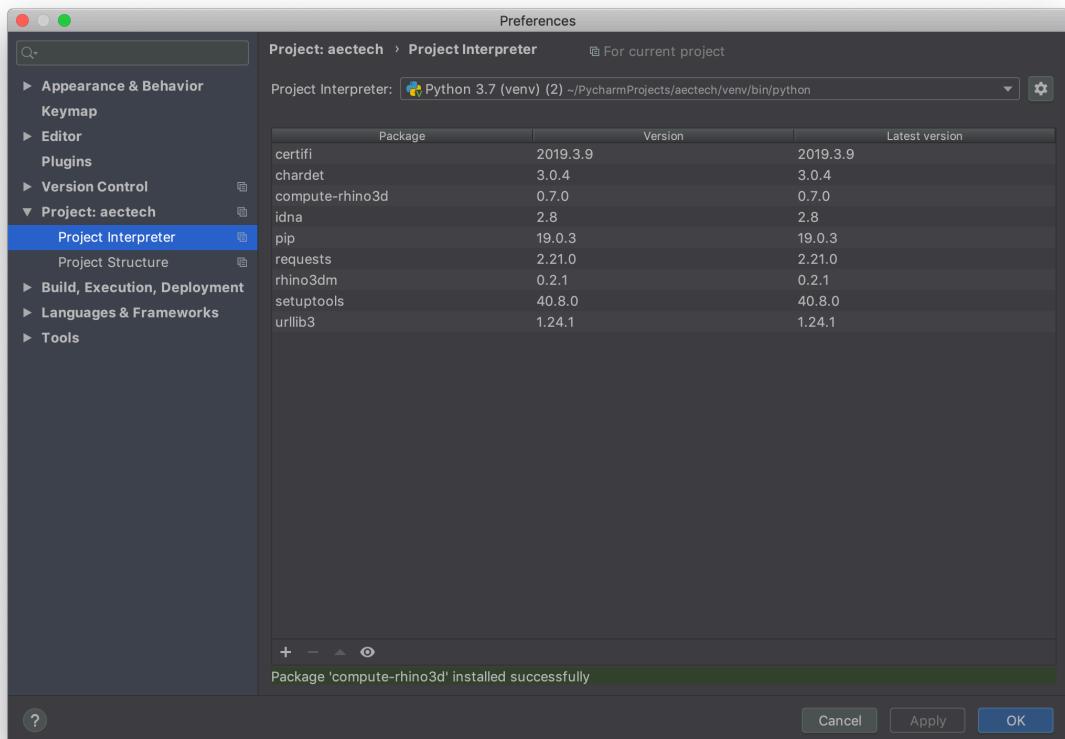


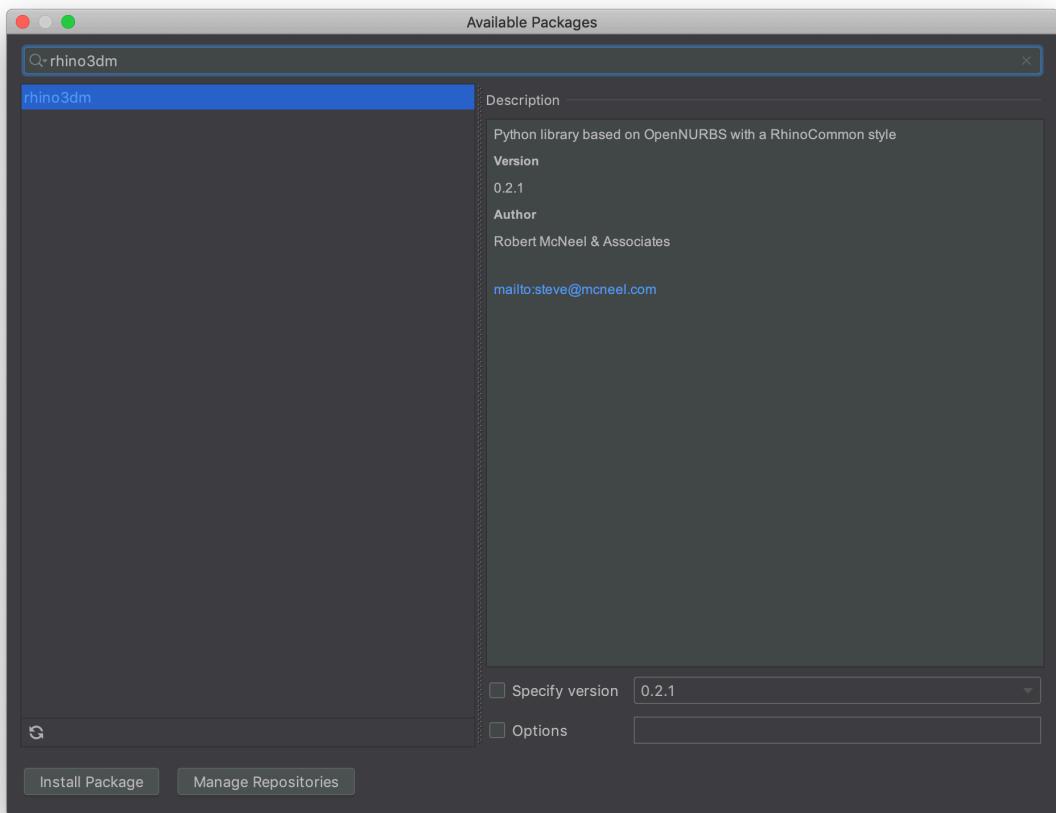
AEC Tech Seattle

Compute workshop

Screenshots!

1. Creating a new project in PyCharm
 1. Settings
 2. Ensure set to a official python distribution (bugs with VS python)
2. Settings > Project: xxx > Project interpreter
 1. Install rhino3dm package





3. Open Rhino (min. v5)

1. Create a curve (~4 control points) and a circle (intersecting)
2. Get locations of curve control points using `_EvaluatePt`
3. `EvaluatePt`
4. PyCharm... Select project in tree and create new file
5. Write some code!
6. Use developer.rhino3d.com/api for help!

The screenshot shows a web browser window displaying the [rhino3dm documentation](https://mcneel.github.io/rhino3dm/). The left sidebar contains a navigation menu with various class names: Arc, ArcCurve, ArchivableDictionary, BezierCurve, Bitmap, BoundingBox, Box, Brep, BrepFace, BrepFaceList, Circle, CommonObject, ComponentIndex, Cone, ConstructionPlane, and Curve. The main content area features a heading "Welcome to rhino3dm's documentation!" and a "Contents:" section with a detailed list of the same class names.

7. Add a break point on the first line

The screenshot shows the PyCharm IDE interface. A Python script named `make3dmfile.py` is open in the editor. A red circular break point is set on the fourth line of code. The code defines a list of points and creates a curve from them. The PyCharm debugger tool window is visible at the bottom, showing the current stack frame and variables. The variable `points` is expanded to show its contents: an empty list.

