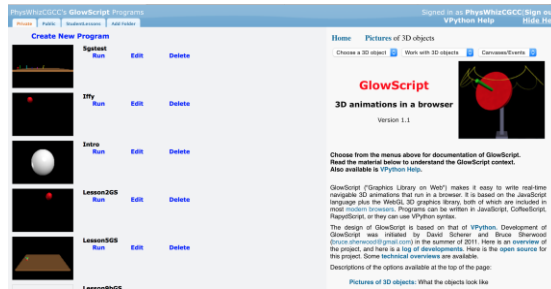
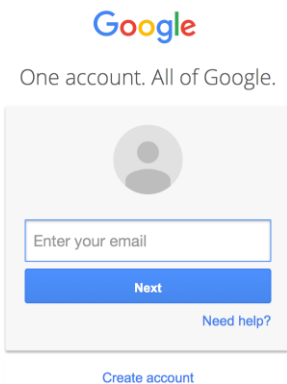


Fun with VPython (4.5b GS)



Create GlowScript Account

If you've had issues with getting VPython to download and install properly on your computer and/or you want to experiment with a browser-based solution, go to glowscript.org, and click on the **Sign In** link in the upper right corner.

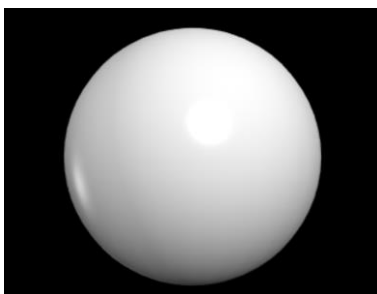


If this is your first time here, then you should click on the **Create account** link towards the bottom of the next screen.

Once you've created an account, you should check that all is well. Click on the **Create New Program** link, provide a name for the program (I called mine **Test**), and you'll be taken to an editing window.

```
Test by PhysWhizCGCC (Saved)
Run this program
1 GlowScript 1.1 VPython
2 # David Weaver Test
3 sphere ( )
```

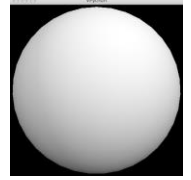
Line #1 is always filled in I added **# David Weaver** beginning with a “#” are a doesn't try to do anything 3 which tells the program



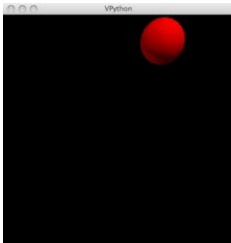
The major change to **GlowScript** for most of **VPython** programs and previous training materials is the need for making object's positions, colors, etc. as **vectors**. Many of the following example programs also include the statement **from visual import *** so these programs can also run directly in VPython. Also, **frames** are not available in **GlowScript**. The VPython version of this training has associated videos, but they are likely not necessary.

Lesson 1 – Create your 1st Program

- You will replicate what you did above and add what is required.
- Create your own 1st Program based on this [sample](#) (use *# Comments* as directed and make sure your name is included in the file name)
- Submit your first program as directed by your instructor.



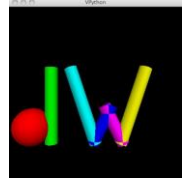
Lesson 2 – Adjust 1st Program to change position, size and color of your sphere



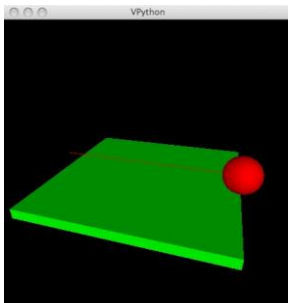
- Look at the Lesson 2 [example](#).
- Create your own 2nd Program (use *# Comments* as directed and make sure your name is included in the file name (use comments and proper file naming for all future programs)) for a named sphere with a position, size and color of your choosing.
- **Challenge** – Place multiple spheres (of different sizes and colors) in multiple places.
- **More challenging** – Go to the VPython Help and Help links in upper right corner of the GlowScript screen. Figure out how to create different sorts of 3-D shapes and add them to your program.
- Submit program(s)

Lesson 3 – Create your initials with VPython (helps you learn about 3-D space)

- Look at the Lesson 3 [example](#).
- Create your initials.
- **Challenge** – Capture this image so you can use it to “brand” your documents, websites, etc.
- Submit program(s)



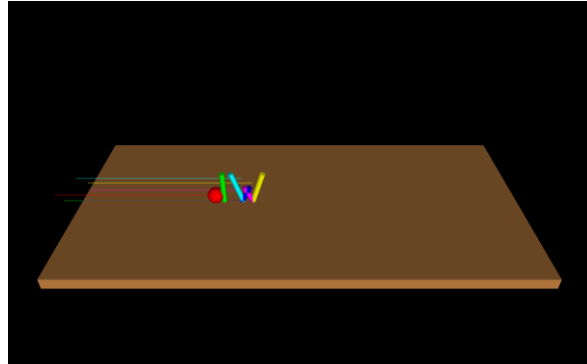
Lesson 4 – Make a sphere move (finally starting to see some physics!)



