

Planning Your Mission to Mars, the Red Planet

[Part 4]

David J. Ritchie, Tim Winder, Craig Weber

©1998 by David J. Ritchie

[Note]

This was independently written to work with the Mars Simulation unit available from Interact, 5937 Darwin Court, Suite 106, Carlsbad, 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 16, 1998

Summary

This is the fourth 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 Schedule - Where Are We in Our Mars Simulation?

The following table shows what we have accomplished (x) and what we are about to do (>):

<u>Day</u>	<u>• Computer Session</u>	<u>• Class Session</u>
Monday:	x Learn Basics of Perl	x Human Factors: Food, Water, Oxygen
Tuesday:	x Add human factors to simulation x Run the simulation.	x Perils of the Journey: Microgravity versus x Artificial Gravity. Exercise. Solar Flares.
Wednesday:	> Add perils to simulation. > Run the simulation.	— Life on Mars. Where to land. Wind, Water, — CO2, hard/soft, high/low,
Thursday:	— Add landing site to simulation	— Trajectory: Direct vs Slingshot around Venus
Friday:	— Add trajectory to simulation — Run the simulation.	

Yesterday in the class sessions, you considered the difficulties of living without gravity and the problems in supplying artificial gravity.

As your perl consultant, I have put a no-gravity/gravity option into the simulation. I have extended the Simulation Story and the Simulation Dictionary to include these new features of your upcoming Mars Mission.

2.0 Your Goals for Today and Tomorrow

I will suggest your goals for today. I would like you to decide your goals for tomorrow.

2.1 Your Goals for Today

It is time that you began using the simulation to tell you something about the mission.

There is no point to waiting until the simulation is "done." No simulation program is ever really done. You can always find some way to make it more realistic, more life-like.

So, today, I want you to accomplish the following. **Make a check mark by the number on this list when you have done them.**

1. Open and Syntax check yesterday's simulation (Example p3s50e1. pl) that you will find on the floppy.
2. Run it and look at the output briefly to make sure it makes sense.
3. Open and Syntax check **today's** simulation (Example p4s50e1. pl).
4. Run it and look at the output briefly to make sure it makes sense.
5. Change the mission names to be the full names (e.g., "David Ritchie") of the people on your team in their proper roles. Set the MW variables correctly to specify each person as a Man or a Woman.
6. Do a "Save As" to save it to your own disk directory (not the floppy but the fileserver disk). You now have your own personal copy of the Mars Mission simulation (Version: Human Factors and Gravity/No-Gravity).
7. Syntax check and run your simulation. Look at the output on the screen. Does it make sense? If so, print the Output Window (the one that has the word "MacPerl" at the top of it).
8. Get together with your team and compare the outputs from each of your simulations. Are they the same or different? Discuss with your team why they are the same or different. Try to figure out how to get them to agree. **You are all going on the same Mars Mission so your simulations had better agree or else you had better understand why they don't.** Ask for help if you need it.

2.2 Your Goals for Tomorrow

Today, think about what you want to get done tomorrow. By now, you should have calculated by hand the amount of food, water, and oxygen for a 31 month mission. Do these numbers agree with the simulation? What other ways can you use the simulation to tell you something about your upcoming trip?

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. A mission has a "sponsoring country" whose name can be specified.

There are four to six crew members with the numbers of men and women to be specified as input into the simulation. There can be one or two scientists and one or two military officers. The second scientist or military officer is on the mission provided he or she has been given a name.

The daily food, drinking water, and oxygen requirements for each woman crew member are 2.55 lbs, 4.25 lbs, and 1.70 lbs, respectively. The daily food, drinking water, and oxygen requirements for each man crew member are 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. (Not Yet Implemented).

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 (Not Yet Implemented).

Each person now has a health factor. The health factor starts at 100.

Each person also has a daily exercise counter. If this is set to 1, the person exercises every day. If set to 2, every second day, etc. Each day the person exercises 0.1 is added to their health factor. Each day the person exercises the daily food, water, and oxygen usage is multiplied by 1.05--increasing the daily food, water and oxygen consumption per exercising person by 5%.

The mission now has a gravity indicator. This indicator is "true" if the ships containing the human beings have artificial gravity. Otherwise, it is false meaning that there is no gravity. If there is no gravity, then 0.075 is subtracted from the person's health factor each day so that exercise wins a bit over the lack of gravity.

4.0 The Simulation Dictionary

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

4.1 Mission Parameters

\$Mi sMonths.	The length of the mission in months.
\$Mi sDays.	The length of the mission in days.
\$Day.	The particular day of the mission.
\$Mi sCountry.	The name of the country sponsoring the mission
\$Mi sGravi ty.	The mission gravity indicator. If true, the mission has artificial gravity. Otherwise, it has no gravity.