```
import rhino3dm
points = []
points.append(rhino3dm.Point3d(-9.75, -4.361, 0))
points.append(rhino3dm.Point3d(8.667, -2.235, 0))
points.append(rhino3dm.Point3d(-5.396, 6.923, 0))
points.append(rhino3dm.Point3d(10.847, 10.302, 0))
curve1 = rhino3dm.Curve.CreateControlPointCurve(points, 3)

model = rhino3dm.File3dm()
model.Objects.AddCurve(curve1)
model.Write('my_first_curve.3dm')
```

8. Step over

The screenshot shows the PyCharm IDE interface. The top window displays the code for `make3dmfile.py`:

```

aectech [-~/PycharmProjects/aectech] - .../make3dmfile.py [aectech]
make3dmfile.py _pydev_execfile.py
1 import rhino3dm
2
3 points = []
4     points: <class 'list'>: [<rhino3dm._rhino3dm.Point3d object at 0x102aa2dc0>]
5 points.append(rhino3dm.Point3d(-9.757, -4.361, 0))
6 points.append(rhino3dm.Point3d(3.667, -2.235, 0))
7 points.append(rhino3dm.Point3d(-5.396, 6.923, 0))
8 points.append(rhino3dm.Point3d(10.847, 10.302, 0))
9 curve1 = rhino3dm.Curve.CreateControlPointCurve(points, 3)
10
11 model = rhino3dm.File3dm()
12 model.Objects.AddCurve(curve1)
model.Write('my_first_curve.3dm')

```

The bottom window shows the debugger's variable view for the variable `points`:

```

Variables
points = (list) <class 'list'>: [<rhino3dm._rhino3dm.Point3d object at 0x102aa2dc0>]
    0 = (Point3d) -9.757,-4.361,0
    len_ = (int) 1

