Python Wars Solo Documentation

Release 1.0.0

Daniel Greenfeld

Sep 27, 2017

Contents

My retro-style Python clone of a Apple][Basic game I made back around 1982.

CHAPTER 1

Features

- Command the Battle Cruiser Pythonista!
- Fight the good fight against the evil Zargons
- Use your ship's spinal mounted beam cannon to slice and dice the enemy to pieces!
- Launch missiles to destroy Zargons in single shots
- Fight up to 9 enemies!
- Old school text based game
- Easy to learn
- Addictive

CHAPTER 2

Usage

From the command-line:

```
python go.py
```

CHAPTER 3

Contents

History of Python Wars Solo

Python Wars Solo is the result of a few hours effort roughly duplicating a text-based Star Trek game I wrote back in 1980-1981. You fought up to 9 Klingons in your Enterprise. Beating one was a piece of cake. Three was a fun challenge. Five was tough. Seven was done only a few times. Nine was never done. The game was simple, fast, easy to learn, and tons of fun.

Now I don't remember much about the mechanics of the code I wrote back in High School. So when I started writing Python Wars Solo I decided not to worry about it. I would code how I felt like coding, and just create a game.

There were a few false starts. I kept trying to add tons of complexity to the code, or lots of neat features. Lots of time was wasted and not much was done. The technical term for what I was doing is 'Scope Creep'. Then someone advised that I just make it really simple and get it done.

And I did. I got it done.

To avoid potential yet likely silly copyright/trademark issues, I renamed the ship the '*Pythonista*'. The enemies are the evil '*Zargons*'.

Future Version Thoughts

Originally I thought of expanding this out to become a game with graphics and maybe a campaign. Then I went onto other things and got too preoccupied. But here was my original list of things to add:

- Keep basic mechanics
- Add 2-D map
- Incorporate PyGame
- Since all components are objects on ship objects which exist in the space object, current version could easily be expanded:
- Let players build their own ships and use introspection to generate menus

- Damage the ship in ways so that parts get broken

Reference: Stuff

class `stuff.`**`Component`**
Parts of the ship

`action`()

class `stuff.`**`Cruiser`**(*name=''*, *id=''*)

`damage_control`()

`recharge`()

class `stuff.`**`ECM`**

`action`(*me=None*, *enemy=None*)

class `stuff.`**`Frigate`**(*name=''*, *id=''*, *dv=None*)

`recharge`()

class `stuff.`**`Missile`**

`action`(*me=None*, *enemy=None*)

class `stuff.`**`Pulsar`**

`action`(*me=None*, *enemy=None*)

class `stuff.`**`Ship`**
The big ship object

`take_damage`(*damage*)

class `stuff.`**`Spinal_Mount`**
Big honking laster

`action`(*me=None*, *enemy=None*)

`recharge`()

class `stuff.`**`Universe`**
Container for ships

`list_ships`()

CHAPTER 4

Indices and tables

- genindex
- modindex
- search

Python Module Index

Index