The following file explains all you need to know about CheMaster.exe, a game programmed by Mathew Polzin; and packaged with a level creator programmed by Camden Segal, and levels created by Will Canan.

The first part of this file explains the chemical formulas this game understands.

Key

<--> reaction capable of taking place in both directions (always possible)

-s synthesis reaction | -d decomposition | -sr singe replacement | -dr double replacement

-c combustion

Level 1 reactions - reactions that have two one element reactants, or one two element reactant that decomposes.

Level 2 reactions - reactions that have one single element reactant and one molecule reactant, or one molecule reactant that decomposes into the latter.

Level 3 reactions - reactions that have two molecule reactants or products.

Atoms

Anion::O_2 (oxygen) , Cl_2 (chlorine)

Cation::H_2 (hydrogen) , Fe (iron)

Other::C (carbon)

NOTE: To keep the game tollerable for both the user, and me the exhausted programmer, only the above atoms can be used, and only the below equations can be used. The diatomic atoms that were not included are: Nitrogen, Fluorine, Iodine, Bromine.

Level 1 reactions

$2H_2$ + O_2 <--> $2H_2O$ (water) -s | -d

H_2 + O_2 <--> 2OH (hydroxide) -s | -d

C + O_2 <--> CO_2 (carbon dioxide) -s | -d

2Fe + O_2 <--> 2FeO (iron oxide) -s | -d

H_2 + Cl_2 <--> 2HCl (hydrogen chloride) -s | -d

Fe + Cl_2 <--> $FeCl_2$ (iron chloride) -s | -d

C + $2H_2$ <--> CH_4 (methane) -s | -d

Level 2 reactions

$H_2 + 2OH <--> 2H_2O$ (water) -s | -d

$Fe + H_2O <--> FeO + H_2$ (iron oxide, hydrogen) -sr

$H_2 + FeCl_2 <--> 2HCl + Fe$ (hydrogen chloride, iron) -sr

$H_2 + 2CO_2 <--> 2HCO_2$ (bicarbonate) -s | -d

Level 3 reactions

$FeCl_2 + H_2O <--> 2HCl + FeO$ (hydrogen chloride, iron oxide) -dr

$CH_4 + 2O_2 <--> CO_2 + 2H_2O$ (carbon dioxide, water) -c

Object of game

The object of the game is to get the main character from where he starts to the exit on each map (level). In order to do this you must pick up and mix chemicals in different combinations to create synthesis, decomposition, single replacement, double replacement, and combustion reactions. Use chemicals obtained from reactions or pickups to fills holes with water, rust away doors, or burn barriers down.

Background Object Description

Wall - Made of rock, cannot be breached.
Metal Door - Can be rusted away in order to pass through it.
Wood Door - Can be burnt down in order to pass through it.
Pit - Can be filled with water in order to swim past it.

Game Board

The game board is tile based. Its size is 15x9 (width x height). The reason for this interesting size is that is how many 64x64 pixel squares you can fit on an average screen with the resolution 1024x768.

How to play

The controls are as follows:

[<-],[/\],[\/],[->] - move the character.

[space] - Info on whatever is in front of the character.

[m] - mix a substance from the chemicals in your inventory. (use plus for multiple reactants, leave plus out and include only one reactant to perform a decomposition reaction)

[r] - use rust (FeO)

[w] - use water. (H_2O)

[i] - refresh the inventory (useless, most things refresh the inventory automatically.)

HINT: The combustion reaction is just like the reactions that yield water and iron oxide, but the most useful part of this reaction is the mixing process, not the products.

MAP CREATION

The following describes the map files themselves, which can be created by hand, or you can skip this section and use the map creator made by Camden Segal.

map file example:

```
2 2
5  5  5  5  5  5  5  5  5  5  5  5  5  5  5
5  1  1  1  1  1  1  1  1  1  1  1  1  1  5
5  1  1  1  5  5  5  1  1  1  1  5  5  1  5
5  1  1  1  5  1  8  1  1  1  1  1  5  1  5
5  1  1  1  5  5  5  1  1  1  1  5  5  1  5
5  1  1  1  1  1  1  1  1  1  1  5  1  1  5
5  1  5  5  5  5  5  5  5  5  5  5  5  1  5
5  1  1  1  1  1  1  1  1  2  1  5  1  1  5
5  5  5  5  5  5  5  5  5  5  9  5  5  5  5

0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  1H 0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  1H 0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  1O 0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
```

Format

player-x player-y (sperated by space)

5 5 5... (15 numbers representing tiles that are explained later. seperate with space.)

5 1 1... (another row like one above, this represents second row of map)

(place more rows here. Each map contains 9 rows in this section *not including the row with player-x and player-y*)

(spereate above rows from rows coming up with an empty row)

0 0 0... (place numbers representing items here. same number of rows and columbs as above.)

Number representation

in first section (first 9 rows):

1 - dirt tile (player can walk here)
2 - hole in ground (player can't get past this yet)
3 - metal door (player can't get past this yet)
4 - rusted metal door (player can get past this right away)
5 - rock wall (player can never get past this)
6 - hole with water in it (player can get past this right away)
7 - wood door (player cant get past this yet)
8 - burnt wood door (player can get past this right away)
9 - exit (player wins each level by reaching the exit)

in second section (second 9 rows, after an empty row):

These numbers have two parts, a number representing the number of the substance in that space, and the name of the substance.

use zero as the first part in a space that you wish to have no element at all. (if the first part of the number is zero the program will not look for a second part. If you don't put an element in a space you may type 0. rather than 0H or 0(whatever).

example: 1H (this means put 1 mole of Hydrogen in that space *this is actually H_2 because hydrogen is diatomic*) the following letters can be added to the end of a number (1-9) to represent different elements in rows and columbs of the map:

H - hydrogen (note that the program IS case sensitive. This must be H, not h.)
O - oxygen
Cl - chlorine
Fe - iron
C - carbon
H2O - water
OH - hydroxide
CO - carbon dioxide
FeO - iron oxide
HCl - hydrogen chloride
FeCl - iron chloride
CH - methane
HCO - bicarbonate

Final notes on mapmaking

remember the gole of the game is to get to the exit.

When the map is all typed up (you must use notepad) go to save as and make sure you do the following:

1. in the "save as type:" box select all files.
2. name it "level_.cgl" of course replace the _ underscore with the level number (i.e. "level1.cgl")
3. in order for the level to work with the actual game there must be a file for all of the previous levels.
4. the CheMaster.exe will always view "level1.cgl" to start out with, and you must beat that to get to level two. Keep in mind that you can always create a copy of CheMaster.exe and move it to a new folder, then put the levels you want to test in that folder and use that CheMaster.exe to test the levels. (ChemMaster.exe will only find levels if the files are in the same folder as CheMaster.exe)
5. especially remember that a lot of the reactions require more than one mole of a substance, and produce more than one mole of another substance.

________________________THATS ALL__________________FOR NOW_________

If you read this and find something doesn't make sense; or something I have done is flatout wrong tell me, because I am only human, and I am not that great at chemistry.