```

9. Run to the end

10. Right-click on the .3dm file that shows up on the left and open in Explorer/Finder

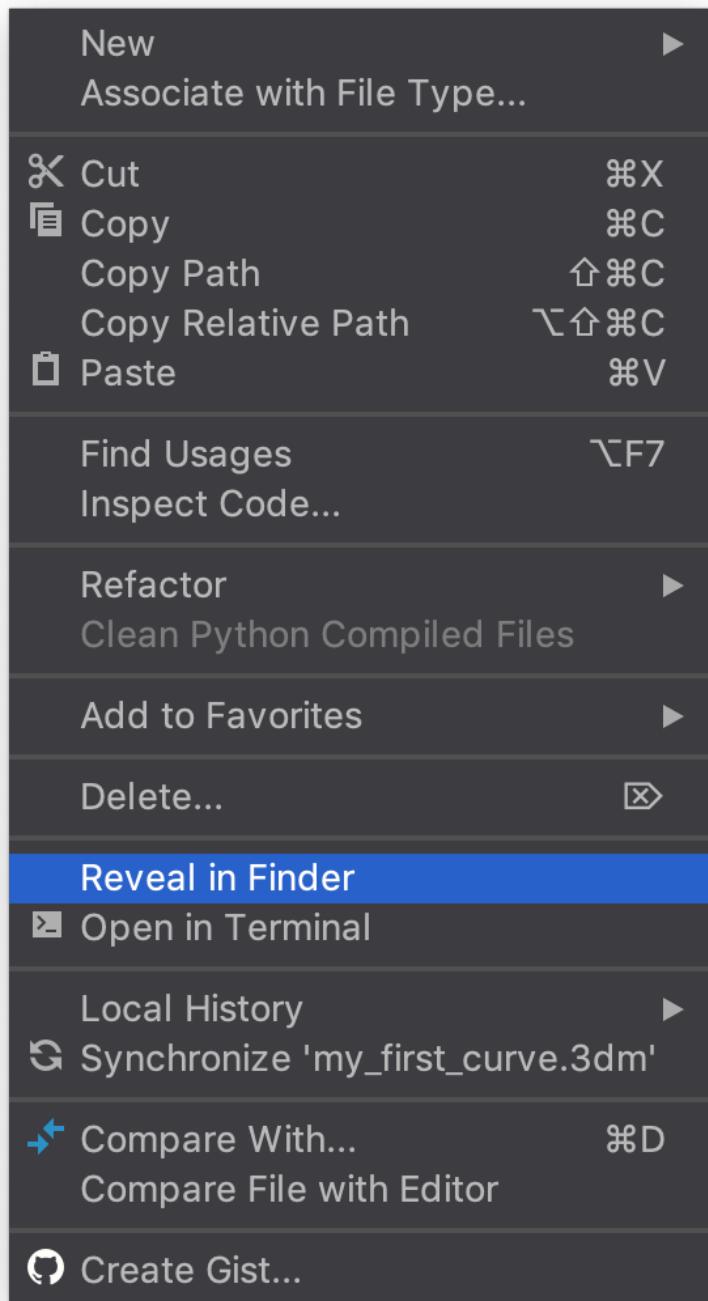
The screenshot shows the PyCharm IDE interface. The top window displays the code for `make3dmfile.py`:

```

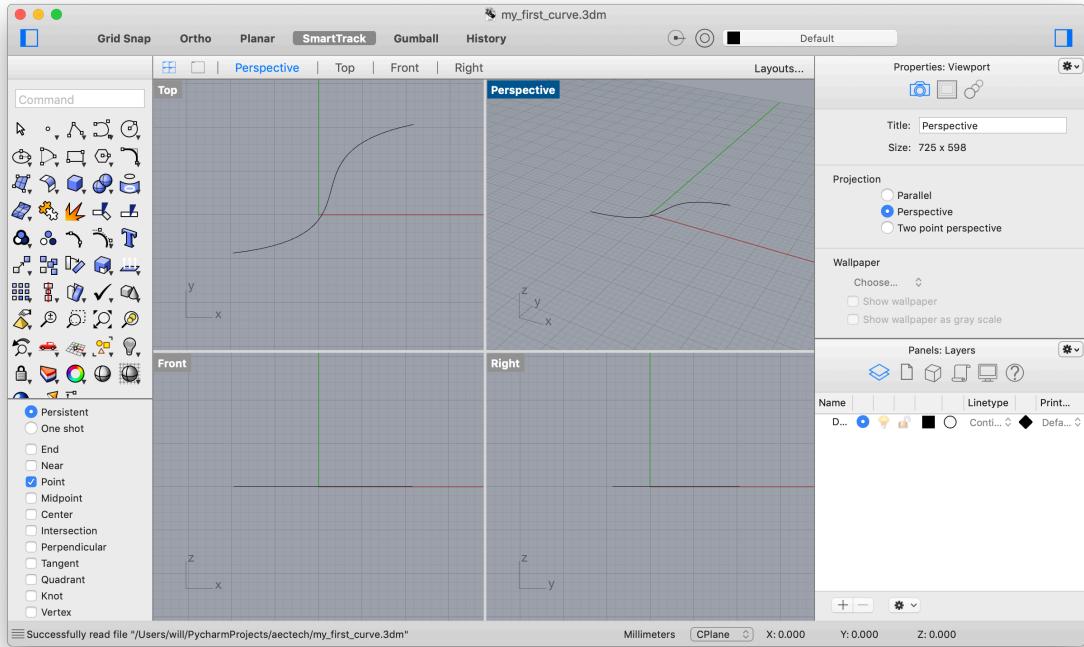
aectech [-~/PycharmProjects/aectech] - .../make3dmfile.py [aectech]
make3dmfile.py _pydev_execfile.py
1 import rhino3dm
2
3 points = []
4     points: <class 'list'>: [<rhino3dm._rhino3dm.Point3d object at 0x102aa2dc0>]
5 points.append(rhino3dm.Point3d(-9.757, -4.361, 0))
6 points.append(rhino3dm.Point3d(3.667, -2.235, 0))
7 points.append(rhino3dm.Point3d(-5.396, 6.923, 0))
8 points.append(rhino3dm.Point3d(10.847, 10.302, 0))
9 curve1 = rhino3dm.Curve.CreateControlPointCurve(points, 3)
10
11 model = rhino3dm.File3dm()
12 model.Objects.AddCurve(curve1)
model.Write('my_first_curve.3dm')

