Creating Games With Python And Java

Davis Silverman

January 7, 2014

About Me

- Amateur programmer and game developer
- High school now, college in the future
- Seeking a job as a developer! *wink wink, nudge nudge*

Jython

- CPython is the reference implementation for Python.
- Jython is a Python implementation in Java
- Offers superb interoperability with Java libraries, along with the amazing benefits of the JVM
- 2.7 betas are out!

Comparisons of CPython and Jython

```
#Jython                  | #CPython
def fibonacci():         | def fibonacci():
    a, b = 0, 1          |     a, b = 0, 1
    while True:          |     while True:
        yield a          |         yield a
        a, b = b, a + b  |         a, b = b, a + b
```

LibGDX

- A cross-platform Java game development framework based on OpenGL (ES) that works on Windows, Linux, Mac OS X, Android, your WebGL enabled browser and iOS.

My Work with LibGDX

- Translated the LibGDX wiki (GoogleCode -> Github)
- Worked on Polyglot LibGDX (as shown in this very talk!)
- Regular on IRC
- Started game-dev club at my school to teach and create games with LibGDX

LibGDX Classes Of Use

- `ApplicationListener` is the base java interface for a LibGDX game
- `OrthographicCamera` for camera magic
- `SpriteBatch` to draw on the screen
- Standard math classes `Vector2`, `Rectangle`, etc.

Small example!

- Small game from our wiki translated to Python
- To the demo! (I hope this works!)

Limitations of LibGDX with Jython

- GWT
  - This backend is Java only, so *non-Java* HTML LibGDX backend is a pipe dream
  - Scala *might* work, as they seem to have some sort of scala-gwt in the works.
- Enforces an OO approach.

Future

- Android Support
  - Once Jython can attain DynamicProxy support, it might be possible to have Jython-LibGDX on Android!
- iOS support
  - the RoboVM backend runs the Android class library, so if it can Android, there is a good chance it can iOS!
- Packaging
  - There has been some work on compiling/packaging Jython into jars, this will make distribution of your awesome Python games very easy!

More pythonic LibGDX (mostly random ideas)

- `with render(batch): ...`
- Extending LibGDX util classes to conform to python: `len(com.badlogic.utils.Array())`
- So much more good stuff (magic so not enforced OO? WHO KNOWS?!?!)
- Reduce need of both `__init__` and `create` methods (possibly using metaclass magic?)
- Runtime introspection so `dispose()` is not needed?
- Jython3k and function annotations could help Jython when interoperating with their static host languages

Thanks!

- Jim Baker, who has given me lots of insight into Jython, and convinced me to do this talk.
- ZPUGDC (DCPython), for having me. <3
- The internet, for helping me learn so much.

Links!

- Jython: http://jython.org
- Jython Book: http://www.jython.org/jythonbook/en/1.0/
- LibGDX: http://libgdx.badlogicgames.com/
- LibGDX Wiki: https://www.github.com/libgdx/libgdx/wiki
- This talk: https://www.github.com/sinistersnare/JythonTalk (needs latex-beamer and pandoc)