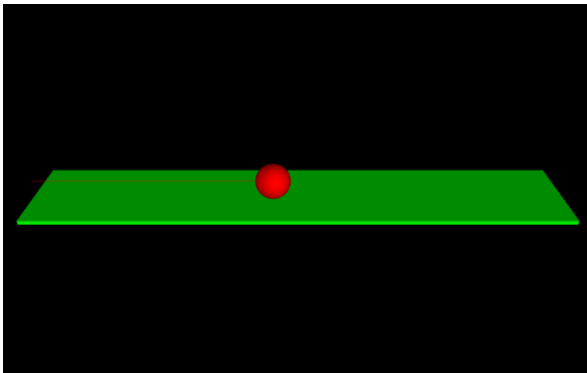
- Look at the Lesson 4 [example](#).
- Make your sphere move at a different velocity than this.
- **Challenge** – Create two different objects and make them move opposite directions.
- **More challenging** – Create several different objects and make them each travel different directions using all three dimensions.
- **Even more challenging** – Look at the Lesson 4b [example](#) and add trails to your moving objects. Also notice the adjusted view.
- Submit program(s)

Lesson 5 – Use what you learned in Lessons 3 & 4 to animate your initials

- Look at the Lesson 5 [example](#).
- Make your initials move similarly. *Be sure to add trails and adjust the view in this and all future programs.*
- **Challenge** – Make each initial move in a different direction.
- Submit program(s)



Lesson 6 – Accelerating ball (more physics!)



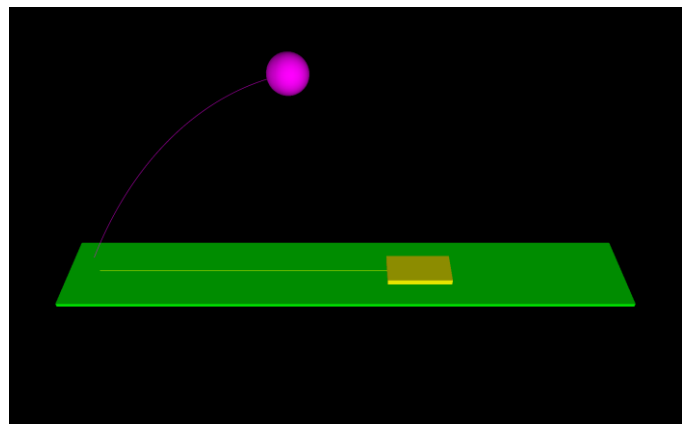
- Look at the Lesson 6 [example](#).
- Make your object a different color, start at a different place, and accelerate at a different rate.
- **Challenge** – Create two objects and have one move at a constant velocity while the other accelerates.
- **More challenging** – Allow the constant velocity object a 5 second head start before the accelerating object

begins.

- Submit program(s)

Lesson 7 – Projectile Motion

- Look at the Lesson 7 [example](#).
- Mimic with different values.
- **Challenge** – Create an object that moves horizontally at a constant rate and another that is fired as a projectile so that it “hits” the other object.
- **More challenging** – Allow the constant velocity object a 5 second head start before the projectile object begins.
- Submit program(s)

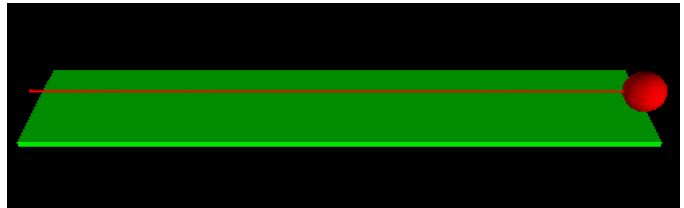


Lesson 8 – Constant Force



- Look at the Lesson 8 [example](#).
- Mimic with your values.
- **Challenge** – Make two objects with the same mass move in two dimensions by applying the same magnitude of force in two directions.
- **More Challenging** – Create a modified Atwood’s machine and explore the effects of changing the value of the hanging mass.
- **Even More Challenging** – Modify your Atwood’s machine to keep the total mass constant and explore the effects of changing the mass ratio.
- Submit program(s)

Lesson 9 – Momentum



- Look at Lesson 9a [example](#).
- Mimic with your own values (change shapes, colors, masses, etc.).
- Look at Lesson 9b [example](#).
- Mimic with your own values (change shapes, colors, masses, etc.).
- **Challenge** – Research the use of momentum within VPython and modify one of your “finds” to include the various naming, labeling, commenting, viewing, trailing, etc. procedures established in previous lessons.
- Submit program(s).