```

The bottom window shows the debugger's variable view for the variable `points`.



11. Open file in Rhino



12. Gotcha: for Rhino 5 users, use `model.Write('my_first_curve.3dm', 5)
13. Go back into Rhino and draw a Circle e.g. centre at {0,0}, radius 5
14. Write some code!

```

aectech [-~/PycharmProjects/aectech] - .../make3dmfile.py [aectech]
make3dmfile.py

import rhino3dm

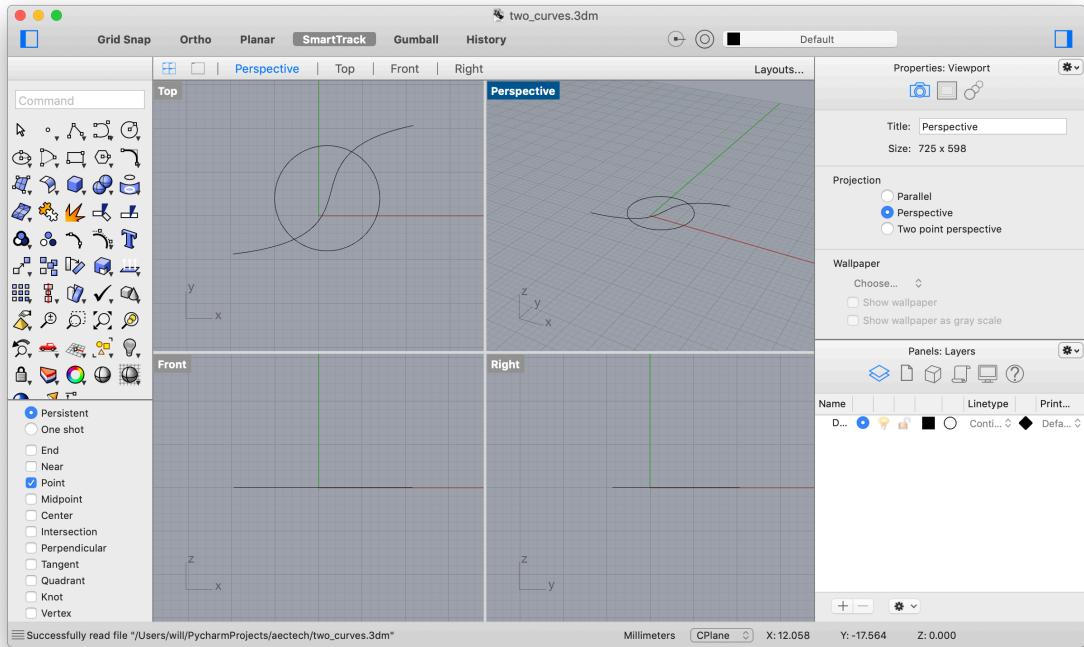
points = []
points.append(rhino3dm.Point3d(-9.757, -4.361, 0))
points.append(rhino3dm.Point3d(8.667, -2.235, 0))
points.append(rhino3dm.Point3d(-5.396, 6.923, 0))
points.append(rhino3dm.Point3d(10.847, 10.302, 0))
curve1 = rhino3dm.Curve.CreateControlPointCurve(points, 3)

circle = rhino3dm.Circle(rhino3dm.Point3d(1, 2, 0), 6)
curve2 = circle.ToNurbsCurve()

model = rhino3dm.File3dm()
model.Objects.AddCurve(curve1)
model.Objects.AddCurve(curve2)
model.Write('two_curves.3dm')

