In [1]:
```
from fighter import Fighter
from side import Side
```

In [2]:
```
def run():
    start = (1, 'Viper', 100, 100, 30)
    f = Fighter(start, Side.Blue)
    t, dt = 0.1, 0.1

    while t < 1000.0:
        if t > 050.0: f.desired_heading = 000.0
        if t > 200.0: f.desired_heading = 180.0
        if t > 300.0: f.desired_heading = 270.0
        if t > 400.0: f.desired_heading = 045.0
        if t > 500.0: f.desired_heading = 000.0
        if t > 600.0: f.desired_heading = 045.0
        if t > 700.0: f.desired_heading = 135.0
        if t > 800.0: f.desired_heading = 180.0
        if t > 900.0: f.desired_heading = 215.0
        f.tick(t, dt)
        t += dt

    return f
```

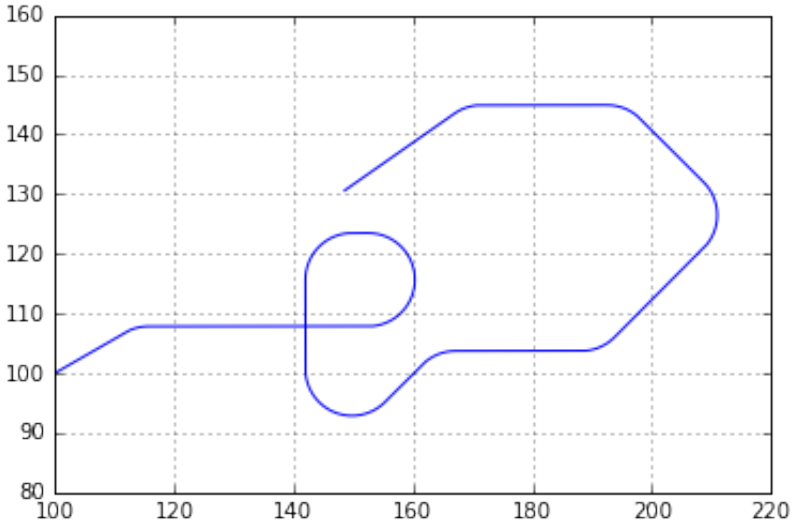
In [3]:
```
f = run()
```

In [4]:
```
%matplotlib inline
#%matplotlib notebook
import matplotlib.pyplot as plt
import numpy as np
```

```
/Users/mikepsn/anaconda/lib/python2.7/site-packages/matplotlib/font_manager.py:
273: UserWarning: Matplotlib is building the font cache using fc-list. This may
take a moment.
  warnings.warn('Matplotlib is building the font cache using fc-list. This may
take a moment.')
```

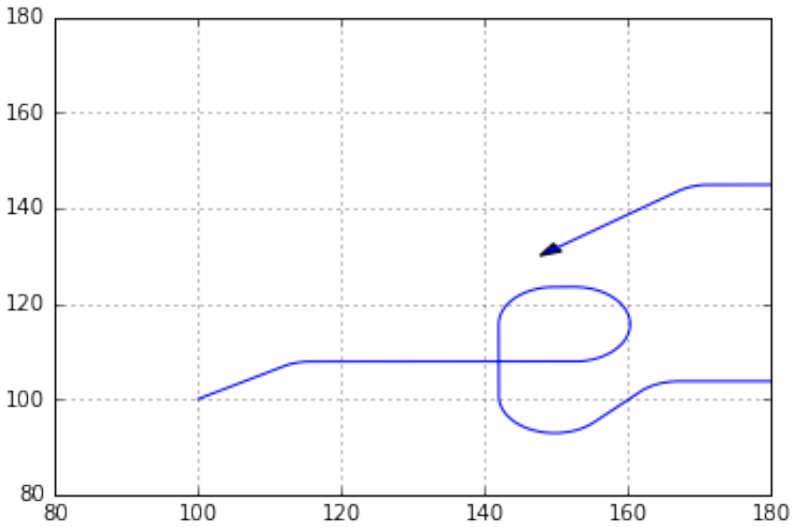
In [5]:
```
x = np.array([pos[1] for pos in f.trace])
y = np.array([pos[2] for pos in f.trace])
h = np.array([pos[3] for pos in f.trace])
plt.plot(x, y, color='b', linestyle='-', markersize=10)
plt.grid(True)
plt.axis('equal')
```

Out[5]: (100.0, 220.0, 90.0, 150.0)



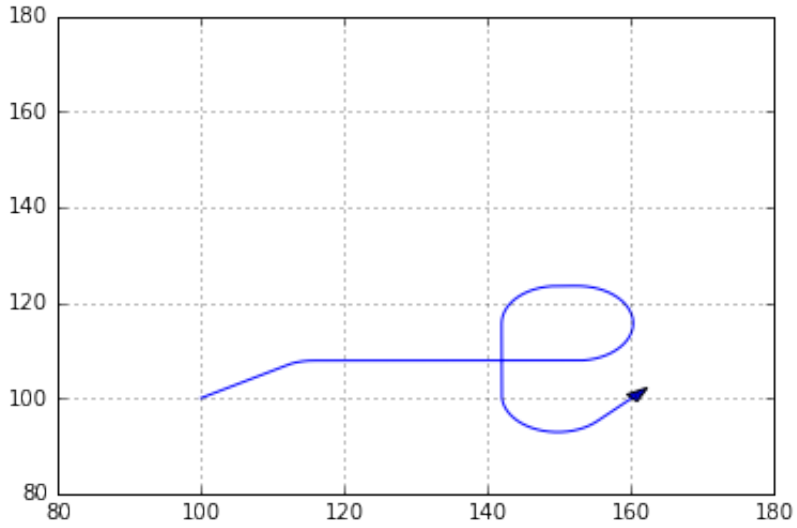
In [6]:
```
from plotrun import plot_path
```

