

CS246 Final Project: Constructor

Demonstration Plan

Running Environment:

This program is expected to run on Linux. Although we only used std functions and objects in coding and no Linux system call was used, we still cannot guarantee the program's behaviour on other operating systems.

Command-line Arguments:

The program supports four different command-line arguments, three of which specify the way of initializing the game.

1. -load xxx: The game loads from a previously saved file specified by the argument after '-load'.

```
$ ./ctor.exe -load test.txt
Builder Blue's turn.
↳ fair roll
↳ No builders gained resources.
↳ board
          |BB|-- 0--| 1|
          |      |
          |      1   0   2
          |      ENERGY |
          | 2|-- 3--| 3| 11 |RB|-- 4--| 5|
          |      GEESE |
          |      5   1   6   7   2   8
          |      ENERGY |
          | 6|-- 9--| 7| 4 |OB|--10--| 9| 12 |10|--11--|11|
          |      |
          |      12  3   13   14   4   15   16   5   17
          |      ENERGY |   PARK |
          |YB| 8 |13|--18--|14| 15|--19--|YB| 6 |17|
          |      |
          |      20  21   6   22   23   7   24   25
          |      BRICK |   BRICK |
          |18|--26--|19| 10 |OB|--27--|21| 9 |22|--28--|23|
          |      |
          |      29  8   30   31   9   32   33   10  34
          |      ENERGY |   GLASS |   GLASS |
          |RB| 9 |25|--35--|26| 5 |BB|--36--|28| 10 |29|
          |      |
          |      37  38   11  39   40   12  41   42
          |      HEAT |   GLASS |
          |30|--43--|31| 3 |32|--44--|33| 11 |34|--45--|35|
          |      |
          |      46  13  47   48   14  49   50   15  51
          |      HEAT |   GLASS |   BRICK |
          |36| 2 |37|--52--|38| 6 |39|--53--|40| 5 |41|
          |      |
          |      54  55  16  56   57   17  58   59
          |      WIFI |   WIFI |
          |42|--60--|43| 3 |44|--61--|45| 8 |46|--62--|47|
          |      |
          |      63  64  18  65   66
          |      HEAT |
          |48|--67--|49| 4 |50|--68--|51|
          |      |
          |      69  70
          |
          |52|--71--|53|
```

2. -board xxx: The program starts a new game with the board layout specified by the file stated after '-board'.

```
$ cat file.txt
1 11 4 4 1 12 1 8 5 7 0 6 0 10 0 9 1 9 2 5 2 10 3 3 2 11 3 2 2 6 0 5 4 3 4 8 3 4
```

```

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

```

3. -random-board: The program starts a new game with a randomly generated board.

```

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

