

To test board behaviour:

Firstly use ./constructor to start the test.

```

      | 0|-- 0--| 1|
      | 1      | 2|
      | 0      | 3|
      | BRICK   | 4|
      | 3      | 5|
      | 2|-- 3--| 3|
      | 5      | 6|
      | 1      | 7|
      | ENERGY | 8|
      | 6|-- 9--| 7| 10| 8|--10--| 9| 5| 10|--11--|11|
      | 12      | 13| 14      | 15      | 16      | 17|
      | ENERGY | 13| 14      | 15      | 16      | 17|
      | 12| 4 |13|--18--|14| 15|--19--|16| 10|17|
      | 20      | 21| 22      | 23      | 24      | 25|
      | 18|--26--|19| 11|20|--27--|21| 3|22|--28--|23|
      | 29      | 30| 31      | 32      | 33      | 34|
      | HEAT     | 8| 25|--35--|26| 2|27|--36--|28| 6|29|
      | 37      | 38| 39      | 40      | 41      | 42|
      | 30|--43--|31| 8|32|--44--|33| 12|34|--45--|35|
      | 46      | 47| 48      | 49      | 50      | 51|
      | ENERGY | 13| 47      | 48      | 49      | 50|
      | 36| 5 |37|--52--|38| 11|39|--53--|40| 4|41|
      | 54      | 55| 56      | 57      | 58      | 59|
      | 42|--60--|43| 6|44|--61--|45| 9|46|--62--|47|
      | 63      | 64| 65      | 66      |
      | GLASS    | 18| 65      | 66      |
      | 48|--67--|49| 9|50|--68--|51|
      | 69      | 70|
      | 52|--71--|53|
Builder Blue where do yo want to build a basement?

```

We will receive the following board as output. Then we input the vertex number where Blue would like to build a basement at. We input 0. Then we need to input all the vertex numbers that Red, Orange and Yellow would like to build on in order, and then in reverse order.

We input: 32 for Red:

```

Builder Red where do yo want to build a basement?
32
Builder Orange where do yo want to build a basement?

```

Then we input 33 for Orange: (notice that an error message has been caught since builders cannot build on the vertices that are adjacent to a vertex that already has residences on it):

```

Builder Orange where do yo want to build a basement?
[33
You cannot build there
Basements already exist as locations: 0 32
Builder Orange where do yo want to build a basement?

```

Now we enter a valid input 27 for O. Then we enter some valid inputs in order: 51,44,18,2,9

```

Builder Orange where do yo want to build a basement?
[27
Builder Yellow where do yo want to build a basement?
[51
the player num is 3
Builder Yellow where do yo want to build a basement?
[44
the player num is 2
Builder Orange where do yo want to build a basement?
[18
the player num is 1
Builder Red where do yo want to build a basement?
2
the player num is 0
Builder Blue where do yo want to build a basement?
9

```

Once Blue has built a basement for the second time, the updated board will automatically being printed:

```

      |BB|-- 0--| 1|
      | 1      2
      | 0      |
      | BRICK  |
      |RB|-- 3--| 3| 3 | 4|-- 4--| 5|
      | 5      6      7      8
      | 1      |
      | ENERGY |
      | 6|-- 9--| 7| 10 | 8|--10--|BB| 5 | 10|--11--|11|
      | 12      13      14      15      16      17
      | 3      |
      | ENERGY |
      |12| 4 |13|--18--|14| 4 |15|--19--|16| 10 |17|
      | 20      21      22      23      24      25
      | 6      |
      | GLASS   |
      |0B|--26--|19| 11 |20|--27--|21| 3 |22|--28--|23|
      | 29      30      31      32      33      34
      | 8      |
      | HEAT    |
      |24| 8 |25|--35--|26| 2 |0B|--36--|28| 6 |29|
      | 37      38      39      40      41      42
      | 11      |
      | ENERGY |
      |30|--43--|31| 8 |RB|--44--|33| 12 |34|--45--|35|
      | 46      47      48      49      50      51
      | 13      |
      | ENERGY |
      |36| 5 |37|--52--|38| 11 |39|--53--|40| 4 |41|
      | 54      55      56      57      58      59
      | 16      |
      | WIFI    |
      |42|--60--|43| 6 |YB|--61--|45| 9 |46|--62--|47|
      | 63      64      65      66
      | 18      |
      | GLASS   |
      |48|--67--|49| 9 |50|--68--|YB|
      | 69      70
      | 52|--71--|53|

```

It tells us now it's blue's turn.

