Creating a GravBox: Visualizing Gravitation Potential Wells

Quinton Snouffer
Dr.'s Charles Kuehn and Courtney Willis

ABSTRACT

Interactions with gravitational potential wells (GPWs) are extremely difficult to imagine, especially when considering multiple GPWs. A gravity sandbox, formerly known as a GravBox, is a piece of technology capable of simulating gravitational potential wells and their surroundings. Gravitational potential wells are created by every object that has mass; the more massive an object, the deeper the potential well will be, and the more difficult it would be to escape its gravitational force. As one creates craters in the sandbox, large and small, a range finder will map out the topology and project a real time, color coded surface along with the behavior of a particle moving through the created potential wells. Many students studying physics and astronomy struggle with conceptualizing more than one GPW; this research is a perfect opportunity to allow students to visualize an object traveling through GPWs, which they created themselves, as if an object was traveling through areas with differing forces of gravity in our universe. I have always had an interest in gravity, as well as finding tools to help others be more successful, and this research was the perfect opportunity to fulfill both of those interests. Currently, I have a perfected fabrication and assembly of the sandbox itself and now in the process of fine tuning the software. Once GPW software has been perfected, some additional coding and physics could be applied to the same sandbox and technology, allowing us to visualize electromagnetic and quantum mechanical potential wells. If successful, gravitational, electromagnetic, and quantum mechanical potential well simulations could be an invaluable teaching tool.

BACKGROUND

Gravity is the force by which all things with mass, therefore energy, are attracted towards. The more mass an object has, the more gravitational force is associated with it. Galaxies have massive gravitational forces, followed by solar systems, stars, planets, etc. Gravitational potential energy is described by the following equation:

$$U = -\frac{GMm}{r}$$

where G is the gravitational constant (6.67*10^-11 m3/kg*s), M is the mass of the attracting body, m is the mass of the attracted body, and r is the distance between the centers of the two bodies.

The best way to visualize a gravitational force between bodies is to imagine a well surrounding the object; as an object becomes more massive, the potential well associated with it becomes deeper. Now imagine space as a flat sheet of elastic material (like what is shown in figure 1); if you place a heavy object on the sheet, a well will form (this is a representation of Einstein's Theory of General Relativity). As more objects get added to the sheet, it is very difficult to imagine the wells and how they might interact. A GravBox is an extremely useful way to visualize this phenomenon. It allows you to create potential wells directly in a sandbox. Technology then allows a topological graph and simulation to show a small body moving throughout the gravitational potential field.

As seen in the figure below, gravitational fields can become crowded with potentials. This leads to an inability to compute the total gravitational potential energy on an object in any sort of timely manner. The GravBox code is embedded with equations and has the ability to read depth perception; this allows for the craters in the sand to be depicted as massive objects, whereas hills depict the absence of mass. The size of the well is converted to a mass in the system, where the magnitude of the gravitational force can be calculated in real time. The system does these calculations for all present wells and projects a color coded topological map onto the surface of the sand (blue indicates large mass, red indicates minimal mass). Finally, a test particle is shown moving throughout the field, affected by the various gravitational forces in real time.

