



Overview

The year was 1848 and California was a peaceful untamed wilderness. There is a smattering of people – who had recently fought a war for independence. A war named after the powerful bears that prosper in the untouched wilderness.

The only attempt, to build a civilization, is in a small town called Sacramento. Under the management of John Sutter, the small town is growing at an impressive rate. So much so, that cut timber is desperate need. To solve this dilemma, John Marshall was hired to build a sawmill on the American River in nearby Coloma. The sawmill harnesses the river's power to spin massive saw. Timber is then floated downriver to an eager population.



One day, you and some fellow workers were playing a friendly game of Chuck-a-Luck. The game was going well, with everyone was in good spirits. Well, not everyone. Your friend Joe Gunchy (from the Boondocks) had lost a considerable amount of money. In a fit of frustration, Joe picked up the dice and threw them into the river.

John Marshall, watching the incident, groaned in annoyance.

"What is wrong with you coots?" He shook his head. "I'll get to retrieve'n 'em. But, y'all owe me 2-bits! And, I mean each of you."

Marshall waded into the river and began the tedious task of locating the three dice. It had taken him nearly an hour to locate the first two. But then something odd happened. He paused for a moment and stared into the depths of the river. Moments later he had snatched a jagged stone covered with mud and moss. A ray of sunlight fell upon the stone, and it sparkled. There was a metal embedded in the stone. Something yellow.... GOLD!

This ignited the blaze that became the Gold Rush! A flood of "Forty-Niners" from all over the United States (and the World), arrived to seek their fortunes in the untapped streams and rivers of the Sierra Nevada.

Some Facts About the Gold Rush

1. Before the Gold Rush, California only had a population of 157,000 people. 150,000 were native-Americans and 6,500 "Californios" (Mexican citizens living in California). The rest were settlers from the United States. California was mostly uninhabited wilderness.

To put it in perspective: that's about one person for every square mile (or if everyone was on a 640-acre plot). Sacramento State is only 305 acres!

2. Gold was discovered at Sutter's Mill by James Marshall in 1848. Wood was needed to construct buildings in Sacramento and the sawmill was built on the American River (yes, the one right next to Sac State). The Mill is located in Coloma – a short distance from here.
3. People began to arrive in Northern California in 1849. It took nearly a year to get to California. This is the origin of the term "Forty-Niner".
4. There were two routes that Forty-Niners to follow to get to California. The first was the overland California Trail which traveled through the Northern Nevada desert. Modern-day Highway 80 follows this path.

The second route was by ship. The destination was the Golden Gate (the entrance into San Francisco). That is why the bridge, that spans the "Golden Gate" is named "The Golden Gate Bridge".

5. By the mid 1850's, more than 300,000 additional people had settled.
6. The Forty-Niners were very diverse. 25% of them came from outside the United States. The rest included religious and ethnic minorities from around North America. 92% Forty-Niners were men.
7. An ounce of gold was worth over a thousand dollars. The largest single gold nugget was 160 pounds making it worth over 3 million dollars today.



Gold Rush Technology

Pan: This is the Forty-Niner's most basic, reliable, and recognizable tool. Usually made of tin or iron, the Forty-Niner would put water and soil into a pan and swirl it in a circular motion. The goal was to have the water carry the lighter particles out of the pan – leaving only heavy particles (rocks and gold).

Cradle: Also called a "rocker", the Forty-Niner would place water and soil into the top of the device and rock it left and right. As the water flowed downwards, horizontal ridges (called riffles) would capture the heavy particles.

Sluice: A sluice uses the same basic technology, as a cradle, but is far more advanced. It is essentially a wooden channel (similar to a rain gutter) with one end elevated. The sluice was connected to a continuous flow of water – usually upstream. This liberated the Forty-Niner from manually adding water (as well as the laborious rocking) and, thus, allowed them to work on several at the same time.



Forty-niners with pans and a sluice.

