Main Idea: A puzzle adventure based game, using unix based commands with in-game commands to hack your way into a secure research facility and sabotage their plans.

Flow of Game

Backstory(In progress):
Using only your wits, a few hacking tools, and puzzle solving skills you will be tasked to hack into one of the securest facilities in the world, NZT, a leading research facility for the colonization of Mars. Your job after you get into their central network is to sabotage their plans and illegally launch their space shuttle, Terraferma.

Introduction text
- Slow typing speed, so the user can read
- Initiate backstory

Tutorial
- Introduce basic tools given to the hacker and notify them that there is a “help” command
- Have the player solve a basic puzzle, to give them a taste of what’s coming

Tools
- Decryptor
    - Either uses brute force or KEY, to decipher messages and unlocking secrets,(basic stuff.)
- Network map
    - Lists a bunch of networks that are connected to your computer, showcasing potential sources of INFO/places to go
- Virtual File Directory
    - Basically your file directory, that you can LS, and CD into files
- PID
    - List running programs and their associated ID number
- Code your own Virus!
    - Basic instructions + parameters to make KEYLOGGERS/MALWARE/SPYBOTS/etc...(Basically there are blanks and we let the user fill it out, but you can only make like 3 types of preset viruses lol)
- Hacker 6th Sense
    - Find exploits(will do more research, not very knowledgeable on this topic)

MISC
- Have a Pound Sign loading bar
- All puzzles + game will be timed
- Helper/Tip Bot

USEFUL articles for future reference:
https://en.wikipedia.org/wiki/Wargame_(hacking)
https://en.wikipedia.org/wiki/SQL_injection
https://en.wikipedia.org/wiki/Penetration_test
https://en.wikipedia.org/wiki/Exploit_(computer_security)
https://en.wikipedia.org/wiki/IP_address_spoofing(keep stealth)

START GAME - At least 10 minutes to digest, include all TOPICS for APCS

Decryptor
- Use Hashes with precoded messages
- Caesar cipher
- Topics/concepts
    - Conditionals
    - Loops
    - String Manipulation
    - Overloading methods(Use key, Bruteforce)

Network Map
- 2D Array, ArrayList
- 1s for aesthetics/ Names(A Random String Name) for actual sources of information = computers
- Ssh (research this more) , basically a command that finds the computer and goes into it(using a loop)
- May have Lock(password)(option to decrypt)

Virtual File Directory
- Command that lists all the files
- Using ID system, where each file system has an id, using a loop and conditionals to check for the items.

Code Your Own Virus
    Use of abstraction, to have a separate class system for Code your own virus

PID
- Simple print out of running processes
- Change processes in different computers and after different events

Classes
Woo class - main procedure of the game
Tools abstract class -
Decryptor class
Simplecommand class
NetworkManager class

Interface class
VirusCreator class
    Subset class that inherits the general blueprint for virusCreator
        KeyLogger - specific functions
        Malware - specific functions
        Spybots - specific functions

Topics: Sort Algorithms(BubbleSort, InsertionSort, SelectionSort), Standard Data Structures(primitives, arrays(2d), ArrayList, classes), Implementation(interfaces, abstract, inheritance, etc.), Scanner Methods(cs1 package(keyboard)), Loops, Conditionals, Typecasting, encapsulation, overriding/overloading methods.

Comment Records
Yes, I think your three puzzles were just stories, so i'm saying we should change it after it gets approved/rejected. To make it more about mechanics and less story. Yes there is a storyline ofc. Ill read through it lol, later.

The theme is cool, the puzzles require ingenuity and they test your terminal skills, a laugh just makes it more light hearted and interesting. Consider that although there is a storyline almost everything I described is mechanics, the player figuring stuff out. The only "sign" is the comment in the ISP's records.

Three puzzles to unlock last puzzle/one long puzzle
1st Puzzle)
Infiltrate government computer. Government is testing your hacking abilities. Tutorial. Put on same network as government computer, use network manager to identify name of computer, use decrypter to get password hash, use ssh to log in.

2nd Puzzle)
Government gives you basic info about NZT networks including ISP. You are added to the ISP's network. You use network manager to identify name of computer, use decrypter to get password hash, use ssh to log in. Same as Tutorial. But you are in a user account and you need admin privileges to gain access to the records about NZT. You sift through folders until you find a hidden one containing a text file containing the password to the admin. You then log out and reconnect using admin password and now have access to the company records.

3rd Puzzle)
Reading the company records you discover that NZT has been sending petabytes of data through one single facility on the Kaffeklubben Island in Greenland. As all NZT facilities use a lot of data this is not too strange at first glance, however, someone has commented on the same line "What in the world are they doing on an abandoned island in the arctic circle? They must be doing a lot of Netflix and chill! HA". This clues you in that something might be off here. You get the ip address from the records.

Last puzzle unlocks the mainframe
You attempt to decrypt password hash but the firewall is way out of your league, you get a weird screen saying "ACCESS DENIED /n PROJECT TERRAFERMA. You then go back to the ISP's records and try to hack into another facility (any other will work). Same routine, network manager, decrypter, ssh. You now start looking for files containing info and see that PROJECT TERRAFERMA was originally called PROJECT TREADSTONE. You try TREADSTONE as a password on the island server and you gain access to the server with admin privileges.

Then one last push to make spaceship launch.
You see a java file: Launch.java, You run it: the system finds and intrusion and it kicks you out. You Lose.

You compile it and then run it: You win

AstroJellies
Jeffrey Weng, Federico Tondolo, Allan Wang

APCS1 pd5
1/9/2017

Final Project Proposal

Shall We Play A Game?