```

15. Open file in Rhino



16. Install compute.rhino3d package
17. Get a Rhino Accounts auth token (with "compute" scope) from rhino3d.com/compute/login
18. Write some more code!

Welcome to compute.rhino3d.js's documentation!

Contents:

- [RhinoCompute.AreaMassProperties](#)
- [RhinoCompute.BezierCurve](#)
- [RhinoCompute.Brep](#)
- [RhinoCompute.BrepFace](#)
- [RhinoCompute.Curve](#)
- [RhinoCompute.Extrusion](#)
- [RhinoCompute.Intersection](#)
- [RhinoCompute.Mesh](#)
- [RhinoCompute.NurbsCurve](#)
- [RhinoCompute.NurbsSurface](#)
- [RhinoCompute.Surface](#)
- [RhinoCompute.VolumeMassProperties](#)

Indices and tables

- [Index](#)
- [Module Index](#)
- [Search Page](#)

```

aechtech [-~/PycharmProjects/aechtech] - .../make3dmfile.py [aechtech]
make3dmfile.py
1 import rhino3dm
2 import compute_rhino3d.Util
3 import compute_rhino3d.Intersection
4
5 compute_rhino3d.Util.authToken = 'TOKEN_HERE'
6
7 points = []
8 points.append(rhino3dm.Point3d(-9.757, -4.361, 0))
9 points.append(rhino3dm.Point3d(8.667, -2.235, 0))
10 points.append(rhino3dm.Point3d(-5.396, 6.923, 0))
11 points.append(rhino3dm.Point3d(10.847, 10.302, 0))
12 curve1 = rhino3dm.Curve.CreateControlPointCurve(points, 3)
13
14 circle = rhino3dm.Circle(rhino3dm.Point3d(1, 2, 0), 6)
15
16 curve2 = circle.ToNurbsCurve()
17
18 model = rhino3dm.File3dm()
19 model.Objects.AddCurve(curve1)
20 model.Objects.AddCurve(curve2)
21
22 intersections = compute_rhino3d.Intersection.CurveCurve(curve1, curve2, 0.01, 0.01)
23
24 model.Write('two_curves.3dm')

```

PEP 8: no newline at end of file

24:30 LF: UTF-8: 4 spaces: Python 3.7 (venv) (2)

19. Debug

```

aechtech [-~/PycharmProjects/aechtech] - .../make3dmfile.py [aechtech]
make3dmfile.py
15 curvez = circle.ToNurbsCurve() curvez: <rhino3dm._rhino3dm.NurbsCurve object at 0x10be6b1b0>
16
17 model = rhino3dm.File3dm() model: <rhino3dm._rhino3dm.File3dm object at 0x10be6b1b8>
18 model.Objects.AddCurve(curve1)
19 model.Objects.AddCurve(curve2)
20
21 intersections = compute_rhino3d.Intersection.CurveCurve(curve1, curve2, 0.01, 0.01) intersections: <class 'list'>: [{"IsPoint": true, "IsOverlap": false, "PointA": {"X": -2.7564721321663814, "Y": -2.678559299641008, "Z": 0.0}, "PointB": {"X": -2.7564721321663814, "Y": -2.678559299641008, "Z": 0.0}, "IsPoint": false, "IsOverlap": false}, {"IsPoint": false, "IsOverlap": false, "PointA": {"X": -2.7564721321663814, "Y": -2.678559299641008, "Z": 0.0}, "PointB": {"X": -2.7564721321663814, "Y": -2.678559299641008, "Z": 0.0}, "IsPoint": false, "IsOverlap": false}]
22
23 model.Write('intersection_2d.3dm')
24