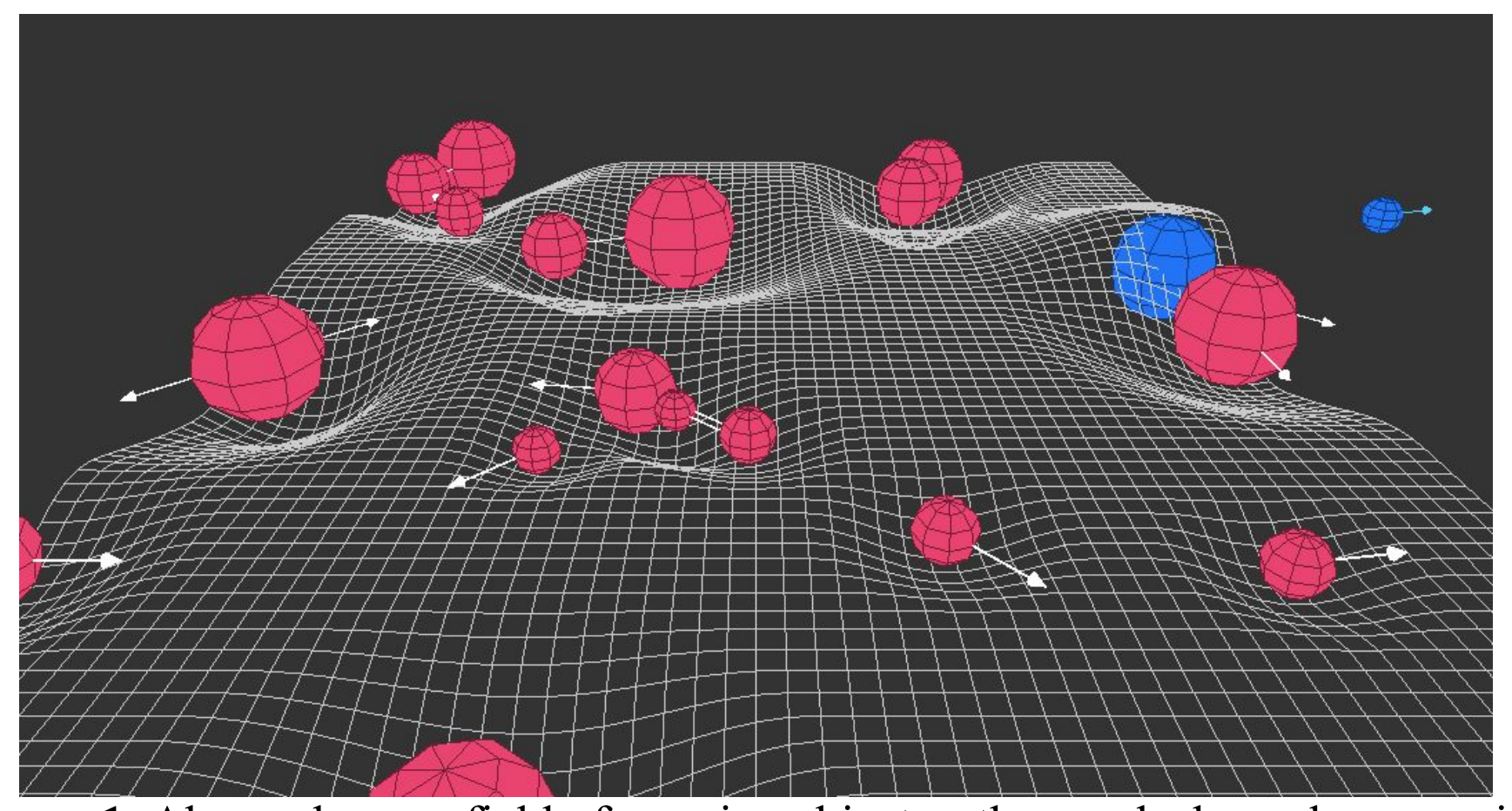


Figure 1. Above shows a field of massive objects-- the graph shows how gravitational potential wells form. An object on the graph would want to "fall" into wells created by other massive objects. (physics stack exchange)

MATERIALS

The materials necessary for this research project included a sandbox consisting of a ¾ inch CDX plywood bottom and 2x8 sides (with sand to fill the box, of course), a cart system to transport the sandbox, projector, Xbox Kinect camera, and a computer with a Linux operating system capable of running large amounts of Python 2.7 code.