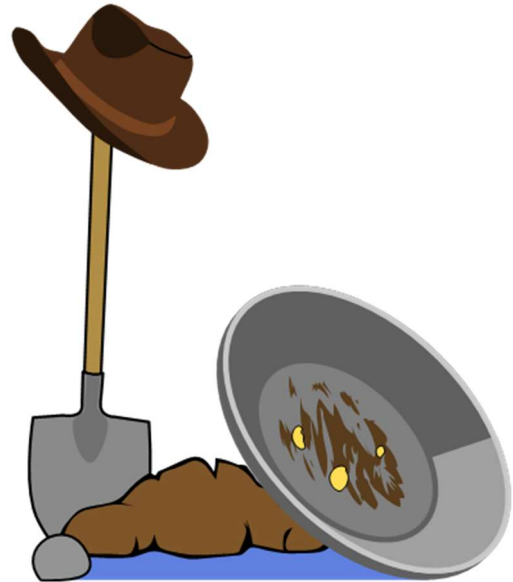
The Game

Overview

Like the classic Oregon Trail (created in 1971), your game is going to create a simple simulation based on resource management. In this style of game, you attempt to maximize your score by using your resources and obtaining assets. In your case, you will be creating a (very simple) game based on the life of Forty-Niner.

The game takes place over a 20-week period. You will start with \$100 (not bad for 1849). Your goal is to make as much money as possible. The gold found in the American River is not endless. When you stake a new claim, there will be between \$0 and \$1000 gold hidden under its pristine waters. In today's dollars, this ranges from \$0 to \$30,000.

So, each time a new claim is made, you will generate a value between 0 and 1000. You can store this in a register, but I recommend using direct storage.



Part 1: Sunday

Even though the Forty-Niners came from different parts of the World, they all followed a set of ad-hoc rules. The most important is that everyone took Sunday off. Some Forty-Niners rested, some did repairs, and others spent the day causing mischief (or being rather naughty). So, each Sunday, you get to decide:

1. Do nothing. It doesn't cost anything but doesn't help either.
2. Go into camp (which may be the towns of Auburn or Placerville) to enjoy yourself. Your endurance will go up from 50% to 75% (it might have been a great day!). However, it will cost you between \$20 and \$100.
3. Leave your claim and find another (this resets the amount of gold to a random value from 0 to 1000).

Part 2: Profits

Each week you remove gold from the "claim" you are working on. This value will range from 0 (really bad luck) to all the gold at the site. So, if *available* is the amount left, you can mine the following:

```
mined = random(available);
```

Of course, the gold that you *mined*, is removed in the process. So, the available gold decreases:

```
available -= mined;
```

This means, each week there is less gold to mine from the river. You may have a great week and removed a large percentage of the gold. On the other hand, you may have a bad week and get very little. How much gold is left? Should you stake another claim? Good questions... and real ones that you will have to make.

Part 3: Wear and Tear

Your endurance (the willpower to keep going) wears down from **10%** to **30%** each week. If it reaches zero, the game ends (you, basically, give up).

Sample Output

Your solution doesn't have to look exactly like this. However, this should show you the basic gameplay. For readability, the user's input is displayed in **red** and randomly generated values are in **green**. You don't have to use color (unless you are going for extra credit). *As always, please feel free to change the wording of the text.*

```
=====
CALIFORNIA GOLD RUSH
=====

Rules:
1. 20 weeks (5 months)
2. Your endurance drops 10% to 30% each week. If it reaches 0%, the game ends.
3. Going to camp adds 50% to 75% to your endurance, but costs $25 to $100.
4. Staking out a new claim will cost a week of work.

WEEK 1

It's Sunday!
You have $100
Your endurance is at 100%
1. Do nothing, 2. Go to camp (-$25 to -$100), 3. Stake a new claim.
1

You mined $370 in gold.
You lost 20% endurance.

WEEK 2

It's Sunday!
You have $470
Your endurance is at 80%
1. Do nothing, 2. Go to camp (-$25 to -$100), 3. Stake a new claim.
1

You mined $16 in gold.
You lost 25% endurance.

WEEK 3

It's Sunday!
You have $486
Your endurance is at 55%
1. Do nothing, 2. Go to camp (-$25 to -$100), 3. Stake a new claim.
3

You head off into the wilderness to stake a new claim. You left $15 behind.

You lost 28% endurance.
```


WEEK 4

It's Sunday!

You have \$486

Your endurance is at 27%

1. Do nothing, 2. Go to camp (-\$25 to -\$100), 3. Stake a new claim.

2

Going to town cost you \$40

You regained 59% endurance.

You mined \$305 in gold.

You lost 10% endurance.