```

Debug: make3dmfile

Main Thread

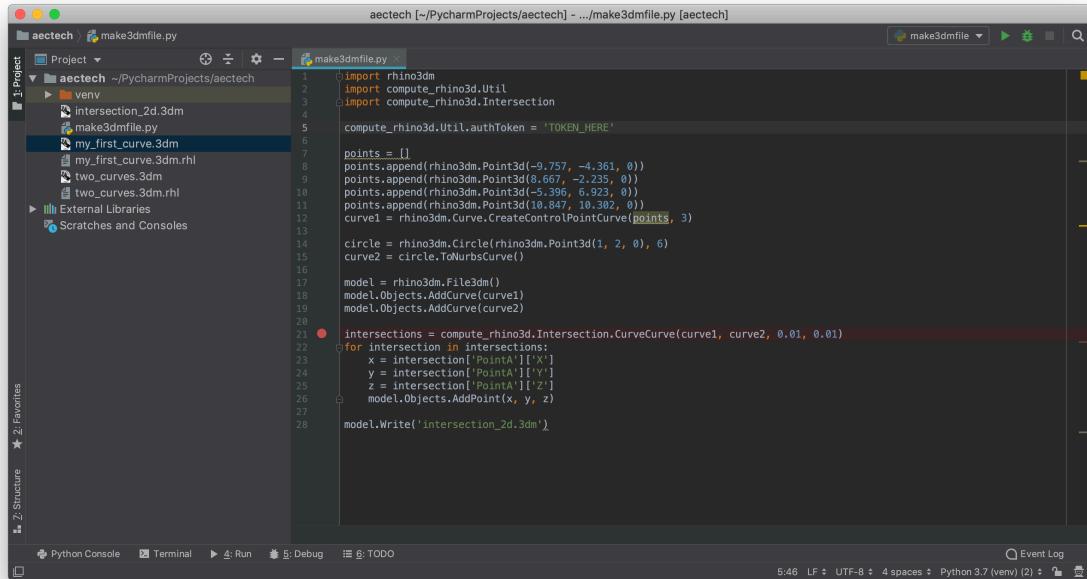
Variables

0 = {"IsPoint": true, "IsOverlap": false, "PointA": {"X": -2.7564721321663814, "Y": -2.678559299641008, "Z": 0.0}, "PointB": {"X": -2.7564721321663814, "Y": -2.678559299641008, "Z": 0.0}, "IsPoint": false, "IsOverlap": false}

1 = {"IsPoint": false, "IsOverlap": false, "PointA": {"X": -2.7564721321663814, "Y": -2.678559299641008, "Z": 0.0}, "PointB": {"X": -2.7564721321663814, "Y": -2.678559299641008, "Z": 0.0}, "IsPoint": false, "IsOverlap": false}

24:1 LF: UTF-8: 4 spaces: Python 3.7 (venv) (2)