4.2 People

\$Sci Name.	The name of the Scientist
\$Sci MW.	= "M" if scientist is a man; = "W" if scientist is a woman
\$Sci H.	The health of the scientist beginning at 100 and increasing or decreasing as situations occur.
\$Sci EF.	The exercise frequency of the scientist: 1 => the person exercises every day, 2 => every other day, etc.
\$Sci 2Name.	The name of the second scientist. If name is "", then the scientist is not on the mission.
\$Sci 2MW.	= "M" if second scientist is a man; = "W" if second scientist is a woman
\$Sci 2H.	The health of the second scientist beginning at 100 and increasing or decreasing as situations occur.
\$Sci 2EF.	The exercise frequency of the second scientist: 1 => the person exercises every day, 2 => every other day, etc.
\$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 beginning at 100 and increasing or decreasing as situations occur.
\$MedEF.	The exercise frequency of the Medical Officer: 1 => the person exercises every day, 2 => every other day, etc.
\$Mi l Name.	The name of the Military Officer
\$Mi l MW.	= "M" if Military Officer is a man; = "W" if Military

	Officer is a woman
\$Mi l H.	The health of the Military Officer beginning at 100 and increasing or decreasing as situations occur.
\$Mi l EF.	The exercise frequency of the Military Officer: 1 => the person exercises every day, 2 => every other day, etc.
\$Mi l 2Name.	The name of the second Military Officer
\$Mi l 2MW.	= "M" if second Military Officer is a man; = "W" if second Military Officer is a woman
\$Mi l 2H.	The health of the second Military Officer beginning at 100 and increasing or decreasing as situations occur.
\$Mi l 2EF.	The exercise frequency of the second Military Officer: 1 => the person exercises every day, 2 => every other day, etc.
\$Mi sName.	The name of the Mission Commander
\$Mi sMW.	= "M" if Mission Commander is a man; = "W" if Mission Commander is a woman
\$Mi sH.	The health of the scientist beginning at 100 and increasing or decreasing as situations occur.
\$Mi sEF.	The exercise frequency of the Mission Commander: 1 => the person exercises every day, 2 => every other day, etc.

4.3 Human Factors

\$Total Food.	The total food on the mission
\$Total Water.	The total water on the mission
\$Total Oxygen.	The total oxygen on the mission
\$Dai l yFoodW.	The daily food consumed by the average woman
\$Dai l yFoodM	The daily food consumed by the average man
\$Dai l yWaterW.	The daily water consumed by the average woman
\$Dai l yWaterM	The daily water consumed by the average man
\$Dai l yOxygenW.	The daily oxygen consumed by the average woman
\$Dai l yOxygenM	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 (not yet implemented)

\$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. (**not yet implemented**)

5.0 Human Factors and Perils Simulation

```
# Example p4s50e1.pl
# People
$SciName = "Mary Smith";
$SciMW   = "W";
$SciH    = 100;
$SciEF   = 1;
$Sci2Name = "Jeff Slattery";
$Sci2MW  = "M";
$Sci2H   = 100;
$Sci2EF  = 1;
$MedName = "James Roberts";
$MedMW   = "M";
$MedH    = 100;
$MedEF   = 1;
$M1Name  = "Susan Limon";
$M1MW    = "W";
$M1H     = 100;
$M1EF    = 1;
$M12Name = "Reggie DeBates";
$M12MW   = "M";
$M12H    = 100;
$M12EF   = 5;
$M1sName = "Kenneth Bates";
$M1sMW   = "M";
$M1sH    = 100;
$M1sEF   = 1;

#
# Resources - must increase these -- just examples
$TotalFood   = 4000.0;
$TotalWater  = 4000.0;
$TotalOxygen = 4000.0;

#
# 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;

#
# some places to hold current exercise counts
$SciEFc = $SciEF;
$Sci2EFc = $Sci2EF;
$MedEFc = $MedEF;
$Mil1EFc = $Mil1EF;
$Mil2EFc = $Mil2EF;
$MilSEFc = $MilSEF;
```

```

#
#   Sponsoring Country
#   $Mi sCountry = "Brazi l";
#
#   say we are beginning the Human Factors Simulation
#   print ("Beginning Human Factors Simulation\n");
#   print ("----Mission for $Mi sCountry----\n");
#   print ("Mission Commander $Mi sName\n");
#   print ("Scientist          $Sci Name\n");
#   print ("Scientist          $Sci 2Name\n") if ($Sci 2Name);
#   print ("Military Officer $Mi lName\n");
#   print ("Military Officer $Mi l2Name\n") if ($Mi l2Name);
#   print ("Medical Officer   $MedName\n");
#
#   The number of months in the mission and days
#   assuming 30 days per month and our gravity flag
#   $Mi sMonths = 31;
#   $Mi sDays   = $Mi sMonths * 30;
#   $Mi sGravi ty = 1;
#
#   for each day of the mission
#   pretend to eat, drink and breathe
#   foreach $Day (1.. $Mi sDays) {
#       Exercise();
#       Handl eGravi ty();
#       EatFood();
#       DrinkWater();
#       BreatheOxygen();
#
#       print the day every 30 days to show we're working
#       if ($Day%30 == 0) { print ("At day $Day\n"); }
#   }

