A

* **Catch\_Phrase.txt** – Input file that works with the program in main to output a phrase to the console. This is written by the player and can be changed after each battle in the main.
* **High\_Scores.txt**- Output file that records and sorts the cumulative scores into a list of rankings. This file is updated every time a player finishes a game. Also this file is in a way also an input file as the main reads the current scores on the list before writing over it.
* **Patrons.txt** - Output file that shows a list of your defeated enemies (in game) that now support you. It is also a list of allies that the player has persuaded to his or her side.
* **Stats**.**txt** – Output file that updates itself as the player progresses through the game. This file tracks the amount of wealth, influence, power, and intuition the player has.
* **“NPC”.txt** –Output file that shows the stats of a specific opponent. This will appear when a battle is initiated**.** These files vary in names and can be confused with text files of other characters, but if a player reads the story text while running the program, the player should not run into that problem. In these generated files it will display the wealth (HP) and the three attributes of the opponent.
* **“Player”.txt** – Output file that shows the scores of previous players. This is so if there is a large quantity of players on the high score list, a person could just find the file of a player the person wants to know the score of. An example of this file would be the file called “Tom.txt”.

B

* Int choice1, choice2, choice3…. choice10, choice11; - These variables are used to take in the input value of a player and then uses it to change outcomes of the game.
* Int quiza,quizb; - These variables are the same as the choice variables. Their purpose is to be a sub-choices for choice5. So if a player chooses option two on choice 5 then the player would possibly have to make two more decisions.
* Int wealth; - This variable is essentially both the score and the hit points (HP) of the player while in game.
* Int influence; - This variable acts as a attribute that’s used as a skill, a way to calculate damage and healing for the “Sue” skill. Also it acts as a determinant for when a player tries to persuade NPC. If the player has enough influence, the player is rewarded for passing the influence check and punished if the player fails.
* Int power; - This is used to calculate the amount of damage the “Corporate sabotage” skill does. Also it also acts as a determinant in decisions in that instead of persuading, the player can instead choose to intimidate the NPC. Like influence, the player is punished or rewarded whether or not the player passes the power check in decisions.
* Int intuition; - This is used to calculate the amount of healing done with the “Self-Finance” move in the battle system. This attribute can also affect decisions made in the game with the player being rewarded or punished whether or not the player passes the intuition check.
* Int basestats; - This is used to set the amount of points the player is given to assign to the three attributes at the beginning of the game.
* Boolean godmode; - This is used in the beginning of the game to see whether or not the player has entered the correct cheat code at one point in the game. If so then the player is granted bonus stats.
* String name; - This is used in certain places in the dialogue to refer to the player in the story.
* String god; - Used to record the cheat code inputted by the player.
* Int gods; - Used to store the “compareTo” result of the players inputted cheat code and the correct cheat code.
* Double wd; - Used to hold the value of wealth as a double at the end of the game. Is also used in calculating double scorez;
* Double scorez; - Used to hold the value of wd when it’s divided by one billion. This is so the score won’t go over the number limit for java when calculating the scores or returning players.
* Int entries; - Used to hold the amount of listings on the high score text file.
* Double old\_score; - Used to hold the score of a returning player from his existing file which hold the players cumulative score. This is so it can be added to the score the returning player achieved in that run thorugh.
* Int s; - Used to solve a bug while the game was in development. Please Ignore
* Int z; used as an index for the ranking of a player when updating or printing to the high score text file.
* Int i; - Used as and index for multiple arrays.

C

* int z; - used in printing the ranking of a player
* int stand; - used to tell the player where he/she stands in ranking on the high score list
* String name; - Used to refer to the player both in game and on the high score list and used to create the name of the player’s text file which holds the cumulative score of the player from previous play troughs.

D

* Capitalism.java-used to hold most of the methods used in main and gives a format for NPC stats to be stored.
  + public int wealth; - used to refer to the NPC’s HP
  + public int influence; - used to refer to the NPC’s attribute.
  + public int power; - used to refer to the NPC’s attribute.
  + public int intuition; - used to refer to the NPC’s attribute.
  + public String name; - used to refer to the NPC’s name.
  + public int tempw; - used to refer to record an NPC’s wealth before battle.
  + public Capitalism(int w, int inf, int t, int p,String name) – Constructor for Capitalism.java
  + public void write\_stats(int intu, int inf, int pow,String file,int w) throws IOException – Used to write the stats of an enemy onto a text file.
  + public String toString
  + public int battle(Capitalism person, int inf, int intu, int pow, int w) throws IOException - Used to conduct all of the battles that take place in the game
  + public void move(String move\_name1,String move\_name2,String move\_name3) – an early concept for the game, but has been discarded. Please ignore this method.
  + public void move(String move\_name1,String move\_name2,String move\_name,String n1,String n2, String n3) – an overloaded version of said above method. It was also discarded, please ignore.
  + public int getintu()
  + public int getinf()
  + public int getpow()
  + public String getname()
  + public int gettempw()
  + public int getw()
  + public void minusw(int p) – used to subtract the wealth of an object.
    - Int p; - amount subtracted/
  + public void sued(int infl) - Here the method takes away HP from the enemy by the player's "sue" skill.
    - Int infl; - used to calculate amount taken away.
  + public void add\_patron(Capitalism patron) – Used to add NPCs to the “Patron” text file.
  + public String catch\_phrase() – used to print the catch phrase after each battle onto the console.
  + public static void sort\_scores (Score[] scores) - sorts the array of recorded scores from the high score list.
  + public int search (Score target,Score[]scores) – helps compute the ranking of a player after they’ve completed a run from the game.
  + public void sab(int p) – takes away HP from the enemy by the player’s “Corporate Sabotage” skill.
    - Int p; - used to calculate the amount taken away.
* File\_test.java – used to test specific lines of code. Not relevant to the main. Please ignore.
* Score.java – used as a mold to record the data of previous players and is passed through the search and sort method as an argument in the main.
  + String name; - used to hold the name of recorded players when turned into an object.
  + Double score; - used to hold the score of recorded players when turned into an object.
  + public Score (String n,double s) – Constructor.
  + public double compareTo(Score q) – used to compare two objects in the main.
  + public String getname()
  + public double getscore()