Planning Your Mission to Mars, the Red Planet

[Part 3]

David J. Ritchie, Tim Winder, Craig Weber

[Note]

This was independently written to work with the Mars Simulation unit available from Interact, 5937 Darwin Court, Suite 106, Carslbad, CA 92008.
Phone: (800)359-0961 Web: http://www.teachinteract.com/
The Mars Simulation is a copyrighted product of Interact.
They have not reviewed or endorsed this document.

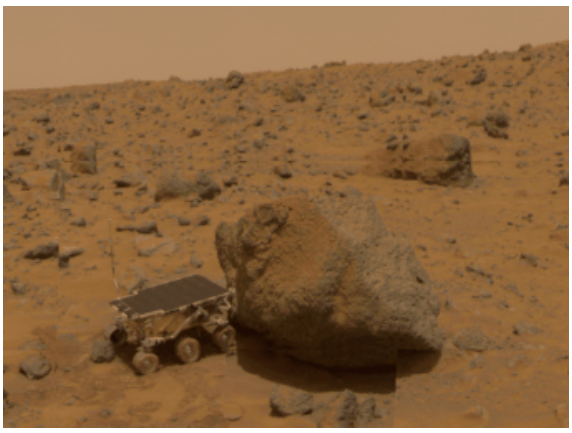
Mars Mission Science Camp
School District 203, Naperville, IL

July 15, 1998

Summary

This is the third of five sessions in which you will use the computer programming language "perl" to:

plan your mission to Mars, the red planet.



---

http://www.jpl.nasa.gov/files/images/browse/81958b.gif

1.0 Introduction

On Monday you learned the basic elements of perl using a series of examples. We went on to do some of Tuesday's examples which explored how we could specify information related to the Mars simulation in Perl variables, such as:

```
$DailyFood =  2.55;
$TotalFood = 20.00;
```

We will continue to look at Tuesday's examples, including how we can simulate the consumption of food by calling a perl subroutine for each day of the simulation to "consume" the food:

```
foreach $Day (1..10) {
     EatFood();
}
```

Specifying information in variables and calling perl subroutines to simulate the Mars mission will be our basic tools for Wednesday and the remainder of the week.

2.0 Schedule

The basic schedule is that in each day's computer session we put into the simulation what we learn about the mission from the day before. So, the week looks like this:

Day	Computer Session	Class Session
Monday:	Learn Basics of Perl	Human Factors: Food, Water, Oxygen
Tuesday:	Add human factors to simulation Run the simulation.	Perils of the Journey: Microgravity versus Artificial Gravity. Exercise. Solar Flares.
Wednesday:	Add perils to simulation. Run the simulation.	Life on Mars. Where to land.
Thursday:	Add landing site to simulation	Trajectory: Direct vs Slingshot around Venus
Friday:	Add trajectory to simulation Run the simulation.	

As your perl consultant, I will make a first pass at adding the information to the simulation from the day before. You as the mission teams will run the simulation, and see if it makes sense. Your role is **crucial.** These days people often times believe something **just because it comes out of a computer. Do not make this mistake. Check the simulation with your own calculations. As members of the mission team, your getting back to the *Green Hills of Earth* may depend on it!**

I will help you modify your perl simulation if you think it is needed. Remember that we need to keep things simple!

3.0 The Simulation Story

We need a simulation story to guide us. The story describes the facts we are going to take into account in our simulation and how we are going to do it. We will add to the story through the week.

3.1 The Human Factors Chapter

The mission lasts for 31 months. We will treat months as having 30 days each. There are four crew members with the numbers of men and women to be specified as input into the simulation.

The daily food, drinking water, and oxygen requirements for each woman crew member is 2.55 lbs, 4.25 lbs, and 1.70 lbs, respectively. The daily food, drinking water, and oxygen requirements for each man crew member is 3 lbs, 5 lbs, and 2 lbs, respectively.

We begin with a total amount of food, drinking water, and oxygen. Each day we reduce the total amount of food, drinking water, and oxygen. If we are recycling everything, there is nothing more to say.

If we are discarding waste, then the space craft is lightened by the amount used.

If we are in suspended animation mode, then we assume that the food, water and oxygen requirement is only 30% of what it is in other modes.

4.0 The Simulation Dictionary

This is a list of the variables in the simulation and what they specify.

4.1 Mission Parameters

$MisMonths..... The length of the mission in months.
$MisDays..... The length of the mission in days.
$Day..... The particular day of the mission.

4.2 People

$SciName..... The name of the Scientist

$SciMW..... ="M" if scientist is a man; = "W" if scientist is a woman

$SciH..... The health of the scientist ranging from 100 (good) to 0 (not good!)

$MedName..... The name of the Medical Officer

$MedMW..... ="M" if Medical Officer is a man; = "W" if Medical Officer is a woman

$MedH..... The health of the scientist ranging from 100 (good) to 0 (not good!)

$MilName..... The name of the Military Officer

$MilMW..... ="M" if Military Officer is a man; = "W" if Military Officer is a woman

$MilH..... The health of the scientist ranging from 100 (good) to 0 (not good!)

$MisName..... The name of the Mission Commander

$MisMW..... ="M" if Mission Commander is a man; = "W" if Mission Commander is a woman

$MisH..... The health of the scientist ranging from 100 (good) to 0 (not good!)