As this time, we input the command help to see which command we can choose to use:

```

Builder Blue's turn.
[help
~ load : changes current builder's dice type to 'loaded'
~ fair : changes current builder's dice type to 'fair'
~ roll : rolls the dice and distributes resources.
~ status : prints the current status of all builders in order from builder 0 to 3.
~ help : prints out the list of commands.

```

Now if we input an invalid command (a command that is not in help list, here we use build-road command), we'll get an error message:

```

[build-road
Invalid command.
Please enter 'help' for a list of valid commands.

```

We now input a valid command: load (this is the type of the dice; if we do not input load and input roll directly, the type of dice would automatically being set to load)

```

[load
Builder Blue now has loaded Dice

```

Now we input roll:

```

[roll
Input a roll between 2 and 12

```

We need to input a roll between 2 and 12 as said before. Notice that here we need to input the roll number because this is a load dice; if we choose to use a fair dice, then the roll number will be randomly chosen from 2 to 12. Here if we input a roll that is not between 2 and 12: (Eg: we input 1)

Input a roll between 2 and 12

1

Input a roll between 2 and 12

We need to input another valid roll again, and the invalid roll number has been automatically thrown. Now we input 4:

Input a roll between 2 and 12

4

The number you rolled is 4

Builder Orange gained:

1 ENERGY

Enter a command:

```
// builder orange gained 1 energy because
```

Next we enter help to see which commands we can use next:

```
[help
~ board : prints the current board.
~ status : prints the current status of all builders in order from builder 0 to 3.
~ residences : prints the residences the current builder has currently completed.
~ build-road <road#> : attempts to builds the road at <road#>.
~ build-res <housing#> : attempts to builds a basement at <housing#>.
~ improve <housing#> : attempts to improve the residence at <housing#>.
~ trade <colour> <give> <take> : attempts to trade with builder <colour>, giving one resource of type <give> and receiving one resource of type <take>.
~ next : passes control onto the next builder in the game.
~ save <file> : saves the current game state to <file>.
~ help : prints out the list of commands.
```

We use board command:

```
[board
      |BB|-- 0--| 1|
      |  |  |  |
      |  | 0  | 2|
      |  |BRICK|  |
      |RB|-- 3--| 3| 3 | 4|-- 4--| 5|
      |  |  |  |  |
      |  | 5  | 6  | 7  | 8  |
      |  |ENERGY|  |HEAT|  |
| 6|-- 9--| 7| 10 | 8|--10--|BB| 5 |10|--11--|11|
|  |  |  |  |  |  |
|12| 3 |13|  |14| 4 |15|  |16| 5 |17|
|ENERGY|  |PARK|  |HEAT|  |
|12| 4 |13|--18--|14|  |15|--19--|16| 10 |17|
|  |  |  |  |  |  |
|20|  |21| 6 |22|  |23| 7 |24|  |25|
|OB|--26--|19| 11 |20|--27--|21| 3 |22|--28--|23|
|  |  |  |  |  |  |
|29| 8 |30|  |31| 9 |32|  |33| 10 |34|
|HEAT|  |BRICK|  |BRICK|  |
|24| 8 |25|--35--|26| 2 |OB|--36--|28| 6 |29|
|  |  |  |  |  |  |
|37|  |38| 11 |39|  |40| 12 |41|  |42|
|30|--43--|31| 8 |RB|--44--|33| 12 |34|--45--|35|
|  |  |  |  |  |  |
|46| 13 |47|  |48| 14 |49|  |50| 15 |51|
|ENERGY|  |WIFI|  |GLASS|  |
|36| 5 |37|--52--|38| 11 |39|--53--|40| 4 |41|
|  |  |  |  |  |  |
|54|  |55| 16 |56|  |57| 17 |58|  |59|
|42|--60--|43| 6 |YB|--61--|45| 9 |46|--62--|47|
|  |  |  |  |  |  |
|  |63|  |64| 18 |65|  |66|
|  |GLASS|  |  |  |  |
|48|--67--|49| 9 |50|--68--|YB|
|  |  |  |  |  |  |
|  |69|  |70|  |  |  |
|52|--71--|53|
```