```

4. -seed: The seed argument specifies the seed of the random generator used by the program. The same seed would generate the same random numbers each time. The board below generated with seed 1 is the same as the random board generated without the seed (1 is the default seed), but different from the board generated with seed 10 above.

```
$ ./ctor.exe -seed 1 -random-board
Builder Blue, where do you want to build a basement?
↳ 0
Builder Red, where do you want to build a basement?
↳ 4
Builder orange, where do you want to build a basement?
↳ 8
Builder yellow, where do you want to build a basement?
↳ 12
Builder yellow, where do you want to build a basement?
↳ 16
Builder orange, where do you want to build a basement?
↳ 20
Builder Red, where do you want to build a basement?
↳ 24
Builder Blue, where do you want to build a basement?
↳ 28
          |BB|-- 0--| 1|
          | 1 0 2 |
          | ENERGY |
          | 2|-- 3--| 3| 11 |RB|-- 4--| 5|
          | 5 1 6    7 2 8 |
          | WIFI   ENERGY |
          | 6|-- 9--| 7| 4 |OB|--10--| 9| 12 |10|--11--|11|
          | 12 3 13   14 4 15   16 5 17 |
          | ENERGY   PARK   BRICK |
          | YB| 8 |13|--18--|14| GEESE |15|--19--|YB| 6 |17|
          | 20 21 6 22   23 7 24   25 |
          | BRICK   GEESE   BRICK |
          |18|--26--|19| 10 |OB|--27--|21| 9 |22|--28--|23|
          | 29 8 30   31 9 32   33 10 34 |
          | ENERGY   GLASS   GLASS |
          | RB| 9 |25|--35--|26| 5 |27|--36--|BB| 10 |29|
          | 37 38 11 39   40 12 41   42 |
          | HEAT   GLASS   GLASS |
          |30|--43--|31| 3 |32|--44--|33| 11 |34|--45--|35|
          | 46 13 47   48 14 49   50 15 51 |
          | HEAT   GLASS   BRICK |
          |36| 2 |37|--52--|38| 6 |39|--53--|40| 5 |41|
          | 54 55 16 56   57 17 58   59 |
          | WIFI   WIFI   WIFI |
          |42|--60--|43| 3 |44|--61--|45| 8 |46|--62--|47|
          | 63 64 18 65   66 67 68 69 |
          | HEAT   HEAT   HEAT |
          |48|--67--|49| 4 |50|--68--|51|
          | 69 70 71 72 73 |
          |52|--71--|53|
```

```
$ ./ctor.exe -random-board
Builder Blue, where do you want to build a basement?
↳ 0
Builder Red, where do you want to build a basement?
↳ 4
Builder orange, where do you want to build a basement?
↳ 8
Builder yellow, where do you want to build a basement?
↳ 12
Builder yellow, where do you want to build a basement?
↳ 16
Builder orange, where do you want to build a basement?
↳ 20
Builder Red, where do you want to build a basement?
↳ 24
Builder Blue, where do you want to build a basement?
↳ 28
          |BB|-- 0--| 1|
          | 1 0 2 |
          | ENERGY |
          | 2|-- 3--| 3| 11 |RB|-- 4--| 5|
          | 5 1 6    7 2 8 |
          | WIFI   ENERGY |
          | 6|-- 9--| 7| 4 |OB|--10--| 9| 12 |10|--11--|11|
          | 12 3 13   14 4 15   16 5 17 |
          | ENERGY   PARK   BRICK |
          | YB| 8 |13|--18--|14| GEESE |15|--19--|YB| 6 |17|
          | 20 21 6 22   23 7 24   25 |
          | BRICK   GEESE   BRICK |
          |18|--26--|19| 10 |OB|--27--|21| 9 |22|--28--|23|
          | 29 8 30   31 9 32   33 10 34 |
          | ENERGY   GLASS   GLASS |
          | RB| 9 |25|--35--|26| 5 |27|--36--|BB| 10 |29|
          | 37 38 11 39   40 12 41   42 |
          | HEAT   GLASS   GLASS |
          |30|--43--|31| 3 |32|--44--|33| 11 |34|--45--|35|
          | 46 13 47   48 14 49   50 15 51 |
          | HEAT   GLASS   BRICK |
          |36| 2 |37|--52--|38| 6 |39|--53--|40| 5 |41|
          | 54 55 16 56   57 17 58   59 |
          | WIFI   WIFI   WIFI |
          |42|--60--|43| 3 |44|--61--|45| 8 |46|--62--|47|
          | 63 64 18 65   66 67 68 69 |
          | HEAT   HEAT   HEAT |
          |48|--67--|49| 4 |50|--68--|51|
          | 69 70 71 72 73 |
          |52|--71--|53|
```

Prompt:

According to 4.5 Handling Input of the project specification, “When waiting for a response from standard input, the program prints ‘↳’.”

4.5 Handling Input

When waiting for a response from standard input, the program prints ‘↳’.

Therefore, our program prints an upside-down question mark and space when waiting for user input.

```
Builder Red, where do you want to build a basement?  
↳
```

Beginning Setting of the Board:

Function: determine the location of the two initial bases for each builder in the order of Blue -> Red -> Orange -> Yellow ->Orange -> Red -> Blue.