4.3 Human Factors

`$TotalFood.....` The total food on the mission

`$TotalWater.....` The total water on the mission

`$TotalOxygen.....` The total oxygen on the mission

`$DailyFoodW.....` The daily food consumed by the average woman

`$DailyFoodM.....` The daily food consumed by the average man

`$DailyWaterW.....` The daily water consumed by the average woman

`$DailyWaterM.....` The daily water consumed by the average man

`$DailyOxygenW.....` The daily oxygen consumed by the average woman

`$DailyOxygenM.....` The daily oxygen consumed by the average man

`$HFMode.....` The mode of handling the human factors during the mission: R - recycle, D - discard, S - suspended animation

`$HFRisk.....` The risk associated with the mode of handling the human factors between 0 and 100. For Recycle and Discard, HFRisk is 0 meaning no risk. For Suspended Animation, HFRisk reduces the health of the person in suspended animation upon being woken up.

5.0 Human Factors Simulation

```
#    Example p3s50e1.pl
#
#    Enter the names of the people on your mission,
#    whether man or woman, and health (all 100%)
     $SciName = "Mary Smith";
     $SciMW   = "F";
     $SciH    = 100;
     $MedName = "James Roberts";
     $MedMW   = "M";
     $MedH    = 100;
     $MilName = "Susan Limon";
     $MilMW   = "W";
     $MilH    = 100;
     $MisName = "Kenneth Bates";
     $MisMW   = "M";
     $MisH    = 100;
#
#    fill in total food, water and oxygen with your numbers
#    for 31 month mission
     $TotalFood    = 20.0;
     $TotalWater   = 40.0;
     $TotalOxygen  = 20.0;
#
#    say how much a woman crew member needs each day
     $DailyFoodW   =   2.55;
     $DailyWaterW  =   4.25;
     $DailyOxygenW =   1.70;
#
#    say how much a man crew member needs each day
     $DailyFoodM   =   3.00;
     $DailyWaterM  =   5.00;
     $DailyOxygenM =   2.00;
```

```
#
#    say we are beginning the Human Factors Simulation
     print ("Beginning Human Factors Simulation\n");
     print ("Mission Commander $MisName\n");
     print ("Military Officer  $MilName\n");
     print ("Medical Officer   $MedName\n");
     print ("Scientist         $SciName\n");
#
#    The number of months in the mission and days
#    assuming 30 days per month
     $MisMonths = 31;
     $MisDays   = $MisMonths * 30;

#
#    for each day of the mission
#    pretend to eat, drink and breathe
     foreach $Day (1..$MisDays) {
         EatFood();
         DrinkWater();
         BreatheOxygen();
         if ($Day%30 == 0) { print ("At day $Day\n"); }
     }
#
#    print food, water and oxygen at the end of mission
     print ("After $MisDays days, amounts left are:\n",
            "Food: $TotalFood\n",
            "Water: $TotalWater\n",
            "Oxygen: $TotalOxygen\n");
     exit;
```

```
#    Subroutine EatFood
sub  EatFood {
#
#    Get total daily food according to man or woman
     $DailyFood =
     $SciMW eq "M" ? $DailyFoodM : $DailyFoodW +
     $MedMW eq "M" ? $DailyFoodM : $DailyFoodW +
     $MilMW eq "M" ? $DailyFoodM : $DailyFoodW +
     $MisMW eq "M" ? $DailyFoodM : $DailyFoodW;

     $TotalFood = $TotalFood - $DailyFood;
#
#    if we have eaten all the food, print out day and exit
     if ($TotalFood <= 0 ) {
         print ("Day $Day:  No food left!\a\n");
         exit;
     }
     return;
}
#    Subroutine DrinkWater
sub  DrinkWater {
#
#    Get total daily water according to man or woman
     $DailyWater =
     $SciMW eq "M" ? $DailyWaterM : $DailyWaterW +
     $MedMW eq "M" ? $DailyWaterM : $DailyWaterW +
     $MilMW eq "M" ? $DailyWaterM : $DailyWaterW +
     $MisMW eq "M" ? $DailyWaterM : $DailyWaterW;

     $TotalWater = $TotalWater - $DailyWater;
#
#    if we have drunk all the water, print out day and exit
     if ($TotalWater <= 0 ) {
         print ("Day $Day:  No water left!\a\n");
         exit;
     }
     return;
}
```

```
#    Subroutine BreatheOxygen
sub  BreatheOxygen {
#
#    Get total daily oxygen according to man or woman
     $DailyOxygen =
     $SciMW eq "M" ? $DailyOxygenM : $DailyOxygenW +
     $MedMW eq "M" ? $DailyOxygenM : $DailyOxygenW +
     $MilMW eq "M" ? $DailyOxygenM : $DailyOxygenW +
     $MisMW eq "M" ? $DailyOxygenM : $DailyOxygenW;

     $TotalOxygen = $TotalOxygen - $DailyOxygen;
#
#    if we have breathed the oxygen, print out day and exit
     if ($TotalOxygen <= 0 ) {
         print ("Day $Day:  No oxygen left!\a\n");
         exit;
     }
     return;
}
```