We then use status to see which resources each Builders have:

Enter a command:

```
[status
Builder Blue  has 2 building points, 0 BRICK, 0 ENERGY, 0 GLASS, 0 HEAT, 0 WIFI.
Builder Red   has 2 building points, 0 BRICK, 0 ENERGY, 0 GLASS, 0 HEAT, 0 WIFI.
Builder Orange has 2 building points, 0 BRICK, 1 ENERGY, 0 GLASS, 0 HEAT, 0 WIFI.
Builder Yellow has 2 building points, 0 BRICK, 0 ENERGY, 0 GLASS, 0 HEAT, 0 WIFI.
```

Then we input next command to switch the turn to the next builder, Red:

```

[next
      |BB|-- 0--| 1|
      |  |
      | 1 0 2
      |  | BRICK
      |RB|-- 3--| 3| 3 | 4|-- 4--| 5|
      |  |
      | 5 1 6 7 2 8
      |  | ENERGY | HEAT |
| 6|-- 9--| 7| 10 | 8|--10--|BB| 5 |10|--11--|11|
|  |
| 12 3 13 14 4 15 16 5 17
|  | ENERGY | PARK | HEAT |
|12| 4 |13|--18--|14| 15|--19--|16| 10 |17|
|  |
| 20 21 6 22 23 7 24 25
|  | GLASS | BRICK |
|08|--26--|19| 11 |20|--27--|21| 3 |22|--28--|23|
|  |
| 29 8 30 31 9 32 33 10 34
|  | HEAT | BRICK | BRICK |
|24| 8 |25|--35--|26| 2 |08|--36--|28| 6 |29|
|  |
| 37 38 11 39 40 12 41 42
|  | ENERGY | WIFI |
|30|--43--|31| 8 |RB|--44--|33| 12 |34|--45--|35|
|  |
| 46 13 47 48 14 49 50 15 51
|  | ENERGY | WIFI | GLASS |
|36| 5 |37|--52--|38| 11 |39|--53--|40| 4 |41|
|  |
| 54 55 16 56 57 17 58 59
|  | WIFI | GLASS |
|42|--60--|43| 6 |YB|--61--|45| 9 |46|--62--|47|
|  |
| 63 64 18 65 66
|  | GLASS |
|48|--67--|49| 9 |50|--68--|YB|
|  |
| 69 70
|  |
|52|--71--|53|

```

The updated board will be printed every time a builder has finished his turn.

Now it's Red's turn: it's similar to Blue's turn, and we need to define which type of dice we want to use. We use fair dice.

```
[fair
```

```
Builder Red now has fair Dice
```

Then we input roll command: notice that since it's a fair dice, the probability of rolling from 2 to 12 is the same, thus we don't know which number we'll get.

```
[roll
```

```
The number you rolled is 5
```

```
Builder Blue gained:
```

```
1 HEAT
```

On this trial we get 5 as the rolled number.