The system will ask the builder to input the index of the vertex that the builder would like to choose as the initial base.

```
Builder Red, where do you want to build a basement?
```

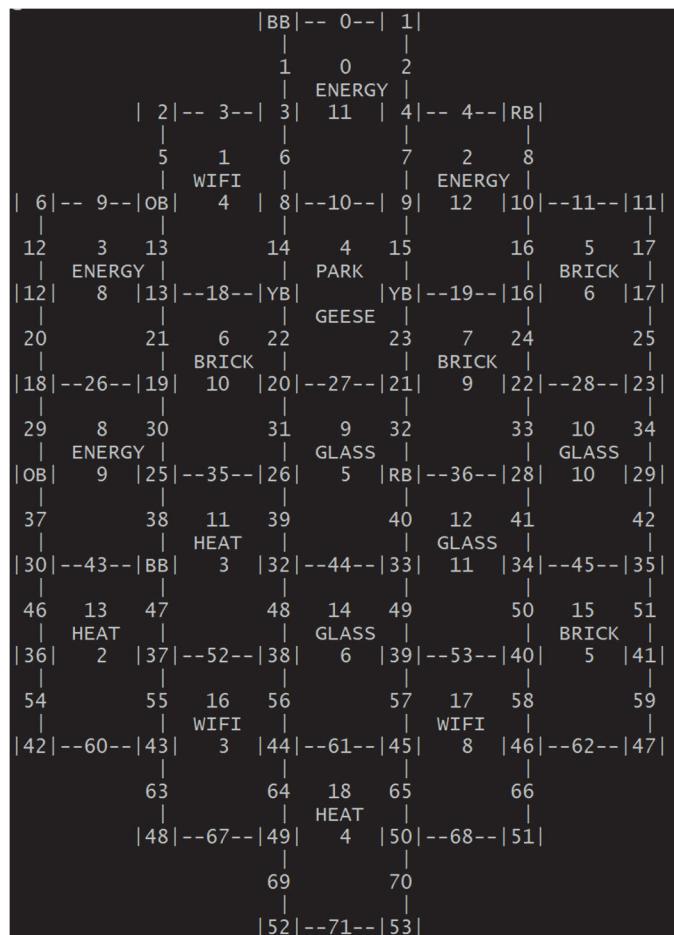
The builder must choose the vertex that can be built. If the vertex already has a basement on it, or there is an adjacent base chose by another builder or the index of the vertex that the user input is larger than the scale, the system will print.

```
linus@linus-virtual-machine:~/ctor-test$ ./ctor -random-board
Builder Blue, where do you want to build a basement?
i 1
Builder Red, where do you want to build a basement?
i 1
You cannot build here.
Builder Red, where do you want to build a basement?
i 100
You cannot build here.
Builder Red, where do you want to build a basement?
i ■
```

If other types of commands are inputted, the system will report invalid.

```
Builder Red, where do you want to build a basement?  
i cs246  
Invalid input.  
Builder Red, where do you want to build a basement?  
i |
```

After all the base locations are chosen correctly, the overall board will be printed out with the number of the vertex being replaced by the first capital character of the builder's colour following "B" that indicates that the type of housing is "Base". Notice that the Geese will be placed in Park at the beginning session of the game and the resources are distributed exactly according to the specification: 3 WIFI, 3 HEAT, 4 BRICK, 4 ENERGY, 4 GLASS, and 1 PARK.



During the Turn:

After the beginning setting of the board. The program will automatically enter the turn of each builder. The order of the builder is:

Blue -> Red -> Orange -> Yellow

After the turn of builder Orange, builder Blue will start the turn again. This cycle would repeat until one builder obtains more than ten building points.