"Shall We Play A Game?" is a puzzle adventure styled game, using unix based commands plus custom in-game commands to hack your way stealthily into a secure research facility hidden behind multiple obstacles.

Several things come into play when you start the game. Using only your wits, a few hacking tools, and puzzle solving skills you will be tasked to hack into one of the securest facilities in the world, NZT, a leading research facility for the colonization of Mars. Your job after you get into their central network is to sabotage their plans and illegally launch their space shuttle, Terraferma. You'll be completing 3 puzzles/challenges, that are hidden throughout the terminal, finding hidden clues/secrets, to find out separate pieces of an overall LOCK. Each challenge can also provide tips/hints for other challenges.

Several basic tools and abilities have been given to the Hacker to use(Not Definite):

The Decryptor:

Where decrypted blocks of text can be decoded using either a brute force method or a KEY found somewhere in the game. Most likely will be encrypting messages using a Caesar Cipher. We'll be using basic: conditionals, loops, string manipulation, overloading methods, and reading of files to achieve this goal. Basic outline:

Decrypt Message: <Enter Message> (Message can be a file or the actual message)
Brute Force(1) or Specify Key(2)? <Input>

Brute Force(Uses up time, we may insert a timed events)

- Lists all 26 possible possibilities

Key

- Prints out the message using this key

Network Manager:

Our entire game would be based around a simple map, where connected computers/hidden computers are shown. You'll probably start off in the enemy network, trying to gain access to a computer that is securely locked. The network manager will use 2D array, that prints out a list of computers that can be accessed. You'll have to insert the specific coordinate (x[rows], y[columns]) to attempt to access. It'll then loop through the 2D array. We can also use ArrayLists if possible.

Basic Outline(0s and 1s are just aesthetics, can be used effectively later on, the random string of characters are the actual computers):

0 0 0 1 0 0 1 0 0 %$# 0 1 1
0 1 1 0 #@t 1 0 0 0 0 0 0 1 (Computer #@t, located at coordinate(2, 5), (2, 6),(2, 7))
0 0 0 0 0 0 0 0 &UR 0 0 0 0
1 1 1 1 0 0 0 0 1 0 1 0 1 0 0
0 0 0 0 0 0 0 0 &UR 0 0 0 0
0 0 0 1 0 0 1 0 0 %$# 0 1 1

The Interface

More specifically, it's an in-game file directory. Nothing super special here. Files, directories, and executables, each will have an ID assigned to the file. ID will be used to find out how to open. For example: all directories will have the same ID, which basically tells us, when we press this directory, we will open up folder into another layer of the interface. Different files can be opened in different ways, in general, you won't be able to open them unless you use some kind of unix command, but basic text files can be open. Executables are the same, but they don't have text, instead they launch programs, which can be harmful or not harmful.

We'll be using basic conditionals, while loops, calls to certain classes, and lots of printing.

Virus Creator

The ability to create viruses. We'll be setting up a basic blueprint for the creation of a virus in an interface which will hold all the basic methods of a virus. Further viruses will be created, implementing the blueprint class. Viruses will be used to gain access to certain systems and used to gain information. Can be deployed into locked systems by finding vulnerabilities.

Tools

A superclass called tools, that allows, in-game commands, however complex, to inherit basic properties from superclass Tools.

MISC

- Obtaining Keyboard input from the user constantly
- Maybe a File Reader?
- Tip Bot, so you won't get lonely :), and so you can get help of course
- Sorting Algorithms can be used to help implement tools stated above
- Special Parsing
    - Look through text,images(ASCII ART), links, etc to find clues
    - Special parsing, could showcase locations of weakness in file, highlight important info, easy navigation of file

Aesthetics

Pound sign loading bar, found this using homebrew....

To progress through the game, you'll need to use the tools provided and your ingenuity/creativity. When all the puzzles have been completed, you'll be directed towards an admin computer in the facility, and be prompted to use all the information you gathered to unlock the final lock. At this point, there will be no further guidelines nor help from the Tips Bot :(, the mission is all up to you the player.

In the case that you are discovered, you'll have to pass smaller quick-time events/challenges in order to continue hacking. We'll be applying some features that'll demonstrate actual hacking to obtain needed in-game information(password hints, plot information, and unprotected, easily accessible in-game computers in the network).

This game is based around a theme similar to games such as Hacknet, Hack run, and Welcome to the game(a game about surfing the deep web), where the player spends a considerable amount of time time looking through data to find essential tools for winning. This game is also based around CTF competitions, where you need to "capture the flag" hidden in websites, images, text, etc. Finally, some further research needs to be done on "actual" hacking and other puzzle elements that we want to include in our game.

The first step in this journey would be to create the basis of the code: the tutorial. The first so-called puzzle which will introduce the player to all the tools at his disposal and the aim and storyline of the game.The beginning of the tutorial would use ASCII messages about the geniuses behind the code and the title. Once the username is selected through simple text-input the user is informed of the backstory(through print statements), and is supplied with a terminal-like interface with which the player can run terminal commands such as "help" (to explain the tools he has at his disposal) and other unix commands such as ls, cd, cat, among others. This would include multiple conditional statements considering the textual input supplied by the user. After the tutorial, the game then proceeds to the first scene, where the end goal of the game will be displayed. The tutorial is the backbone of the game, all the rest of the parts of this "car" are based around the methods and algorithms utilised in the tutorial therefore finishing the tutorial first makes it infinitely easier to complete the rest of the project.