Next we use status to see what resources each builder owns.

```
[status
Builder Blue has 2 building points, 0 BRICK, 0 ENERGY, 0 GLASS, 1 HEAT, 0 WIFI.
Builder Red has 2 building points, 0 BRICK, 0 ENERGY, 0 GLASS, 0 HEAT, 0 WIFI.
Builder Orange has 2 building points, 0 BRICK, 1 ENERGY, 0 GLASS, 0 HEAT, 0 WIFI.
Builder Yellow has 2 building points, 0 BRICK, 0 ENERGY, 0 GLASS, 0 HEAT, 0 WIFI.
```

We now input residences to see what the residences the current builder has currently completed:

```
Enter a command:
```

```
[residences
```

```
Red has built:
```

```
2 B
```

```
32 B
```

```
Enter a command:
```

We use next command to pass to the next builder's turn (Notice that now is orange's turn):

```

Enter a command:
next

      |BB|-- 0--| 1|
      |  |  |  |
      | 0  | 2  |
      |BRICK|  |
|RB|-- 3--| 3| 3 | 4|-- 4--| 5|
  |  |  |  |  |  |
  | 5  | 1  | 6  | 7  | 2  | 8  |
  |  |  |  |  |  |  |
| 6|-- 9--| 7| 10 | 8|--10--|BB| 5 |10|--11--|11|
  |  |  |  |  |  |  |
  |12| 3  |13| 14 | 4  |15| 16 | 5  |17|
  |ENERGY|  |  |PARK|  |HEAT|  |
|12| 4  |13|--18--|14| 15|--19--|16| 10 |17|
  |  |  |  |  |  |  |
  |20| 21 | 6  |22| 23 | 7  |24| 25 |
  |GLASS|  |  |BRICK|  |  |
|08|--26--|19| 11 |20|--27--|21| 3 |22|--28--|23|
  |  |  |  |  |  |  |
  |29| 8  |30| 31 | 9  |32| 33 |10 |34 |
  |HEAT|  |  |BRICK|  |BRICK|  |
|24| 8  |25|--35--|26| 2 |08|--36--|28| 6 |29|
  |  |  |  |  |  |  |
  |37| 38 |11 |39| 40 |12 |41| 42 |
  |ENERGY|  |  |WIFI|  |  |
|30|--43--|31| 8 |RB|--44--|33| 12 |34|--45--|35|
  |  |  |  |  |  |  |
  |46| 13 |47| 48 | 14 |49| 50 |15 |51 |
  |ENERGY|  |  |WIFI|  |GLASS|  |
|36| 5  |37|--52--|38| 11 |39|--53--|40| 4 |41|
  |  |  |  |  |  |  |
  |54| 55 |16 |56| 57 |17 |58| 59 |
  |WIFI|  |  |GLASS|  |  |
|42|--60--|43| 6 |YB|--61--|45| 9 |46|--62--|47|
  |  |  |  |  |  |  |
  |63| 64 |18 |65| 66 |
  |GLASS|  |  |  |
|48|--67--|49| 9 |50|--68--|YB|
  |  |  |  |
  |69| 70 |
  |  |
|52|--71--|53|

```

The updated board has been printed again.

We use load command to make the dice be a loaded one, then roll the dice and choose the rolled number to be 3:

```

Builder Orange's turn.
[load
Builder Orange now has loaded Dice
[roll
Input a roll between 2 and 12
3
The number you rolled is 3
Builder Blue gained:
2 BRICK
Builder Orange gained:
1 BRICK

```

We use the status command:

```

status
Builder Blue has 2 building points, 2 BRICK, 0 ENERGY, 0 GLASS, 1 HEAT, 0 WIFI.
Builder Red has 2 building points, 0 BRICK, 0 ENERGY, 0 GLASS, 0 HEAT, 0 WIFI.
Builder Orange has 2 building points, 1 BRICK, 1 ENERGY, 0 GLASS, 0 HEAT, 0 WIFI.
Builder Yellow has 2 building points, 0 BRICK, 0 ENERGY, 0 GLASS, 0 HEAT, 0 WIFI.