```

```

#
# print status at the end of mission
# resources left
print ("-----Status At End of Mission-----\n");
print ("      Time: $Mi sDays (days)\n");
print ("  Food Left: $TotalFood\n",
      "  Water Left: $TotalWater\n",
      "  Oxy Left:   $TotalOxygen\n");
#
# print status of people
print ("Scientist:  $Sci Name\n",
      "Health:     $Sci H\n");
print ("Scientist:  $Sci 2Name\n",
      "Health:     $Sci 2H\n") if ($Sci 2Name);
print ("Med. Off.:  $MedName\n",
      "Health:     $MedH\n");
print ("Mil. Off.:  $Mi l Name\n",
      "Health:     $Mi l H\n");
print ("Mil. Off.:  $Mi l 2Name\n",
      "Health:     $Mi l 2H\n") if ($Mi l 2Name);
print ("Mis. Cmd.:  $Mi sName\n",
      "Health:     $Mi sH\n");
exit;

```

```
##### Beginning of all our subroutines#####  
sub HandleGravity {  
#  
# if no gravity then adjust health down  
if ($MisGravity == 0) {  
    $SciH = $SciH - 0.075;  
    $Sci2H = $Sci2H - 0.075 if ($Sci2Name ne "");  
    $MedH = $MedH - 0.075;  
    $MilH = $MilH - 0.075;  
    $Mil2H = $Mil2H - 0.075 if ($Mil2Name ne "");  
    $MisH = $MisH - 0.075;  
}  
return;  
}
```

```

sub Exercise {
#
# Scientist: Count down exercise frequency
# when get to 0, add 0.1 to health and reset exer freq cnt
$SciEFc = $SciEFc - 1;
$SciH = $SciH + 0.1 if ($SciEFc == 0);
$SciEFc = $SciEF if ($SciEFc == 0);
# Second Scientist
if ($Sci2Name ne "") {
    $Sci2EFc = $Sci2EFc - 1;
    $Sci2H = $Sci2H + 0.1 if ($Sci2EFc == 0);
    $Sci2EFc = $Sci2EF if ($Sci2EFc == 0);
}
# Medical Officer
$MedEFc = $MedEFc - 1;
$MedH = $MedH + 0.1 if ($MedEFc == 0);
$MedEFc = $MedEF if ($MedEFc == 0);
# Military Officer
$MilEFc = $MilEFc - 1;
$MilH = $MilH + 0.1 if ($MilEFc == 0);
$MilEFc = $MilEF if ($MilEFc == 0);
# Second Military Officer
if ($Mil2Name ne "") {
    $Mil2EFc = $Mil2EFc - 1;
    $Mil2H = $Mil2H + 0.1 if ($Mil2EFc == 0);
    $Mil2EFc = $Mil2EF if ($Mil2EFc == 0);
}
# Mission Commander
$MisEFc = $MisEFc - 1;
$MisH = $MisH + 0.1 if ($MisEFc == 0);
$MisEFc = $MisEF if ($MisEFc == 0);
return;
}

```

```

sub EatFood {
#
# Initialize Daily Food Usage
$DailyFood = 0.0;
#
# Assume Scientist is just regularly hungry
# and then see if Scientist just finished exercising
# and increase hunger accordingly
$Hungry = 1.00;
if ($SciEF eq $SciEFc) {$Hungry = 1.05;}
if ($SciMW eq "M") { $DailyFood += $DailyFoodM*$Hungry;}
else { $DailyFood += $DailyFoodW*$Hungry;}
if (!$Sci2Name) {
$Hungry = 1.00;
if ($Sci2EF eq $Sci2EFc) {$Hungry = 1.05;}
    if ($Sci2MW eq "M") {$DailyFood +=
        $DailyFoodM*$Hungry;}
    else { $DailyFood +=
        $DailyFoodW*$Hungry;}
}
$Hungry = 1.00;
if ($MedEF eq $MedEFc) {$Hungry = 1.05;}
if ($MedMW eq "M") {$DailyFood += $DailyFoodM*$Hungry;}
else { $DailyFood += $DailyFoodW*$Hungry;}
$Hungry = 1.00;
if ($MilEF eq $MilEFc) {$Hungry = 1.05;}
if ($MilMW eq "M") {$DailyFood += $DailyFoodM*$Hungry;}
else { $DailyFood += $DailyFoodW*$Hungry;}
if (!$Mil2Name) {
    $Hungry = 1.00;
    if ($Mil2EF eq $Mil2EFc) {$Hungry = 1.05;}
    if ($Mil2MW eq "M") {$DailyFood +=
        $DailyFoodM*$Hungry;}
    else { $DailyFood +=
        $DailyFoodW*$Hungry;}
}
$Hungry = 1.00;
if ($MisEF eq $MisEFc) {$Hungry = 1.05;}
if ($MisMW eq "M") {$DailyFood += $DailyFoodM*$Hungry;}
}