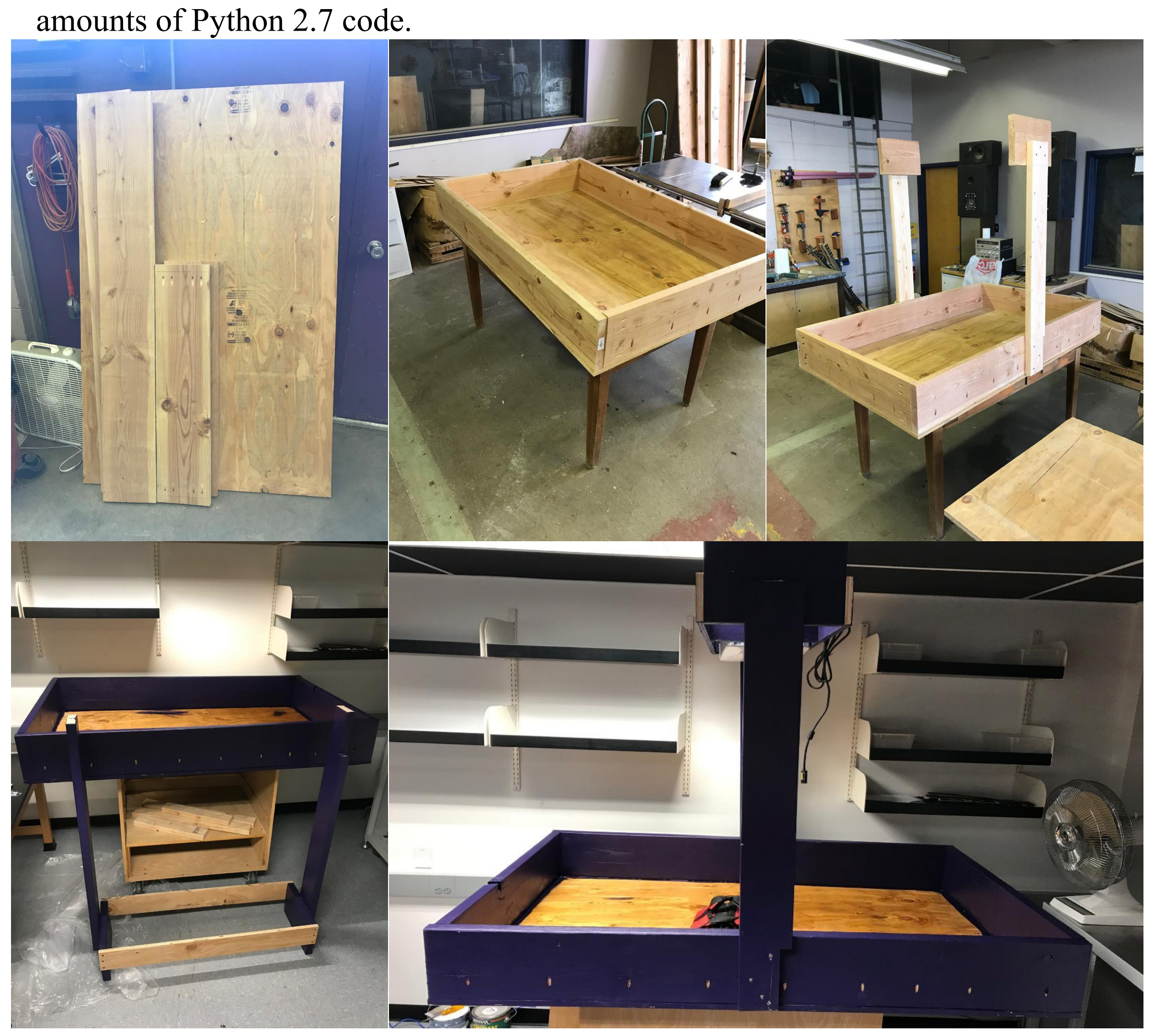


Figure 2. showing the process of building the sandbox.

As far as electronics go, I used a Dell Precision 5820 for the computer, BenQ mw632st for the projector, and an Xbox 360 (model 1414) Kinect camera system for spatial recognition. Initially, I was using the newest version of the Kinect but lacked success; I was unable to resolve errors in the code using the older Kinect due to the different way the system communicates with the camera.

METHODS

As you might have heard before-- measure twice, cut once. I learned this lesson one too many times during the creation process of the sandbox.

In order to build the sandbox, I used several different woodworking tools to obtain a sturdy, effective design. I created the bottom surface out of ¾ inch CDX plywood with 2x8 walls. I used pocket holes with screws to connect the side walls and to achieve a snug fit; the walls were connected to the bottom surface with another set of pocket holes and screws. After sealing with polyurethane and silicone the box was filled with sand, leaving me with a surface perfect to use as a screen (this took me two tries because I did not measure twice and cut once). In my case the projector was mounted at a distance of one meter; this can change depending on aspect ratio of the projector as well as picture size of the projector. It was important to mount the projector and kinect near each other as well as back to back in order to fulfill calibration and code requirements.