I removed several weeks.

WEEK 20

It's Sunday!

You have \$1648

Your endurance is at 30%

1. Do nothing, 2. Go to camp (-\$50 to -\$150), 3. Stake a new claim.

1

You mined \$44 in gold.

You lost 17% endurance.

===== GAME OVER =====

You ended the game with \$1692

Your endurance was 13%

Have Fun

Use your imagination. Your game doesn't have to be the Gold Rush. You can base your game on any fun theme that you want. **But... only if you keep the same gameplay.**

For example, here are some possible scenarios:

- Kittens
- Cartoon: Spongebob Squarepants, Rick and Morty, Archer, etc....
- Politics
- Movies: comedy, sci-fi, horror, etc...
- Video games
- Television programs
- Characters from a book
- etc...

Due Date

The assignment is due the **April 22 by midnight**. I strongly suggest that you get to work on this assignment as early as possible. If you did well on your labs, it shouldn't take more than a few hours.

Requirements



YOU MUST DO YOUR OWN WORK. DO NOT ASK OTHER STUDENTS FOR HELP.

If you ask for help, both you and the student who helped you will receive a 0. Based on the severity, I might have to go to the University.

1. Print the title of your program.

2. Print the game rules. Let the player know how the game works!

3. Loop for 20 weeks.

If you change the project theme, please feel free to change this value (e.g. 12 for months)

4. Part 1: Decision

Input the player's choice. There needs to be at least 3 choices. The program must do different things based on the input.

5. Part 2: Profits

Your program must calculate how much they earned. This must be a random value of the resource (claim) left.

6. Part 2: Depletion of Resource

Each time part of the resource (claim) is taken, the amount left must be computed. Otherwise, the resource is essentially infinite.

7. Part 3: Depletion of Endurance

8. Part 3: Game Ends if endurance is depleted.

Decrease both your endurance and the sluice.

9. Comment your code!

10. Proper formatting:

Labels are never indented. Instructions are always indented the same number of spaces. Add blank lines for readability.

Extra Credit

1. Color – 5 points

Make use of color to enhance your game. The color must be meaningful – don't just set the color at the beginning of the program.

2. ASCII Art – 5 points each for a max of 10

Use ASCII-art to make your program exciting. The ASCII-art must be meaningful and not something overly simple like:

===== :)	It's a happy worm!
----------	--------------------

3. At least 2 more decisions – 10 points

There is more to do than go to town and wandering the wilderness. Give the program more decisions that can help or hinder the game.

4. Another resource – 10 points

Right now, there is only three resources – money, endurance, and the gold hidden in your claim. What else can be considered? Food? Happiness? Whiskey? Love?

5. Random events – 5 each for a max of 15.

What other types of events, good and bad, can occur. The more you add, the more you can capture the feeling of the Gold Rush.

Tips

1. Write you program in parts

DON'T attempt to write the entire program at one time. If you do, you won't be able to debug it. Experienced programmers use incremental design. Make a basic program and, very slowly, add the features you need.

So, first get the main loop working... then, bit by bit, add the rest of the functionality.

If you get stuck in an infinite loop, you can press Control+C to exit any UNIX program.

2. Random Numbers

The library has a built-in subroutine called "Random" that you must use to make your project work. Please read the documentation on how to use it.

Project Pseudocode

```

assign Endurance = 100;
assign Funds = 100;
assign Available = random number from 0 to 1000 (range of 1001) ...Between $0 to $1000
assign Week = 1;

while Week <= 20 and Endurance > 0
    output "Week ", Week

    ...Part 1: Sunday

    output "It's Sunday!"
    output "You have $", Funds
    output "Your Endurance is at ", Endurance, "%"
    output "1. Do nothing, 2. Go to camp (-$25 to -$100), 3. Stake a new claim."
    input Choice

    switch Choice
        case 2:
            ...Go to camp
            assign Cost = random number from 25 to 100 (range of 76)
            subtract Cost from Funds
            assign Gain = random number from 50 to 75 (range of 26)
            add Gain to Endurance

            output "Going to town Cost you $", Cost
            output "You regained ", Gain, "% Endurance."

            assign CanMine = true ...You can JMP to a label called CanMine:

        case 3:
            ...Stake a new claim
            output "You head off to stake a new claim. You left $", Available, " behind."

            assign Available = random number from 0 to 1000 (range of 1001)
            assign CanMine = false ...Lose a week of work

        default:
            ...Do nothing
            assign CanMine = true

    end switch

    ...Part 2: Profit

    if CanMine
        assign Mined = random number from 0 to Available

        add Mined to Funds
        subtract Mined from Available
        output "You mined $", Mined, " in gold."
    end if

    ...Part 3: Wear and Tear

    assign Loss = random number from 10 to 30 (range of 21)
    subtract Loss from Endurance
    output "You lost ", Loss, "% Endurance."

    ...Next Week
    add 1 to Week
end while

output "===== GAME OVER ====="
output "You ended the game with $", Funds
output "Your Endurance was ", Endurance, "%"

