Python Driverless Car Project

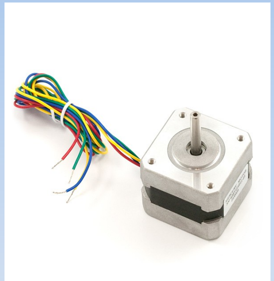
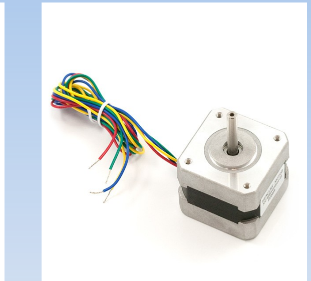
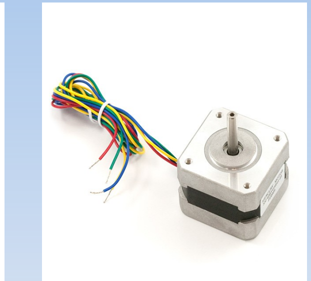
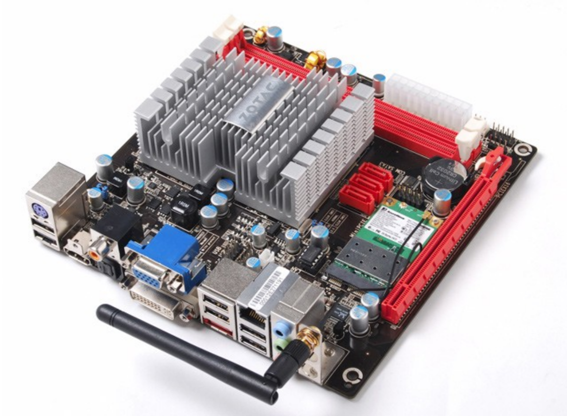
- Speaking at the American Association for the Advancement of Science conference in San Francisco, **Sebastian Thrun**, a Stanford University computer science and electrical engineering professor, estimated that **robot-driven cars will be road ready by 2030**.

Shown here is the Google Driverless Car. It's being road tested right now

Ok cool... but why go Open Source? Why Python?

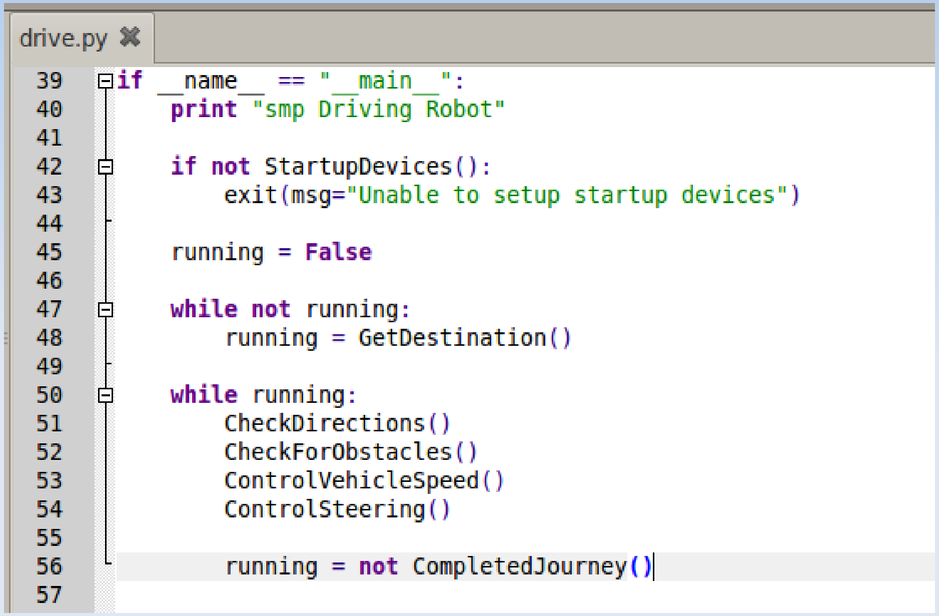
- The Google project isn't Open Source. No way for people to jump in and 'hack python code'.
- We want to write something cool in Python that we can have in our car at all times.. spending time in the car without python is worrying..
- Actually Python has everything needed to make a fantastic Driverless car project..