Finally, after assembly and the connection of all components, it was time to dig into the code. The source code can be found on the *GravBox - The University of Iowa* website. The team at University of Iowa has their code uploaded to GitHub for public use; although the code is easily accessible, it was quite the process of deciphering and troubleshooting. For several weeks I was learning the process of using Linux as well as Python, all the while trying to decipher each subsequent error message. Each error message was resolved in as little as a few minutes up to several days. At one point mentioned above, I was unable to resolve an error code received from the Xbox One Kinect. After much research I realized that the Xbox Kinect is difficult to communicate with using a computer. I switched to the model 1414 Kinect and was able to open the program without any error messages. The program ran and looked beautiful-- until the sand was moved and the program shut down. My last error I had to decipher involved this issue; the program would run but would shut down if anyone moved that sand. This was a problem because being able to move the sand is the entire point of this project. At last, after many days of research and troubleshooting, I was able to increase the height restrictions of the reading surface and the GravBox was a working piece of technology (although still not perfected).

CALIBRATION

After running the code for the first time, my calibration for the projector's image did not align with the wells of the sand. Additionally, when one placed their hands in the sandbox to create new wells, the program shut itself off due to distance constraints from the Kinect. After much troubleshooting and frustration, I realized that I had skipped a step in the calibration process. The calibration process involved downloading the SARndbox from UC Davis; this calibration process was very in depth and required many steps involving both the Kinect as well as the projector in order to align the projection. After completing the calibration process, my projection aligned with the sand, but still left a vertical distance constraint.

I determined that the code is written as such to ignore somebody's arm/hand while inside the sandbox; my code was not doing this, leading to a crash of the system. I was able to locate a line in the code to allow for somebody's arm to be in the sandbox, however it was still not working as expected. The vertical distance constraint was gone at this point, but caused for peculiar reactions from the topology. The original line of code came from the file "topogra.py" and called for a floor height of 500:

```
if len(np.where(topo<FLOOR)[0]) + len(np.where(topo<FLOOR)[1]) >500:
```

The floor height refers to a vertical distance above the sand that the scale factor for topology is no longer exponential, but instead linear. Since the linear scaling is not working as it should, I raised this floor height to allow someone's hand to be in the sandbox without crashing:

```
if len(np.where(topo<FLOOR)[0]) + len(np.where(topo<FLOOR)[1]) >
1000000000:
```

This is a temporary fix whilst I determine how to correct the linear scaling.

VISUALS

The Gravity Sandbox can be utilized to examine either realistic potential wells or abstract potential wells. As you can see in the figures below, the first figure is more realistic and representative of an actual potential field (i.e. several varying sized planets, stars, galaxies, etc.). The latter figure shows a more abstract potential that is less realistic but still useful for portraying the effects of gravitational potential wells on a test particle.