```

Builder Orange could use the trade <colour> <give> <take> command to trade one resource that Orange has (give) with another resource with another Builder (<colour>, here B is Blue, R is Red, O is Orange, Y is Yellow) that Builder has, and the Builder that is asked for trade can decide whether to trade or not (by answering yes or no):

```

Enter a command:
[trade B ENERGY BRICK
Orange offers Blue one ENERGY for one BRICK
Does Blue accepts the offer?
[no
no trade is made

```

In this trial O wants to trade with B by giving out one ENERGY and take one BRICK, but B refuses. Thus no trade has been made. But O wants to trade again:

```

[trade B GLASS BRICK
you don't have enough GLASS

```

The trade cannot be made since O does not have any GLASS.

```

Enter a command:
[trade B ENERGY WIFI
B doesn't have enough WIFI

```

This trade again cannot be completed, since B does not have any WIFI.

```
[trade B ENERGY BRICK
Orange offers Blue one ENERGY for one BRICK
Does Blue accepts the offer?
yes
Orange gains one BRICK and loses one ENERGY
Blue gains one ENERGY and loses one BRICK
```

This time B said yes to the trade, then the trade has been completed and O loses one ENERGY and gains one BRICK while B loses one BRICK and gains one ENERGY.

```
[status
Builder Blue has 2 building points, 1 BRICK, 1 ENERGY, 0 GLASS, 1 HEAT, 0 WIFI.
Builder Red has 2 building points, 0 BRICK, 0 ENERGY, 0 GLASS, 0 HEAT, 0 WIFI.
Builder Orange has 2 building points, 2 BRICK, 0 ENERGY, 0 GLASS, 0 HEAT, 0 WIFI.
Builder Yellow has 2 building points, 0 BRICK, 0 ENERGY, 0 GLASS, 0 HEAT, 0 WIFI.
```

We can see that after the trade has completed, the status has also been updated.

```
Enter a command:
[next
      |BB|-- 0--| 1|
      |      | 1| 2|
      |      | 0| 3|
      |      | BRICK|
      |RB|-- 3--| 3| 4| 4-- 5|
      |      | 3| 4| 5|
      |      | 5| 6| 7| 8|
      |      | 1| 2| 3| 4|
      |      | ENERGY| HEAT|
      |6|-- 9--| 7| 10| 8| 10--11--|11|
      |      | 10| 11| 12| 13|
      |      | 12| 13| 14| 15| 16| 17|
      |      | ENERGY| PARK| HEAT|
      |12| 4| 13| 14| 15| 16| 17|
      |      | 13| 14| 15| 16| 17|
      |      | 20| 21| 22| 23| 24| 25|
      |      | GLASS| BRICK|
      |OB|--26--|19| 11| 20| 21| 22| 23|
      |      | 19| 20| 21| 22| 23|
      |      | 29| 30| 31| 32| 33| 34|
      |      | HEAT| BRICK| BRICK|
      |24| 8| 25| 26| 27| 28| 29|
      |      | 25| 26| 27| 28| 29|
      |      | 37| 38| 39| 40| 41| 42|
      |      | ENERGY| WIFI|
      |30|--43--|31| 8| RB|--44--|33| 12| 34| 45| 35|
      |      | 31| 32| 33| 34| 35|
      |      | 46| 47| 48| 49| 50| 51|
      |      | ENERGY| WIFI| GLASS|
      |36| 5| 37| 38| 39| 40| 41| 42|
      |      | 37| 38| 39| 40| 41| 42|
      |      | 54| 55| 56| 57| 58| 59|
      |      | WIFI| GLASS|
      |42|--60--|43| 6| YB|--61--|45| 9| 46| 62| 47|
      |      | 43| 44| 45| 46| 47|
      |      | 63| 64| 65| 66|
      |      | GLASS|
      |48|--67--|49| 9| 50| 68| YB|
      |      | 49| 50| 51| 52|
      |      | 69| 70|
      |      | 52| 71| 53|
```

Now it's Yellow's turn.