At the beginning of each player's turn, the system will show the name of the builder that owns the turn.

```
Builder Blue's turn.  
Builder Red's turn.  
Builder Orange's turn.  
Builder Yellow's turn.
```

During the turn, the builders will build roads and buildings in order to obtain the building points. Once one builder obtains ten building points, that builder will win the game, the system will stop and ask if another game is required.

Under each turn, there are two stages: Dice Roll, Command. After the Dice Roll, the program will automatically enter to stage Command. The order is Dice Roll -> Command.

Stage 1: Dice Roll

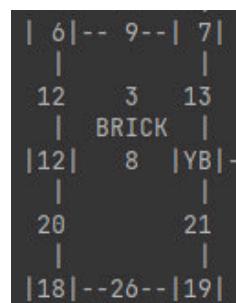
The purpose of this stage is to allocate resources to four builders. This function is achieved by rolling dice. There are two different types of dice: **loaded dice** and **fair dice**. Choosing loaded dices means that the builder can determine the number generated by the dice, whereas the fair dices will generate the number randomly from 1 to 6. In this game, two types of dice are available. The builder needs to choose between loaded dice and fair dice by typing "load" or "fair" and type "roll" to generate the result.

If fair dices are used, the builder should type "fair" and "roll" to generate result:

```
Builder Blue's turn.  
↳ fair  
↳ roll  
Builder Blue gained:  
1 GLASS  
↳ █
```

Two dices are being rolled, which means the number will be generated from zero to twelve. The result of rolling will determine the builders that receive corresponding resources.

To determine which builder receives resources, the following one tile would be a great example.



If the results of rolling dices are 3 and 5 ($3 + 5 = 8$) which is equal to the value of the tile (the number under

BRICK), the result indicates the tile selected and builders who own surrounding buildings on vertices will receive corresponding resources. The resource type and level of buildings will determine which kinds of resources are allocated on what amount. In this example, the builder yellow will receive 1 Brick.

If the tile rolled has no residence, a “No builders gained resources” message will be given.

```
No builders gained resources.  
i |
```

If loaded dices are used, the builder will have to input the number from two to twelve to receive the corresponding result.

```
Builder Blue's turn.  
i load  
i roll  
Input a roll between 2 and 12:  
i 6  
Builder Yellow gained:  
1 BRICK  
i |
```

On the board, if more than one tiles own the same value, when the result of the rolling is generated to be this value, all the tiles with the same value should be considered to allocate resources to the corresponding builders.

If the tile rolled has no residence, a “No builders gained resources” message will be given.

```
Builder Orange's turn.  
i load  
i roll  
Input a roll between 2 and 12:  
i 3  
No builders gained resources.  
i |
```

If other commands are typed instead of “fair”, “load” or “roll”, the program will report “invalid command”.

```
Builder Blue's turn.  
i run  
Invalid command.  
i |
```

If commands “roll” is typed before selecting dice, the program will report “invalid roll”.

```
Builder Red's turn.  
i roll  
Invalid roll.  
i |
```

When the loaded dice is used, but the input of the number is not in the range or is not a number, the program will report “Invalid roll”

```
Builder Orange's turn.  
i load  
i roll  
Input a roll between 2 and 12:  
i 13  
Invalid roll.  
Input a roll between 2 and 12:  
i cs246  
Invalid roll.  
Input a roll between 2 and 12:  
i █
```

If 7 is rolled

The program prints the line “Choose where to place the GEESE”. It then accepts the current player to input any tile index that is not its current index. If the index is valid, it would print the builders with buildings on that tile, whom the current player can steal from. The current player can then input the name of one of the builders and can steal a random resource from the builder.

```
i Choose where to place the GEESE.  
i 2  
Builder orange can choose to steal from Red, Yellow.  
choose a builder to steal from.  
i Red  
Builder orange steals ENERGY from builder Red .
```

If the index the player inputs is not valid, the program would print “invalid input”.

```
i Choose where to place the GEESE.  
i 19  
Invalid input.  
Choose where to place the GEESE.  
i -1  
Invalid input.  
Choose where to place the GEESE.  
i 9
```

