The Endless Square? [SOLUTION]

Code with loop and Wait statements

```ada
with Lego;
use Lego;

procedure Turnrover is
  Left_Wheel : constant Output_Port := Output_A;
  Right_Wheel: constant Output_Port := Output_C;
  Left_Rot   : constant Sensor_Port := Sensor_1;
  Right_Rot  : constant Sensor_Port := Sensor_3;

begin
  Config_Sensor(
    Sensor => Left_Rot,
    Config => Config_Rotation);
  Config_Sensor(
    Sensor => Right_Rot,
    Config => Config_Rotation);
    Output_Power(
    Output => Left_Wheel,
    Power  => Power_High);

  Output_Power(
    Output => Right_Wheel,
    Power  => Power_High);
  Output_Power(
    Output => Left_Wheel,
    Power  => Power_High);

  Clear_Sensor(Left_Rot);
  Clear_Sensor(Right_Rot);

  loop
    Output_On_Reverse(Left_Wheel);
    Output_On_Reverse(Right_Wheel);
    Wait(1000); -- Timing may have to be adjusted

    Output_On_Forward(Left_Wheel);
    Output_On_Reverse(Right_Wheel);
    Wait(500); -- Timing may have to be adjusted
  end loop;

end Turnrover;
```

Use of enclosing loop (1 pt)

Motors turn in same direction (1 pt)

Motors turn in opp. direction (1 pt)

Code with loops, exit when, and rotation sensors.

There are many ways to implement this, here is one way. Replace the loop in the code above with this loop:

```
loop

  loop
    Output_On_Reverse(Left_Wheel);       ⎤⎯ Motors turn in same direction (1 pt)
    Output_On_Reverse(Right_Wheel);      ⎦
    exit when (Get_Sensor_Value(Left_Rot)<-500); -- Timing may have to be adjusted  ⎤⎯
  end loop;
                                         Use of conditional (if or exit statement) (1 pt)
                                         Use of "Get_Sensor_Value" (1 pt)
  Clear_Sensor(Left_Rot);
  Clear_Sensor(Right_Rot);
                                      ⎤⎯ Use of nested loop (1 pt)
  loop                                ⎦
    Output_On_Forward(Left_Wheel);    ⎤⎯ Motors turn in opp. direction (1 pt)
    Output_On_Reverse(Right_Wheel);   ⎦
    exit when (Get_Sensor_Value(Left_Rot)<300); -- Timing may have to be adjusted
  end loop;

  Clear_Sensor(Left_Rot);      ⎤⎯ Clear Sensors (1 pt)
  Clear_Sensor(Right_Rot);     ⎦

end loop;
```

Pre/Post Conditions, Inputs & Outputs

Precondition: Rover placed on the ground with certain orientation. **(1 pt)**

Postcondition: Rover will continuously drive in a square like shape. **(1 pt)**

Inputs*: Rotation Sensor Values from port 1 and port 2. **(1 pt)**

Outputs*: Motor Power to ports A and B. **(1 pt)**

* Since this is a physical system, the inputs and outputs here have been described as measurements from the world. In a more typical program that only runs on a computer, the inputs would be the parameters passed into the procedure and the outputs are the values returned. You will be learning about parameters for procedures and return functions soon.