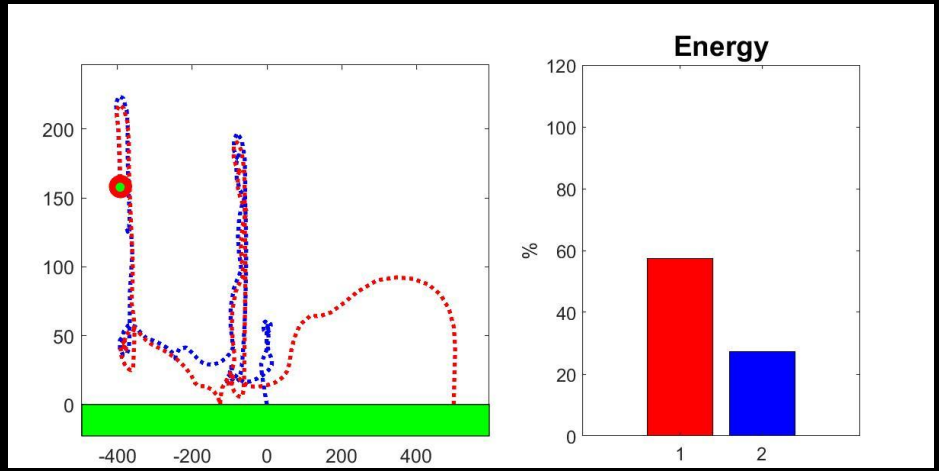
Project 2: Predator-Prey

Group 21: Dhaamin Buford, Caiden Puma, Justin Moustouka, Brian Delgado



Predator Strategy

- Predict the motion of the prey using the position and velocity of the prey

```
dt = 5;
predictdiry = ((py+vy*dt)-(pr+vr*dt))/norm((py+vy*dt)-(pr+vr*dt));
```

- Otherwise, includes ground avoidance to keep from crashing

```
if (amiapredator)
    if (t<5)
        F = Frmax*[0;1];
    else
       if (Er<150000)
        if (pr(2)>h_crit_r)
            F = 0*[0;1];
            if (pr(2) == h_crit_r)
                F = Frmax*[0;1];
            end
        elseif (pr(2)<h_crit_r)
            F = Frmax*(vr/(-1*norm(vr+0.25))); %Force opposes current motion in i and j
            if(pr(2) == h_crit_r)
                F = [0;0];
            end
    if (norm(pr-py)<10)
        dt = 3;
    end
        end
       else
           F = Frmax*predictdiry + (mr*g + 1000/(pr(2)+0.25))*[0;1];
       end
    end
```

Prey Strategy

- Dependent on how far the predator is

```
if (norm(py-pr)<50)
    F = Fymax*((py+vy*cos(t))-(pr+vr*sin(t)))/norm((py+vy*cos(t))-(pr+vr*sin(t))) + (my*g + 1000/(py(2)+0.25))*[0;1];
  if (norm(py-pr)<15)
      F = Fymax/sqrt(2)*cos(t)*[1;0] + (Fymax/sqrt(2)*sin(t) + my*g + 1000/(py(2)+0.25))*[0;1];
  end
end
```

- As predator gets close
  - The prey will move in the direction of the vector that points away from the predator
  - Variation in movement so that motion is less predictable
- Otherwise
  - Moves with variations in movement in both the x and y direction

Ground Avoidance

Acting Force

- For Predator
  - F = Frmax*predictdiry + (mr*g + **1000/(pr(2)+0.25**))*[0;1]
- For Prey
  - F=Fymax/sqrt(2)*cos(t)*[1;0]+(Fymax/sqrt(2)*sin(t)+my*g+ **1000/(py(2)+0.25**))*[0;1];]

Bold terms increase as altitude decreases

- Vertical force grows to push particles away from ground.
- Large magnitude ensures force is great enough to repel particles.
- A constant 0.25 was added to the denominator to ensure that the function has real outputs, otherwise the vertical force would grow to infinity.

Refueling

- **Calculating Critical Height for Predator and Prey**
  - Work-Energy Relation
    - Work = dE
    - Work $= T_1 + U_1 - (T_0 + U_0)$
  - Apply Work-Energy Relation
    - Work = Fmax * d
    - $T_1 = 0, U_1 = 0$ ← On ground
    - $T_0 = - \frac{1}{2} mV^2, U_0 = mgh$
      - Fmax * d $= \frac{1}{2} mV^2 + mgh$
  - Solve Equation for h
    - h (Fmax * d - mg) $= \frac{1}{2} mV^2$
    - h $= \frac{1}{2} mV^2$ / (Fmax * d - mg)
  - Adjustment for Random Force
    - h $= 1.5 * (\frac{1}{2} mV^2$ / (Fmax * d - mg))

```
h_crit_r = 1.5*(0.5*mr*norm(vr)^2)/(Frmax-(mr*g));
```

```
h_crit_y = 1.5*(0.5*my*norm(vy)^2)/(Fymax-(my*g));
```

Refueling

- **Predator**
  - Refueling begins when remaining fuel goes below 150,000
  - If above critical height, free fall, F = [0;0]
  - When at critical height, max force in + j (upwards)
  - If below critical height, force directly opposing current velocity (both i, j directions)
- **Prey**
  - Refueling begins when remaining fuel goes below 14000
  - ** If above critical height, max applied force downward until crit height
  - When at crit height, max force upward
  - If below crit height force, oppose motion

```
    if (Er<150000)
     if (pr(2)>h_crit_r)
         F = 0*[0;1];
         if (pr(2) == h_crit_r)
             F = Frmax*[0;1];
         end
     elseif (pr(2)<h_crit_r)
         F = Frmax*(vr/(-1*norm(vr+0.25))); %Force opposes current motion in i and j
         if(pr(2) == h_crit_r)
             F = [0;0];
         end
 if (norm(pr-py)<10)
     dt = 3;
 end
     end
```

```
 if (Ey<14000)
  if (py(2)>h_crit_y)
      F = Fymax/2*[0;-1];
      if (py(2) == h_crit_y)
          F = Fymax*[0;1];
      end
  elseif (py(2)<h_crit_y)
      F = Fymax*(vy/(-1*norm(vy+0.25))); %Force opposes current motion in i and j
      if (py(2) == h_crit_y)
          F = [0;0];
      end
 end
```