Hardware – what do we need ?

- Multicore 12v ITX motherboard (10cm x 17cm), $130
- 2 x Webcams, 2 x @ $30 each
- Ultrasonic distance sensor
- Stepper motors to drive accelorator, brakes and steering
- Usb stick to hold O/S and program.. GPS..
- Other things not shown...

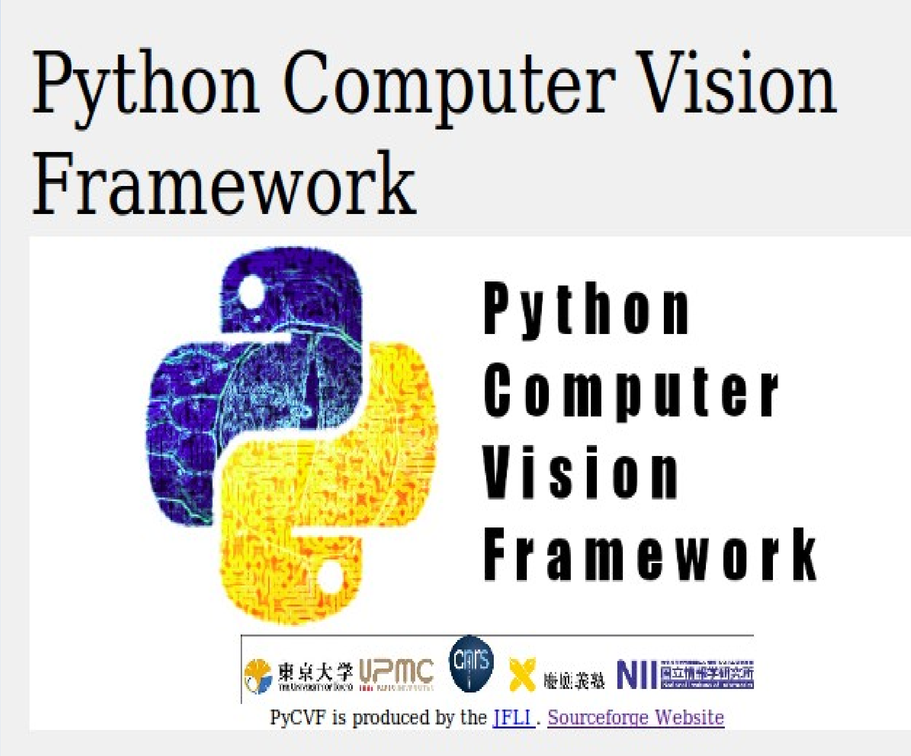
Driving is simple right? 20 lines of python ?



```python
if __name__ == "__main__":
    print "smp Driving Robot"

    if not StartupDevices():
        exit(msg="Unable to setup startup devices")

    running = False

    while not running:
        running = GetDestination()

    while running:
        CheckDirections()
        CheckForObstacles()
        ControlVehicleSpeed()
        ControlSteering()

        running = not CompletedJourney()

```

- The great thing about python is that it's very high level and very easy to understand what's going on.
- In fact, we might be able to write the high level portion of the controller in about twenty lines of code...

How will the car 'see' ?