18. More code



```

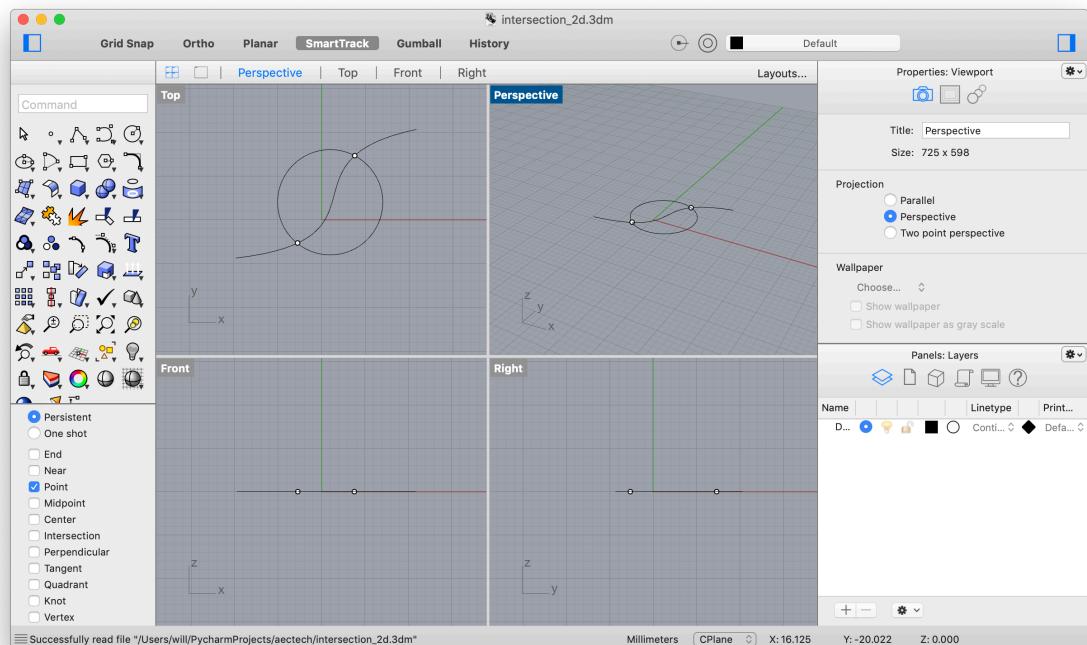
aechtech [~/PycharmProjects/aechtech] - .../make3dmfile.py [aechtech]
  make3dmfile.py
  Project 1: aechtech ~/PycharmProjects/aechtech
    my_first_curve.3dm
    my_first_curve.3dm.rhl
    two_curves.3dm
    two_curves.3dm.rhl
  External Libraries
  Scratches and Consoles

  1 import rhino3dm
  2 import compute_rhino3d.Util
  3 import compute_rhino3d.Intersection
  4
  5 compute_rhino3d.Util.authToken = 'TOKEN_HERE'
  6
  7 points = []
  8 points.append(rhino3dm.Point3d(-9.75, -4.361, 0))
  9 points.append(rhino3dm.Point3d(8.667, -2.235, 0))
  10 points.append(rhino3dm.Point3d(-5.396, 6.923, 0))
  11 points.append(rhino3dm.Point3d(10.847, 10.302, 0))
  12 curve1 = rhino3dm.Curve.CreateControlPointCurve(points, 3)
  13
  14 circle = rhino3dm.Circle(rhino3dm.Point3d(1, 2, 0), 6)
  15 curve2 = circle.ToNurbsCurve()
  16
  17 model = rhino3dm.File3dm()
  18 model.Objects.AddCurve(curve1)
  19 model.Objects.AddCurve(curve2)
  20
  21 intersections = compute_rhino3d.Intersection.CurveCurve(curve1, curve2, 0.01, 0.01)
  22 for intersection in intersections:
  23     x = intersection['PointA']['X']
  24     y = intersection['PointA']['Y']
  25     z = intersection['PointA']['Z']
  26     model.Objects.AddPoint(x, y, z)
  27
  28 model.Write('intersection_2d.3dm')

```

The screenshot shows the PyCharm IDE interface with a Python script named 'make3dmfile.py'. The script uses the 'rhino3dm' and 'compute_rhino3d' libraries to create a 3D model from two curves. It defines points, creates a circle, converts it to a Nurbs curve, adds curves to a model, and finds intersections between them. The resulting file is 'intersection_2d.3dm'.

19. Rhino!



20. SVG

A screenshot of a Mac OS X desktop showing a web browser window for the PyPI project page of the `svgwrite` library. The URL in the address bar is `pypi.org`. The page content includes:

- Abstract**: A Python library to create SVG drawings.
- Project links**: Includes [Homepage](#) and [Download](#).
- Statistics**: View statistics for this project via [Libraries.io](#), or by using [Google BigQuery](#).
- Meta**: License: MIT License (MIT License), Author: [Manfred Moitzi](#).
- Maintainers**: [!\[\]\(00b8f49f43788bd3c06dc4fe2b9a6269_img.jpg\) mozman](#).

The right side of the page contains code examples and installation instructions:

```
import svgwrite
dwg = svgwrite.Drawing('test.svg', profile='tiny')
dwg.add(dwg.line((0, 0), (10, 0), stroke=svgwrite.rgb(10, 10, 16, '%')))
dwg.add(dwg.text('Test', insert=(0, 0.2), fill='red'))
dwg.save()
```

for more examples see: [examples.py](#)

Installation

with pip:

```
pip install svgwrite
```

or from source:

```
python setup.py install
```

A screenshot of a Mac OS X desktop showing a web browser window for the PyPI project page of the `svgwrite` library. The URL in the address bar is `pypi.org`. The page content includes:

Search projects and **Help**, **Donate**, **Log in**, **Register** buttons.

svgwrite 1.2.1

[pip install svgwrite](#) 

Latest version  Last released: Nov 15, 2018

A Python library to create SVG drawings.

