview\operators\operator_object_visibility.py       135     27    80%   116-127, 131-142, 149-151, 185, 214, 219
nview\operators\operator_reset.py                    92     19    79%   67-74, 127-133, 169-173
----------------------------------------------------------------------------
                                                   1604    204    87%
```



```yaml
name: tests

on:
    push:
      branches: ["main"]
    pull_request:
      branches: ["*"]

jobs:
    blender:
        runs-on: ${{ matrix.os }}
        strategy:
            max-parallel: 4
            fail-fast: false
            matrix:
              version: ["4.5", "daily"]
              os: [ubuntu-latest, macos-latest, windows-latest]
        steps:
            - uses: actions/checkout@v4
            - uses: BradyAJohnston/setup-blender@v3
              with:
                version: ${{ matrix.version }}
            - name: Install in Blender
              run: |
                blender -b -P tests/python.py -- -m pip install ".[test]"
            - name: Run Tests
              run: |
                blender -b -P tests/run.py -- -vv tests --cov --cov-report=xml

            - name: Expose coverage as a CI download
              uses: actions/upload-artifact@v4
              if: matrix.os == 'ubuntu-latest' && matrix.version == '4.2.5'
              with:
                name: coverage.xml
                path: coverage.xml

            - name: Upload coverage reports to Codecov
              if: matrix.os == 'ubuntu-latest'
              uses: codecov/codecov-action@v3
```

Automate your automation

"Extensions are Coming: Rethinking add-on development by unlocking Python's full potential" - Friday, 14:30

What about testing Blender with the UI?

Event Simulation

Command line: `--enable-event-simulate`

bpy.types.Window.event_simulate(type, value, x=0, y=0, ...)

Demo/Example with Light Painter

Hey, hey Spencer…

Blender has something to say…

Resources

- Add-on Testing Template
  - Codeberg
  - GitHub
  - Light Painter
- "Scripting Your First Blender Add-on" course for add-on beginners