Python Computer Vision Framework

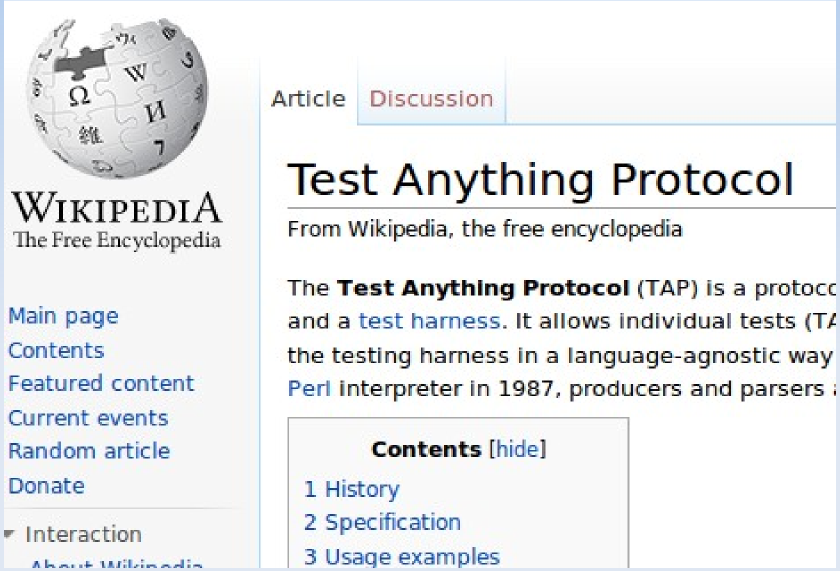
PyCVF is produced by the JFLI . Sourceforge Website

- Python can see. It binds to OpenCV, an Open Source Computer Vision Project.

Surely it will need some tests ?

- Actually, how do you test a car robot?
- Can we go crash testing ?

No – we use TAP and UnitTests !