```



"Paint Your Wagon" (1969)

Connecting to the Server from Home

Step 1 – Connecting to the VPN

To get access to the server, which we will use for our labs, you must connect to the GlobalProtect VPN. There are instructions on our website.

Step 2 – For Windows

If you are using Windows, you need to download and install a copy of MobaXterm or PuTTY. Once you have installed it, open the application, and connect to the following address using SSH (Secure Shell).

```
coding1.ecs.csus.edu
```

Step 2 – For Macintosh

Open the Terminal program. This is the same UNIX prompt that you get when you connect to Athena. Mac-OS X is a version of UNIX. Neat! Once at the prompt, type the following where **username** is your ECS username. You might have to manually type "yes".

```
ssh username@coding1.ecs.csus.edu
```

Step 3 – Logging In

Once you are connected, you will be given the standard UNIX prompt:

1. Enter your username. You don't have to enter your entire e-mail address. Just enter the part before the @.
2. Enter your password. When you are on-campus, this part is normally skipped. However, when off-campus, you have to enter it. **Note: UNIX doesn't echo characters when entering a password.** You won't see anything visual on the screen, but UNIX is listening. Hit enter when done.

Download the Example Program

If you want, you can download the program that I used in-class. To do so, you need to type the following commands from the UNIX prompt. The first command should look familiar. You used it to download the CSC35 object library. In this case, you are grabbing a copy of the `goldrush.out` program.

```
curl devincook.com/csc/35/goldrush.out > goldrush.out
```

Now that you have the program in your account, you need to run one more command to make it executable. To accomplish this, you will use a command called `chmod` which can change file permissions. Type the following:

```
chmod 755 goldrush.out
```

Let's confirm that it worked. Type `ll` (that's 2 lowercase L's) and hit enter. The file should be either be displayed in green or with an asterisk. The file size should be about 14 kilobytes.

Submitting Your Project



This project may only be submitted in Intel Format. Using AT&T format will result in a zero.

To submit your lab, you must run Alpine by typing the following, and, then, enter your username and password.

```
alpine
```

Please send an e-mail to yourself (on your Outlook, Google account) to check if Alpine is working. To submit your project, send the assembly file (do not send the `a.out` or the object file). Send the `.asm` file to:

```
dcook@csus.edu
```

UNIX Commands

Editing

Action	Command	Notes
Edit File	<code>nano filename</code>	"Nano" is an easy-to-use text editor.
E-Mail	<code>alpine</code>	"Alpine" is text-based e-mail application. You will e-mail your assignments it.
Assemble File	<code>as -o object source</code>	Don't mix up the <i>object</i> and <i>source</i> fields. It will destroy your program!
Link File	<code>ld -o exe object(s)</code>	Link and create an executable file from one (or more) object files

Folder Navigation

Action	Command	Description
Change current folder	<code>cd foldername</code>	"Changes Directory"
Go to parent folder	<code>cd ..</code>	Think of it as the "back button".
Show current folder	<code>pwd</code>	Gives the current a file path
List files	<code>ls</code>	Lists the files in current directory.

File Organization

Action	Command	Description
Create folder	<code>mkdir foldername</code>	Folders are called directories in UNIX.
Copy file	<code>cp oldfile newfile</code>	Make a copy of an existing file
Move file	<code>mv filename foldername</code>	Moves a file to a destination folder
Rename file	<code>mv oldname newname</code>	Note: same command as "move".
Delete file	<code>rm filename</code>	Remove (delete) a file. There is no undo.