We set the dice to be a load one, roll and set the roll number to be 7.

```
Builder Yellow's turn.
load
Builder Yellow now has loaded Dice
[roll
Input a roll between 2 and 12
7
The number you rolled is 7
Choose where to place the GEESE
```

Notice that when the roll number is 7, the current player has the opportunity to place the GEESE on a tile, but the GEESE cannot be moved to some tiles that already has a GEESE on it. When the geese is placed on a tile, the builder that has residences on that tile will not receive any resources, and the current could randomly steal one resource from that builder. If the geese is placed on a tile that has no residences on it, or the geese is placed on a tile that contains a current builder's residence, nothing will happen. Moreover, if any player has total resources greater than or equal to 10, then he/she will randomly lose half of his/her resources (round down scheme).

```
Choose where to place the GEESE
[18
Builder Yellow has no builders to steal from.
```

We use next to enter the next builder's turn.

```
[next
      |BB|-- 0--| 1|
      | 1 0 2
      | 0 BRICK |
      |RB|-- 3--| 3| 3 | 4|-- 4--| 5|
      | 5 1 6 7 2 8
      | ENERGY | HEAT |
| 6|-- 9--| 7| 10 | 8|--10--|BB| 5 |10|--11--|11|
| 12 3 13 14 4 15 16 5 17
| ENERGY | PARK | HEAT |
|12| 4 |13|--18--|14| |15|--19--|16| 10 |17|
| 20 21 6 22 23 7 24 25
| GLASS | BRICK |
|OB|--26--|19| 11 |20|--27--|21| 3 |22|--28--|23|
| 29 8 30 31 9 32 33 10 34
| HEAT | BRICK | BRICK |
|24| 8 |25|--35--|26| 2 |OB|--36--|28| 6 |29|
| 37 38 11 39 40 12 41 42
| ENERGY | WIFI |
|30|--43--|31| 8 |RB|--44--|33| 12 |34|--45--|35|
| 46 13 47 48 14 49 50 15 51
| ENERGY | WIFI | GLASS |
|36| 5 |37|--52--|38| 11 |39|--53--|40| 4 |41|
| 54 55 16 56 57 17 58 59
| WIFI | GLASS |
|42|--60--|43| 6 |YB|--61--|45| 9 |46|--62--|47|
| 63 64 18 65 66
| GLASS |
|48|--67--|49| 9 |50|--68--|YB|
| GEESE |
| 69 70
| 52|--71--|53|
```

Now it's Blue's turn again. We use the fair dice to roll :

Builder Blue's turn.

[fair

Builder Blue now has fair Dice

[roll

The number you rolled is 8

Builder Red gained:

1 ENERGY

Builder Orange gained:

1 HEAT

The rolled number is 8 and two builders receive resources.

Enter a command:

[build-road 4

You do not have enough resources

Here B wants to build a road on edge 4, but he does not have enough resources. Notice that a road can only be built if an adjacent road or residence has been built by the same builder. Also, a road needs to be built with one HEAT and one WIFI resource. On this trial, the builder B does not have any WIFI resource.

Enter a command:

[build-res 39

You do not have enough resources

Here B wants to build a residence on vertex 39, but he does not have enough resources. Notice that a new residence can only be built if it's not adjacent to a vertex with an existing residence, and the builder has already built a road that is adjacent to the vertex. Building a new basement costs one BRICK, one ENERGY, one GLASS, and one WIFI. Otherwise the builder cannot build a basement.

Enter a command:

[improve 0

You do not have enough resources

Here B wants to improve its basement on vertex 0, but he does not have enough resources. A residence can only be improved if the builder has already built a residence and the residence has not been improved to a tower yet, moreover, if the residence is a basement, it costs two GLASS and three HEAT to be improved to a house; if the residence is a house, it costs three BRICK, two ENERGY, two GLASS, one WIFI and two HEAT to be improved to a tower. Otherwise the builder cannot improve his existing residence. If the build improves a vertex that does not have his existing residences (or out-of-range), then it will output invalid residence:

```
Enter a command:
[improve 60
Invalid residence.
```

Now we input the command: save t1.txt

```
Enter a command:
[save t1.txt
```

The current game state has been saved to file: t1.txt

Notice that the game automatically ends when a builder has a total of at least 10 building points. The first builder to have 10 building points is the winner. At this point, the builders are prompted : Would you like to play again?

