Classes:

main.py uses:

- Ygraphics
- Yvector2D.
- Yworld
- Yagent

Agent.py uses:

- Yvector2D
- Ypoint2D
- Ygraphics

Graphics.py uses: None

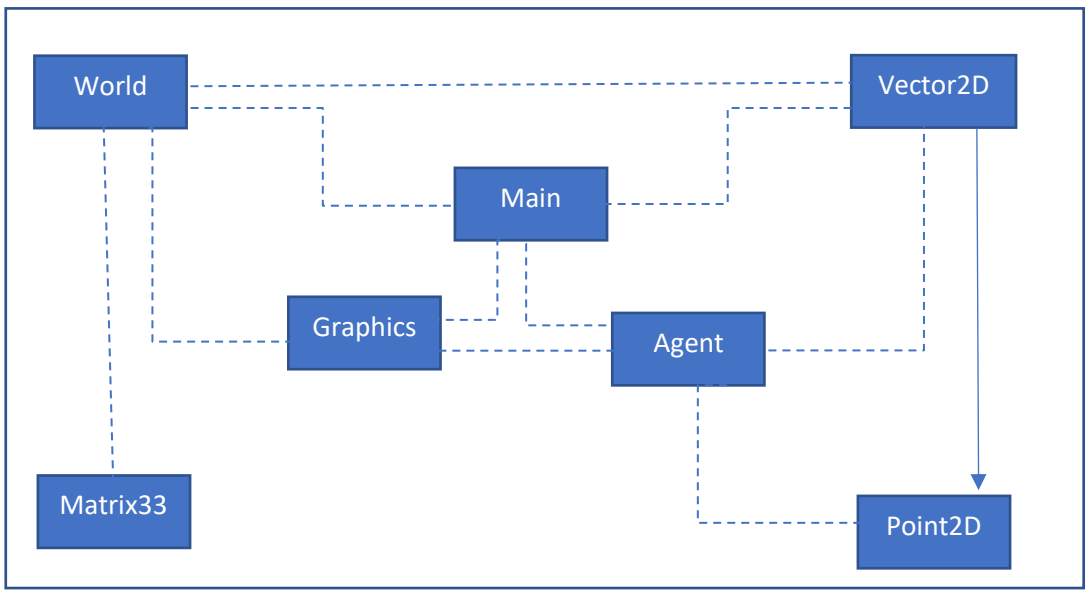
Matrix33.py uses: None

Point2D.py uses: None

World.py uses:

- Vector2D
- Ymatrix33
- Ygraphics

UML:



Flee:

```
    def flee(self, hunter_pos):
        ''' move away from hunter position '''
## add panic distance (second)
        panic_distance = 1000.0
        hunter = hunter_pos - self.pos
        dist = hunter.length()
        if dist < panic_distance:
## add flee calculations (first)
            return (self.pos - hunter_pos).normalise() * self.max_speed
            return Vector2D()
```

Deceleration:

```
class Agent(object):

    # NOTE: Class Object (not *instance*) variables!
    DECELERATION_SPEEDS = {
        'slow': 0.9, 'normal': 0.5, 'fast': 0.1
        ### ADD 'normal' and 'fast' speeds here
    }
```

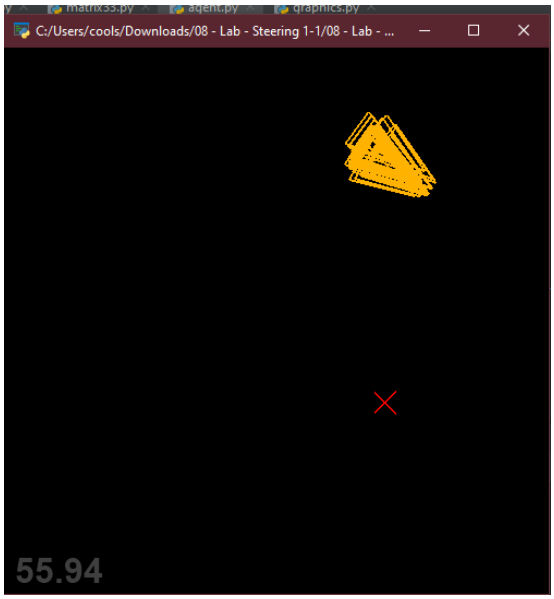
Calculate:

```
    def calculate(self):
        # reset the steering force
        mode = self.mode
        if mode == 'seek':
            accel = self.seek(self.world.target)
        elif mode == 'arrive_slow':
            accel = self.arrive(self.world.target, 'slow')
        elif mode == 'arrive_normal':
            force = self.arrive(self.world.target, 'normal')
        elif mode == 'arrive_fast':
            force = self.arrive(self.world.target, 'fast')
        elif mode == 'flee':
            accel = self.flee(self.world.target)
        elif mode == 'pursuit':
            force = self.pursuit(self.world.hunter)
        else:
            accel = Vector2D()
        self.acceleration = accel
        return accel
```

Default mode set to seek and max speed to 1000.0:

```
    def __init__(self, world=None, scale=30.0, mass=1.0, mode='seek'):
        # keep a reference to the world object
        self.world = world
        self.mode = mode
        # where am i and where am i going? random
        dir = radians(random()*360)
        self.pos = Vector2D(randrange(world.cx), randrange(world.cy))
        self.vel = Vector2D()
        self.heading = Vector2D(sin(dir), cos(dir))
        self.side = self.heading.perp()
        self.scale = Vector2D(scale, scale)  # easy scaling of agent size
        self.acceleration = Vector2D()  # current steering force
        self.mass = mass
        # limits?
        self.max_speed = 1000.0
        # data for drawing this agent
        self.color = 'ORANGE'
        self.vehicle_shape = [
            Point2D(-1.0,  0.6),
            Point2D( 1.0,  0.0),
            Point2D(-1.0, -0.6)
        ]
```

Output:



Changing properties to set mode to flee and max speed to 900:

```
    def __init__(self, world=None, scale=30.0, mass=1.0, mode='flee'):
        # keep a reference to the world object
        self.world = world
        self.mode = mode
        # where am i and where am i going? random
        dir = radians(random()*360)
        self.pos = Vector2D(randrange(world.cx), randrange(world.cy))
        self.vel = Vector2D()
        self.heading = Vector2D(sin(dir), cos(dir))
        self.side = self.heading.perp()
        self.scale = Vector2D(scale, scale)  # easy scaling of agent size
        self.acceleration = Vector2D()  # current steering force
        self.mass = mass
        # limits?
        self.max_speed = 900.0
        # data for drawing this agent
        self.color = 'ORANGE'
        self.vehicle_shape = [
            Point2D(-1.0,  0.6),
            Point2D( 1.0,  0.0),
            Point2D(-1.0, -0.6)
        ]
```

Output:


C:/Users/cools/Downloads/08 - Lab - Steering 1-1/08 - Lab - ...
50.60
Terminal
Python Console
6: TODO