```

```
else                {$DailyFood += $DailyFood*$Hungry;}
$TotalFood = $TotalFood - $DailyFood;
if ($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 = 0.0;
$Thirsty = 1.00;
if ($SciEF eq $SciEFc) {$Thirsty = 1.05;}
if ($SciMW eq "M") { $DailyWater +=
                    $DailyWaterM*$Thirsty;}
else                { $DailyWater +=
                    $DailyWaterW*$Thirsty;}

if (!$Sci2Name) {
    $Thirsty = 1.00;
    if ($Sci2EF eq $Sci2EFc) {$Thirsty = 1.05;}
    if ($Sci2MW eq "M") {$DailyWater +=
                        $DailyWaterM*$Thirsty;}
    else                { $DailyWater +=
                        $DailyWaterW*$Thirsty;}
}
$Thirsty = 1.00;
if ($MedEF eq $MedEFc) {$Thirsty = 1.05;}
if ($MedMW eq "M") {$DailyWater +=
                    $DailyWaterM*$Thirsty;}
else                { $DailyWater +=
                    $DailyWaterW*$Thirsty;}

$Thirsty = 1.00;
if ($MilEF eq $MilEFc) {$Thirsty = 1.05;}
if ($MilMW eq "M") {$DailyWater +=
                    $DailyWaterM*$Thirsty;}
else                { $DailyWater +=
                    $DailyWaterW*$Thirsty;}

if (!$Mil2Name) {
    $Thirsty = 1.00;
    if ($Mil2EF eq $Mil2EFc) {$Thirsty = 1.05;}
    if ($Mil2MW eq "M") {$DailyWater +=
                        $DailyWaterM*$Thirsty;}
    else                { $DailyWater +=
                        $DailyWaterW*$Thirsty;}
}
}

```

```

$Thirsty = 1.00;
if ($MiseF eq $MiseFc) {$Thirsty = 1.05;}
if ($MismW eq "M") {$DailyWater +=
                    $DailyWaterM*$Thirsty;}
else                {$DailyWater +=
                    $DailyWaterW*$Thirsty;}
$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 = 0.0;
$Huff = 1.00;
if ($SciEF eq $SciEFc) {$Huff = 1.05;}
if ($SciMW eq "M") { $DailyOxygen +=
                    $DailyOxygenM*$Huff;}
else                { $DailyOxygen +=
                    $DailyOxygenW*$Huff;}

if (!$Sci2Name) {
    $Huff = 1.00;
    if ($Sci2EF eq $Sci2EFc) {$Huff = 1.05;}
    if ($Sci2MW eq "M") {$DailyOxygen +=
                        $DailyOxygenM*$Huff;}
    else                {$DailyOxygen +=
                        $DailyOxygenW*$Huff;}
}
$Huff = 1.00;
if ($MedEF eq $MedEFc) {$Huff = 1.05;}
if ($MedMW eq "M") {$DailyOxygen +=
                    $DailyOxygenM*$Huff;}
else                {$DailyOxygen +=
                    $DailyOxygenW*$Huff;}

$Huff = 1.00;
if ($MilEF eq $MilEFc) {$Huff = 1.05;}
if ($MilMW eq "M") {$DailyOxygen +=
                    $DailyOxygenM*$Huff;}
else                {$DailyOxygen +=
                    $DailyOxygenW*$Huff;}

if (!$Mil2Name) {
    $Huff = 1.00;
    if ($Mil2EF eq $Mil2EFc) {$Huff = 1.05;}
    if ($Mil2MW eq "M") {$DailyOxygen +=
                        $DailyOxygenM*$Huff;}
    else                {$DailyOxygen +=
                        $DailyOxygenW*$Huff;}
}
}

```

```

$Huff = 1.00;
if ($MisEF eq $MisEFc) {$Huff = 1.05;}
if ($MisMW eq "M") {$DailyOxygen +=
                    $DailyOxygenM*$Huff;}
else                {$DailyOxygen +=
                    $DailyOxygenW*$Huff;}
$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;
}

```