In [7]:
```
plot_path(x, y, h, len(x)-1)
```



In [8]:
```
from ipywidgets import interact, interactive, fixed
import ipywidgets as widgets
```

In [9]:
```
p = interact(plot_path, x=fixed(x), y=fixed(y), h=fixed(h), t=(0,len(x)-1,1))
```



In [10]:
```
def draw_run(positions, t=25, style="-"):
    """ Note t is an index into an array representing the timeslice we are intere
sted in."""
    #px, py, pz = p[0:t,0], p[0:t,1], p[0:t,2]
    colours = ['Blue', 'Red', 'Green']
    callsigns = {'Blue' : 'Warlock', 'Red' : 'Shogun', 'Green' : 'Viper'}
    figure = plt.figure(figsize=(10,10))
    #figure = plt.figure(figsize=(8,7))
    axes = figure.gca(projection='3d')
    axes.set_xlim([-15, 15])
    axes.set_ylim([-15, 15])
    axes.set_zlim([45, 55])
    for ball in xrange(positions.shape[0]):
        p = positions[ball]
        px, py, pz = p[0:t,0], p[0:t,1], p[0:t,2]
        timeslice_plot = axes.plot(px, py, pz, c=colours[ball], linestyle=style)
        timeslice_plot = axes.text(p[t,0], p[t,1], p[t,2], callsigns[colours[ball
]], color=colours[ball])
    figure.show()
    #return timeslice_plot
```

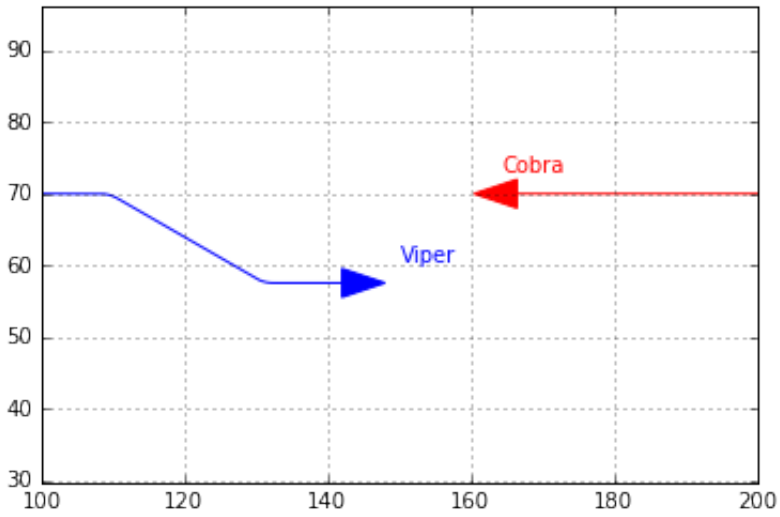
In [11]:

```
def sim():
    viper_start = (1, 'Viper', 100, 70, 0)
    cobra_start = (2, 'Cobra', 200, 70, 180)

    viper = Fighter(viper_start, Side.Blue)
    cobra = Fighter(cobra_start, Side.Red)

    t, dt = 0.1, 0.1

    while t < 1000.0:
        if t > 050.0: viper.desired_heading = 330.0
        if t > 200.0: viper.desired_heading = 000.0
        if t > 330.0: viper.desired_heading = 090.0
        if t > 400.0: viper.desired_heading = 180.0
        if t > 500.0: cobra.desired_heading = 300.0
        if t > 550.0: viper.desired_heading = 300.0
        if t > 600.0: cobra.desired_heading = 200.0
        if t > 650.0: viper.desired_heading = 220.0
        if t > 700.0: viper.desired_heading = 350.0
        if t > 750.0: cobra.desired_heading = 340.0

        viper.tick(t, dt)
        cobra.tick(t, dt)

        t += dt

        traces = [cobra, viper]

    return traces
```

In [12]:
```
def plot_path(x, y, h, t):
    plt.plot(x[:t], y[:t], color='b', linestyle='-')
    #plt.xlim([80, 180])
    #plt.ylim([80, 180])
    plt.grid(True)

    r, theta = 1, np.deg2rad(h[t])
    dx, dy = r * np.cos(theta), r * np.sin(theta)

    plt.arrow(x[t], y[t], dx, dy,
              overhang=0, width=0.1,
              shape='full', length_includes_head=True,
              head_starts_at_zero=False)
    plt.show()

def plot_traces(traces, t=6000):
    plt.grid(True)
    plt.axis('equal')
    #plt.xlim(50,200)
    #plt.ylim(50,100)
    colour = {Side.Blue: 'b', Side.Red: 'r'}
    for fighter in traces:
        x = np.array([pos[1] for pos in fighter.trace])
        y = np.array([pos[2] for pos in fighter.trace])
        h = np.array([pos[3] for pos in fighter.trace])

        plt.plot(x[:t], y[:t], color=colour[fighter.side])
        plt.text(x[t]+3, y[t]+3, fighter.callsign, color=colour[fighter.side])

        r, theta = 1, np.deg2rad(h[t])
        dx, dy = r * np.cos(theta), r * np.sin(theta)
        plt.arrow(x[t-1], y[t-1], dx, dy, overhang=0, width=0.2, fc=colour[fighte
r.side], ec=colour[fighter.side],
              shape='full', length_includes_head=True,head_starts_at_zero=False)

    plt.show()
```

In [13]:
```
traces = sim()
```

In [14]:
```
i = interact(plot_traces, traces=fixed(traces), t=(1,9999-1,1))
```



In [ ]:
```

```

In [ ]:
```

```