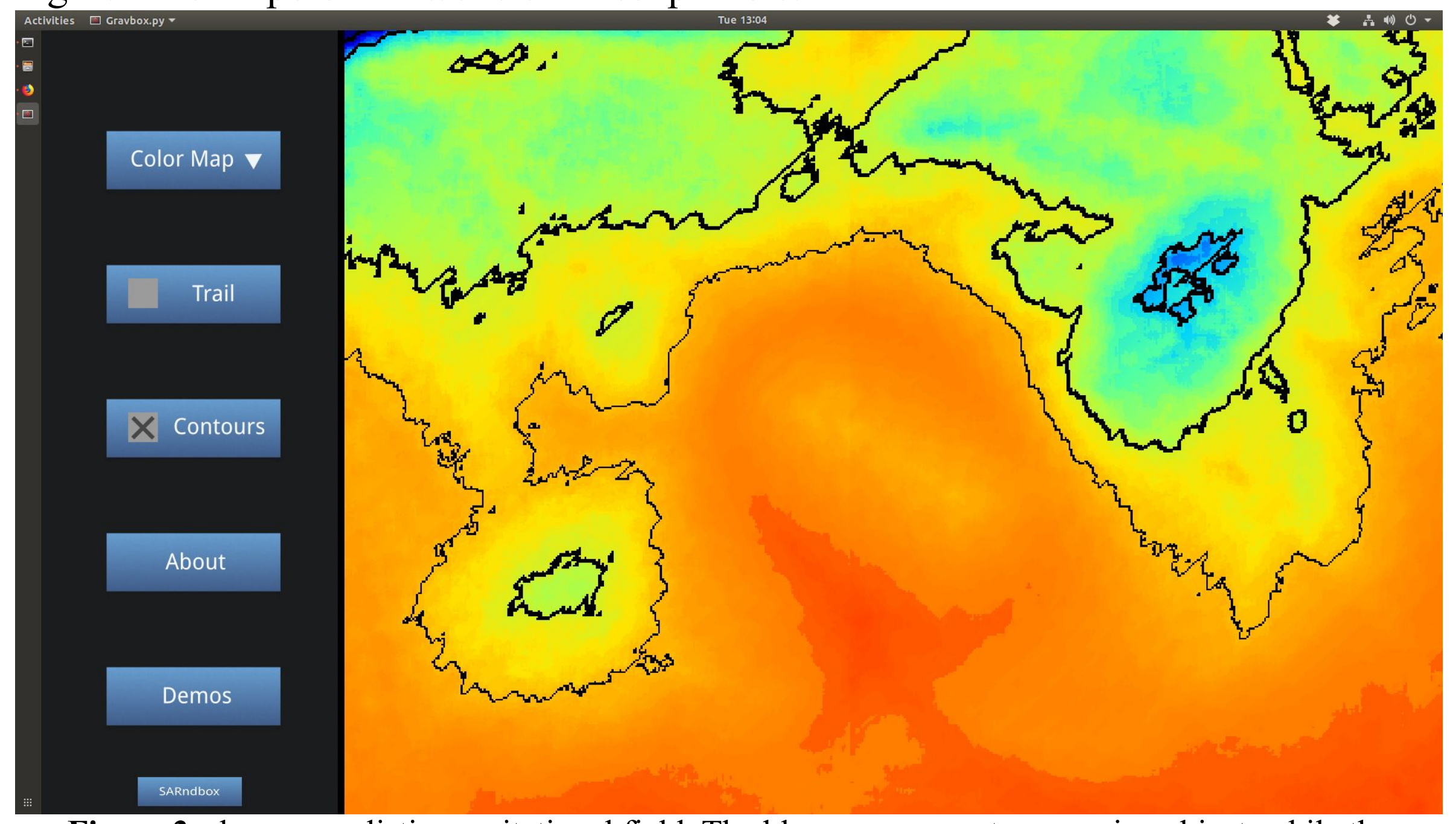


Figure 3. shows a realistic gravitational field. The blue area suggests a massive object, while the orange-red suggests absence of mass. This could be representative of a large galaxy (right, blue) with a smaller galaxy (left, green) surrounded by other groups of masses.