If the answer is yes, then the game starts over from the beginning; if the answer is no, the game exists. If the player chooses to play again, the player will return to the state when the player first enters the game. For example, if a game starts with a loaded file, when the player plays again, the statues will return to the state when the file is loaded.

Notice that in the above case, we use ./constructor as the command line argument. We can also use other command line arguments, but the remaining steps does not have differences:

- 1) We use ./constructor -seed 3:


```
[x686liu@ubuntu1804-010:~/cs246/1209/projects/constructor/codeForStudents/mycode$ ./constructor --seed 3
```

```

      | 0|-- 0--| 1|
      | 1      | 2|
      | 0      | GLASS|
      | 2|-- 3--| 3| 9 | 4|-- 4--| 5|
      | 5      | 1 6 | 7 2 8|
      | 6|-- 9--| 7| 11| 8|--10--| 9| 5 |10|--11--|11|
      | 12     | 3 13| 14 4 15| 16 5 17|
      | 12| 11 |13|--18--|14| 5 |15|--19--|16| 4 |17|
      | 20     | 21 6 22| 23 7 24| 25|
      |18|--26--|19| 20|--27--|21| 10 |22|--28--|23|
      | 29     | 8 30| 31 9 32| 33 10 34|
      | 24| 2 |25|--35--|26| 6 |27|--36--|28| 8 |29|
      | 37     | 38 11 39| 40 12 41| 42|
      |30|--43--|31| 6 |32|--44--|33| 8 |34|--45--|35|
      | 46     | 13 47| 48 14 49| 50 15 51|
      |36| 3 |37|--52--|38| 10 |39|--53--|40| 9 |41|
      | 54     | 55 16 56| 57 17 58| 59|
      |42|--60--|43| 3 |44|--61--|45| 4 |46|--62--|47|
      | 63     | 64 18 65| 66|
      |48|--67--|49| 12 |50|--68--|51|
      | 69     | 70|
      |52|--71--|53|

```

Set seed sets the random number generator's seed to `xxx`. If we use the same seed number to generate a board each time, the board will always be the same.

2) We use `./constructor -load t1.txt` (t1.txt is a previous game that is saved):

```
[x686liu@ubuntu1804-010:~/cs246/1209/projects/constructor/codeForStudents/mycode$ ./constructor -load t1.txt
```

```

      |BB|-- 0--| 1|
      | 1      | 2|
      | 0      | BRICK|
      | RB|-- 3--| 3| 3 | 4|-- 4--| 5|
      | 5      | 1 6 | 7 2 8|
      | 6|-- 9--| 7| 10| 8|--10--|BB| 5 |10|--11--|11|
      | 12     | 3 13| 14 4 15| 16 5 17|
      |12| 4 |13|--18--|14| 5 |15|--19--|16| 10 |17|
      | 20     | 21 6 22| 23 7 24| 25|
      |0B|--26--|19| 11 |20|--27--|21| 3 |22|--28--|23|
      | 29     | 8 30| 31 9 32| 33 10 34|
      |24| 8 |25|--35--|26| 2 |0B|--36--|28| 6 |29|
      | 37     | 38 11 39| 40 12 41| 42|
      |30|--43--|31| 8 |RB|--44--|33| 12 |34|--45--|35|
      | 46     | 13 47| 48 14 49| 50 15 51|
      |36| 5 |37|--52--|38| 11 |39|--53--|40| 4 |41|
      | 54     | 55 16 56| 57 17 58| 59|
      |42|--60--|43| 6 |YB|--61--|45| 9 |46|--62--|47|
      | 63     | 64 18 65| 66|
      |48|--67--|49| 9 |50|--68--|YB|
      | 69     | 70|
      |52|--71--|53|

```

This is the board that was saved in t1.txt from the previous turn. We can assume that the file being loaded as a valid game state as described in the saving section. We just need the game saved in file (t1.txt).