Navigation: Project description, Release history, Download files.

Project description: **svgwrite** docs passing

Abstract: A Python library to create SVG drawings.

```

aectech [-~/PycharmProjects/aectech] - .../writesvg.py [aectech]
 1 import rhino3dm
 2 import svgwrite
 3
 4 points = []
 5 points.append(rhino3dm.Point3d(-9.757, -4.361, 0))
 6 points.append(rhino3dm.Point3d(8.667, -2.235, 0))
 7 points.append(rhino3dm.Point3d(-5.396, 6.923, 0))
 8 points.append(rhino3dm.Point3d(10.847, 10.302, 0))
 9 curve1 = rhino3dm.Curve.CreateControlPointCurve(points, 3)
10
11 circle = rhino3dm.Circle(rhino3dm.Point3d(1, 2, 0), 6)
12 curve2 = circle.ToNurbsCurve()
13
14 dwg = svgwrite.Drawing('test.svg', profile='tiny')
15
16 domain = curve1.Domain
17 for i in range(100):
18     t0 = domain.T0 + (domain.T1 - domain.T0) * (i / 100.0)
19     t1 = domain.T0 + (domain.T1 - domain.T0) * ((i+1) / 100.0)
20     pt0 = curve1.PointAt(t0)
21     pt1 = curve1.PointAt(t1)
22     s = 30
23     dwg.add(dwg.line((pt0.X * s, pt0.Y * s), (pt1.X * s, pt1.Y * s), stroke=svgwrite.rgb(10, 10, 16, '%')))
24
25 dwg.save()

```

Project pane shows files: aectech, venv, intersection_2d.3dm, intersection_2d.3dm.rhl, make3dmfile.py, my_first_curve.3dm, my_first_curve.3dm.rhl, test.svg, two_curves.3dm, two_curves.3dm.rhl, writesvg.py.

Toolbars: Python Console, Terminal, Run, Debug, TODO.

Status bar: PEP 8: no newline at end of file, 25:11 LF, UTF-8, 4 spaces, Python 3.7 (venv) (2).

21. 3d intersections

```

aectech [-~/PycharmProjects/aectech] - .../intersection3d.py [aectech]
 1 import rhino3dm
 2 import compute_rhino3d.Util
 3 import compute_rhino3d.Intersection
 4
 5 compute_rhino3d.Util.authToken = 'TOKEN_HERE'
 6
 7 points = []
 8 points.append(rhino3dm.Point3d(-9.757, -4.361, -10))
 9 points.append(rhino3dm.Point3d(8.667, -2.235, 0))
10 points.append(rhino3dm.Point3d(-5.396, 6.923, 0))
11 points.append(rhino3dm.Point3d(10.847, 10.302, 10))
12 curve1 = rhino3dm.Curve.CreateControlPointCurve(points, 3)
13
14 sphere = rhino3dm.Sphere(rhino3dm.Point3d(1, 2, 0), 6)
15 brep1 = sphere.ToBrep()
16
17 model = rhino3dm.File3dm()
18 model.Objects.AddCurve(curve1)
19 model.Objects.AddBrep(brep1)
20
21 intersections = compute_rhino3d.Intersection.CurveBrep(curve1, brep1, 0.01)
22 points = intersections[2]
23 for intersection in points:
24     x = intersection['X']
25     y = intersection['Y']
26     z = intersection['Z']
27     model.Objects.AddPoint(x, y, z)
28
29 model.Write('intersection_3d.3dm')
30

```

Project pane shows files: aectech, venv, intersection3d.py, intersection_3d.3dm, intersection_3d.3dm.rhl, make3dmfile.py, my_first_curve.3dm, my_first_curve.3dm.rhl, test.svg, two_curves.3dm, two_curves.3dm.rhl, writesvg.py.

Toolbars: Python Console, Terminal, Run, Debug, TODO.

Status bar: PEP 8: line too long (531 > 120 characters), 5:45 LF, UTF-8, 4 spaces, Python 3.7 (venv) (2).