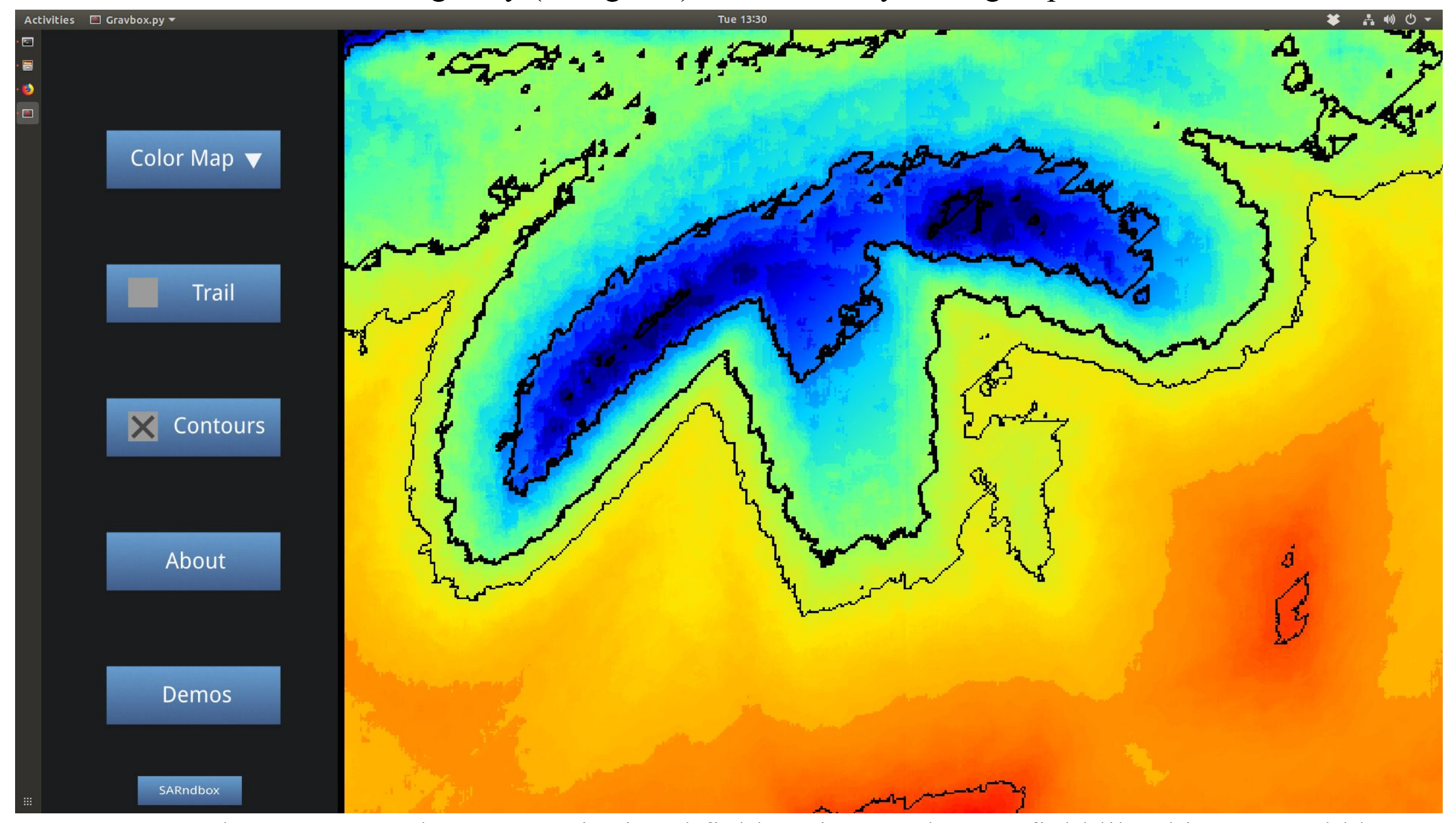
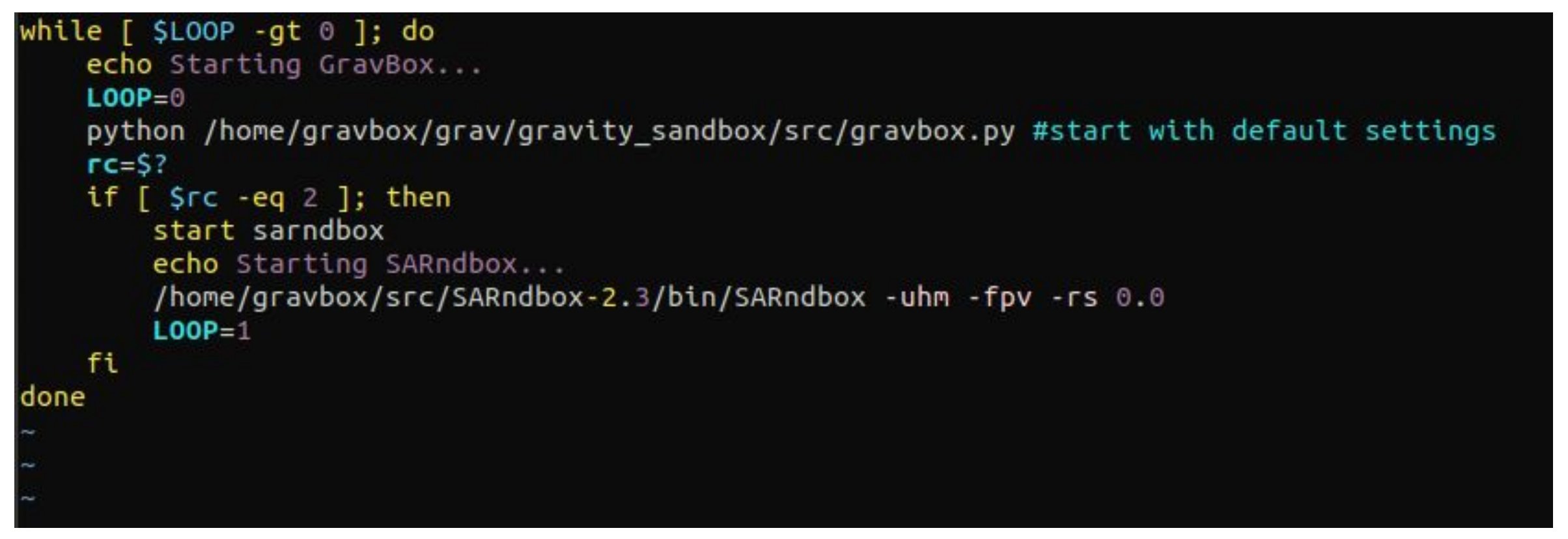