3) We use `./constructor -board layout.txt`:

```

~
[x6861iu@ubuntu1804-010:~/cs246/1209/projects/constructor/codeForStudents/mycode$ ./constructor -board layout.txt

      | 0|-- 0--| 1|
      | 1      | 2|
      | 0      | 3|
      | 2|-- 3--| 3| 4|-- 4--| 5|
      | 5      | 6| 7      | 8|
      | 1      | 6| 2      | 8|
      | ENERGY| 10| 8|--10--| 9| 5|10|--11--|11|
      | 12      |13| 14      |15| 16      |17|
      | ENERGY| 13| 14      |15| 16      |17|
      |12| 4|13|--18--|14|15|--19--|16| 10|17|
      | 20      |21| 22      |23| 24      |25|
      |18|--26--|19| 11|20|--27--|21| 3|22|--28--|23|
      | 29      |30| 31      |32| 33      |34|
      | HEAT    | 8| 25|--35--|26| 2|27|--36--|28| 6|29|
      | 37      |38| 39      |40| 41      |42|
      | ENERGY| 11|39|--44--|33| 12|34|--45--|35|
      |30|--43--|31| 8|32|--44--|33| 12|34|--45--|35|
      | 46      |47| 48      |49| 50      |51|
      | ENERGY| 5|37|--52--|38| 11|39|--53--|40| 4|41|
      | 54      |55| 56      |57| 58      |59|
      |42|--60--|43| 6|44|--61--|45| 9|46|--62--|47|
      | 63      |64| 65      |66|
      |48|--67--|49| 9|50|--68--|51|
      | 69      |70|
      |52|--71--|53|
Builder Blue where do yo want to build a basement?

```

This command loads the game with the board specified in the file xxx(layout.txt) instead of using random generation. This file will contain the order of the tiles for the game as described in the saving the game section. The number of each tile present does not have to match the random generation. E.g.: it's valid to have a board which is entirely WIFI resources with the value 12.

4) We use ./constructor -random-board:

```

[x6861iu@ubuntu1804-010:~/cs246/1209/projects/constructor/codeForStudents/mycode$ ./constructor -random-board

      | 0|-- 0--| 1|
      | 1      | 2|
      | 0      | 3|
      | 2|-- 3--| 3| 4|-- 4--| 5|
      | 5      | 6| 7      | 8|
      | 1      | 6| 2      | 8|
      | WIFI    | 9| 8|--10--| 9| 10|--11--|11|
      | 12      |13| 14      |15| 16      |17|
      | GLASS   |11|13|--18--|14| 5|15|--19--|16| 12|17|
      | 20      |21| 22      |23| 24      |25|
      |18|--26--|19| 8|20|--27--|21| 9|22|--28--|23|
      | 29      |30| 31      |32| 33      |34|
      | HEAT    | 4|25|--35--|26| 2|27|--36--|28| 6|29|
      | 37      |38| 39      |40| 41      |42|
      | BRICK   |11|39|--44--|33| 3|34|--45--|35|
      |30|--43--|31| 10|32|--44--|33| 3|34|--45--|35|
      | 46      |47| 48      |49| 50      |51|
      | HEAT    | 5|37|--52--|38| 11|39|--53--|40| 3|41|
      | 54      |55| 56      |57| 58      |59|
      |42|--60--|43| 8|44|--61--|45| 6|46|--62--|47|
      | 63      |64| 65      |66|
      |48|--67--|49| 10|50|--68--|51|
      | 69      |70|
      |52|--71--|53|

```

This command starts a game with a randomly generated board. If a game is being loaded, this command is ignored and the layout specified in the saved game is used instead.