Test Anything Protocol

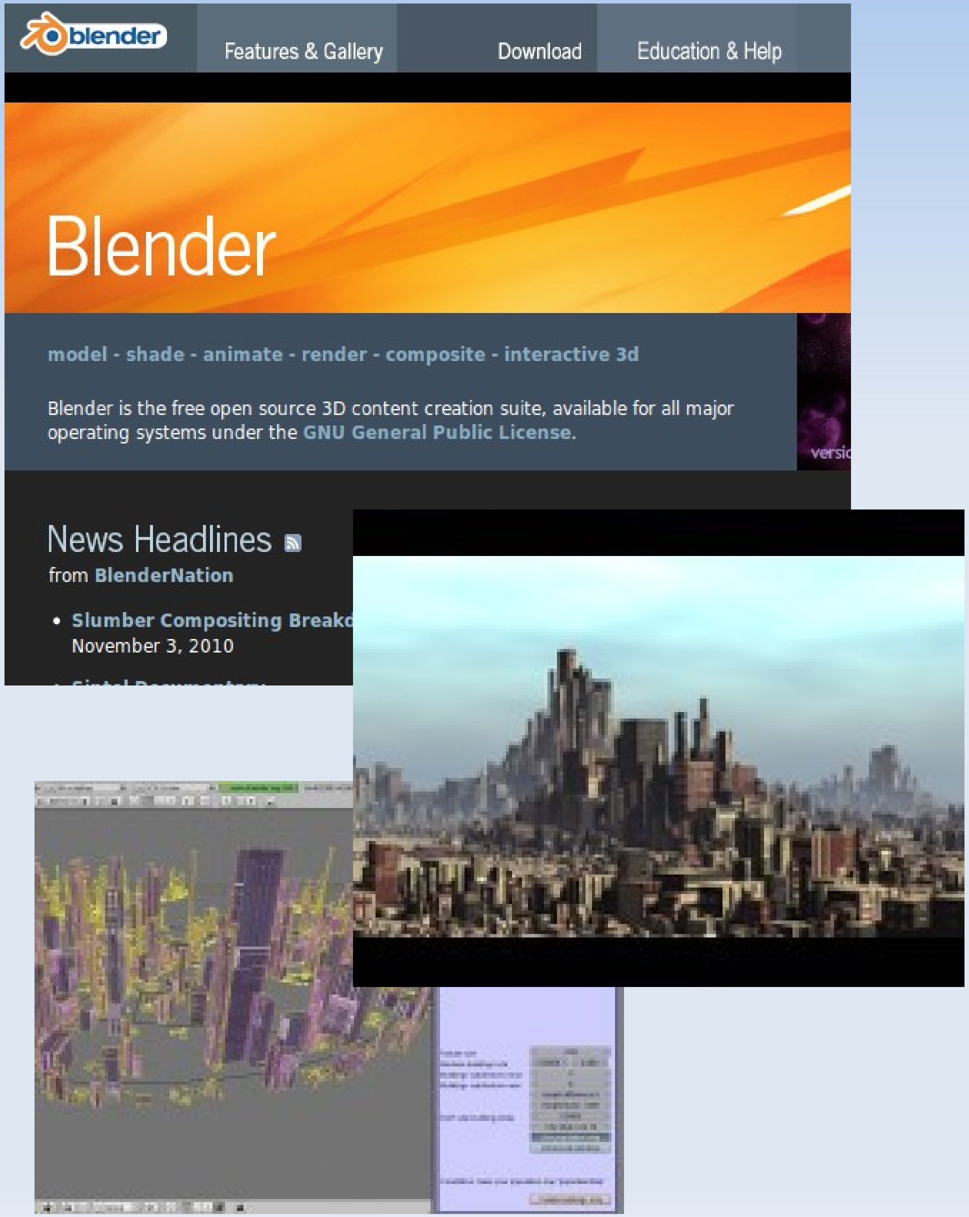
From Wikipedia, the free encyclopedia

The **Test Anything Protocol** (TAP) is a protoco
and a test harness. It allows individual tests (TA
the testing harness in a language-agnostic way
Perl interpreter in 1987, producers and parsers a

Contents [hide]
1 History
2 Specification
3 Usage examples

- Actually we can test a driverless car robot just like we would test any other piece of software
- But what tests ?
- Every test we can think of:
  - Traffic ie cars ahead
  - Pedestrians
  - Navigation etc etc

Don't we need 3D tests ?



model - shade - animate - render - composite - interactive 3d

Blender is the free open source 3D content creation suite, available for all major operating systems under the GNU General Public License.

News Headlines

from BlenderNation

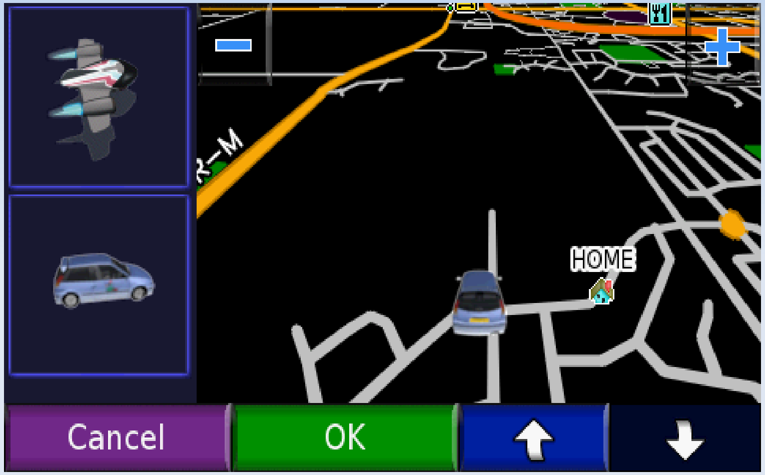
- Slumber Compositing Breakd
  November 3, 2010

- Yes, python can generate 3D scenes using Blender
- Any scene or scenario can be generated
- The tests can be controlled in blender via python scripting.

3D Output becomes test vision input..

- The Blender games engine generates simulations that are surprisingly realistic
- Python can read the 3D output as though it was coming from a real video feed.
- Every possible scenario can be tested in a way not possible with real testing.

Making and testing 3D models..



- Have some fun in python with 3D...
- Generate test cases for all conceivable situations
- Generate maps and routes
- Read 'real' road imagery from Google Streetview

Interface to the car...

- PySerial to talk to the serial ports
- Use python bluetooth to talk to the GPS
- Control Steering Navigation and Braking with industrial controllers...

After all that...

- Use the car like you know it was made for..

Thank you

www.bitbucket.org/djlyon/smp-driverless-car