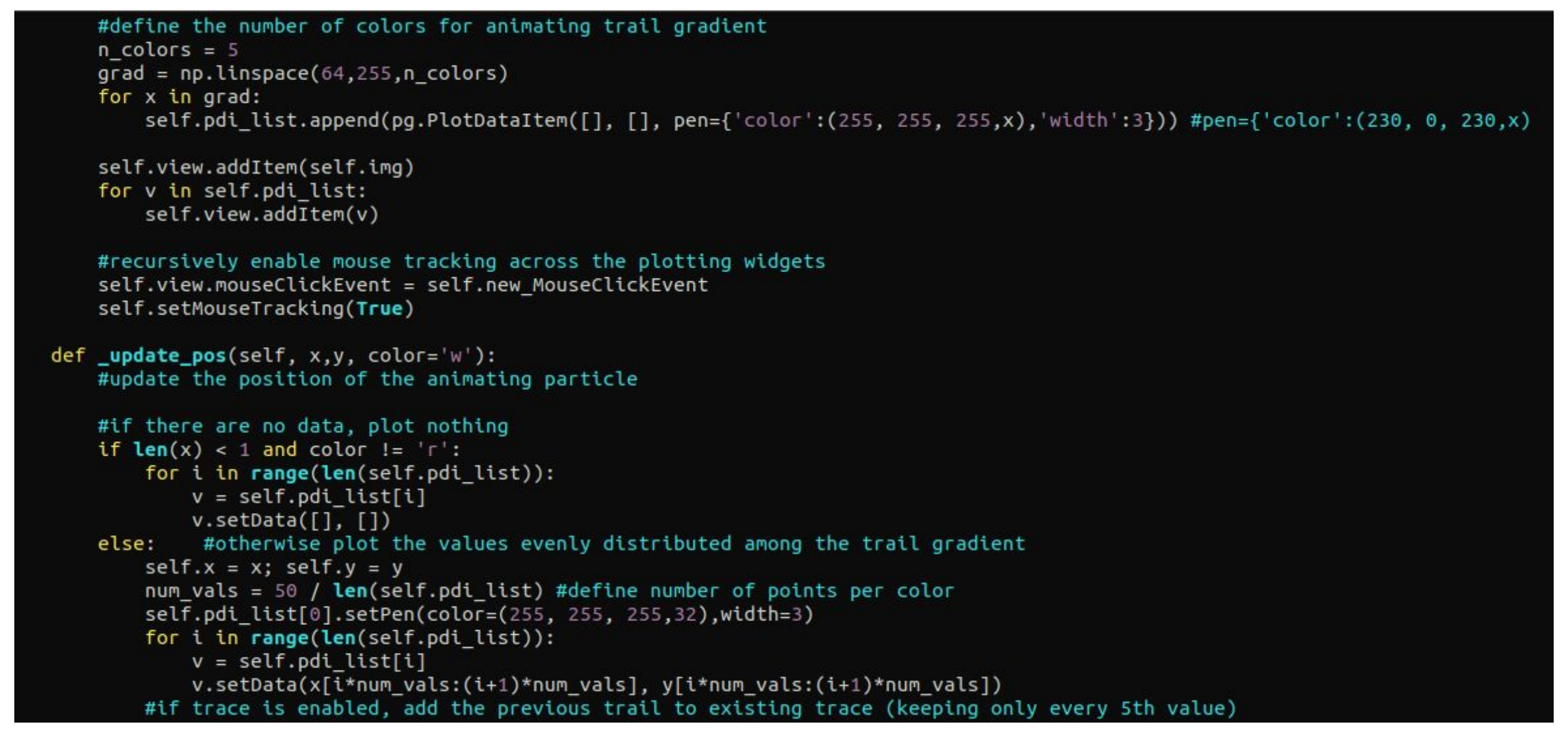
Figure 4. shows a more abstract gravitational field. Using an abstract field like this one could be useful for getting a better understanding of how gravitational potential wells interact with other, less massive objects.

VISUALS



```
while [ $LOOP -gt 0 ]; do
    echo Starting GravBox...
    LOOP=0
    python /home/gravbox/grav/gravity_sandbox/src/gravbox.py #start with default settings
    rc=$?
    if [ $rc -eq 2 ]; then
        start sarndbox
        echo Starting SARndbox...
        /home/gravbox/src/SARndbox-2.3/bin/SARndbox -uhm -fpv -rs 0.0
        LOOP=1
    fi
done
~
~
~
```

Figure 5. shows the directory named "start_sandbox.sh" where all other directories and necessary information is called from. This directory is the essence of the GravBox code.



```
        #define the number of colors for animating trail gradient
        n_colors = 5
        grad = np.linspace(64,255,n_colors)
        for x in grad:
            self.pdi_list.append(pg.PlotDataItem([], [], pen={'color':(255, 255, 255,x),'width':3})) #pen={'color':(230, 0, 230,x)

        self.view.addItem(self.img)
        for v in self.pdi_list:
            self.view.addItem(v)

        #recursively enable mouse tracking across the plotting widgets
        self.view.mouseClickEvent = self.new_MouseClickEvent
        self.setMouseTracking(True)

    def _update_pos(self, x,y, color='w'):
        #update the position of the animating particle

        #if there are no data, plot nothing
        if len(x) < 1 and color != 'r':
            for i in range(len(self.pdi_list)):
                v = self.pdi_list[i]
                v.setData([], [])
        else:    #otherwise plot the values evenly distributed among the trail gradient
            self.x = x; self.y = y
            num_vals = 50 / len(self.pdi_list) #define number of points per color
            self.pdi_list[0].setPen(color=(255, 255, 255,32),width=3)
            for i in range(len(self.pdi_list)):
                v = self.pdi_list[i]
                v.setData(x[i*num_vals:(i+1)*num_vals], y[i*num_vals:(i+1)*num_vals])
            #if trace is enabled, add the previous trail to existing trace (keeping only every 5th value)
```

Figure 6. shows a small section of the directory named "gravbox.py". This directory has the most information of all; the specific section of the code above shows where the virtual particle is created and given attributes. My code does not seem to show this virtual particle, but I am working on finding the error in order to have a seamlessly working GravBox.

CONCLUSION

Although not yet perfected, the Gravity Sandbox has proven to be a very successful learning tool and has become quite popular around the Physics and Astronomy Department. This sandbox currently has the capability of creating augmented reality gravitational potential wells (not in real time, but I plan to achieve this hastily) as well as the ability to show environmental science simulations such as rivers, mountains, water flow, etc. Using these files, we hope to be able to simulate other science based phenomenon such as those occurring in electricity and magnetism, and quantum mechanics. Quantum mechanical behaviors are especially difficult to visualize and this augmented reality tool would prove to be extremely useful if done successfully.

ACKNOWLEDGMENTS

I would like to thank the team at University of Iowa whom first conducted this research; their outline for reproducing this research was extremely useful and made the process much easier to understand. I would especially like to thank Jacob Isbell for his expertise with the GravBox. I would also like to thank all friends and family for their support throughout the process, as well as Dr.'s Galovich, Kuehn, Semak, and Willis from the University of Northern Colorado for their help and expertise throughout the project.

REFERENCES

Design, materials, and source code
Isbell, Jacob, et al. "GRAVBOX." *An Augmented Reality Sandbox for Gravitational Dynamics*, The University of Iowa, Department of Physics and Astronomy, astro.physics.uiowa.edu/gravbox/team.html.
Physics
N/A. "Gravity Fields and Potentials." *Physics A-Level*, www.physbot.co.uk/gravity-fields-and-potentials.html.
Pictures
N/A. "A Gravity Question." *Physics Stack Exchange*, Stack Exchange Inc., 13 Apr. 2017, physics.stackexchange.com/questions/231024/a-gravity-question.