If there are no builders with buildings on the tile the player inputs other than the player itself, the program would tell the player there is no builder to steal from after moving the geese to that position.

```
i Choose where to place the GEESE.  
i 15  
Builder orange has no builders to steal from.
```

If the builders with buildings on the tile do not have any resources to steal from, the program would produce the same output that there are no builders to steal from.

```
i status  
Blue has 2 building points, 0 brick, 2 energy, 1 glass, 0 heat, and 0 WiFi.  
Red has 2 building points, 0 brick, 0 energy, 0 glass, 0 heat, and 0 WiFi.  
Orange has 2 building points, 1 brick, 2 energy, 0 glass, 0 heat, and 1 WiFi.  
Yellow has 2 building points, 1 brick, 1 energy, 0 glass, 0 heat, and 0 WiFi.  
i next  
Builder Yellow's turn.  
i load roll  
i Input a roll between 2 and 12:  
i 7  
Choose where to place the GEESE.  
i 0  
Invalid input.  
Choose where to place the GEESE.  
i 2  
Builder yellow has no builders to steal from.
```

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

In the above pictures, the builder Red had no resources. When the builder Yellow moved the geese to tile 2, the only two builders with residences on that tile were Red and Yellow. Hence Yellow had no builders to steal from. After moving the geese, its new position would appear on the board layout as shown above under the tile value.

Stage 2: Command

The purpose of this stage is for a builder to obtain building points by giving different commands to exchange resources, build roads and bases, improve residences, and view the current situation on board. After stage 1: Dice-Roll, the program will automatically shift to the command stage. All the commands are input by directly typing to the command board.

Command 1# help:

Function: print a list of valid commands that can be used by the builder during the command stage.

```
i help
Valid commands:
board
status
residences
build-road <edge#>
build-res <housing#>
improve <housing#>
trade <colour> <give> <take>
next
save <file>
help
i
```

Command 2# board:

Function: prints the current board and presents the latest change. The builder can receive information

from the board, including roads and residences.

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

Command 3# status:

Function: prints the current information of all builders including building points, resource types and amount. The modification will be updated after each change to resources.

```
i status
Blue has 2 building points, 0 brick, 0 energy, 0 glass, 0 heat, and 0 WiFi.
Red has 2 building points, 0 brick, 0 energy, 0 glass, 1 heat, and 0 WiFi.
Orange has 2 building points, 0 brick, 0 energy, 0 glass, 0 heat, and 0 WiFi.
Yellow has 2 building points, 0 brick, 0 energy, 0 glass, 0 heat, and 0 WiFi.
```

Command 4# residences:

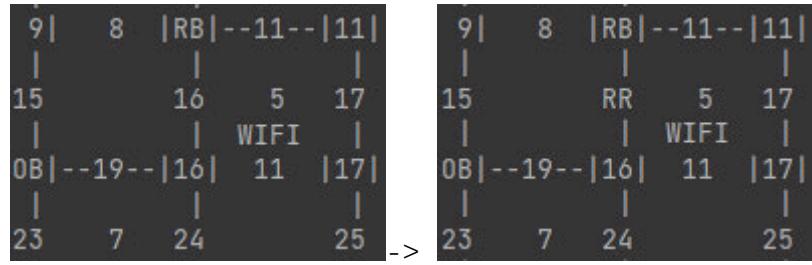
Function: prints the residences of the current turn's builder including information about the current location and building types. "B" stands for a basement, "H" stands for a house, and "T" stands for a tower.

```
i residences
Blue has built:
0 B
52 B
```

Command 5# build-road + (edge_num):

Function: build the road to the location with the corresponding index. The road must be built with adjacent roads or residences. Each build of the road will cost one Heat and WIFI. After building the road, the number of the corresponding edge will change to the first character followed by "R" indicating that is a road.

Using "build-road 16" in Red's turn:



The resources will be deducted from the builder as well.

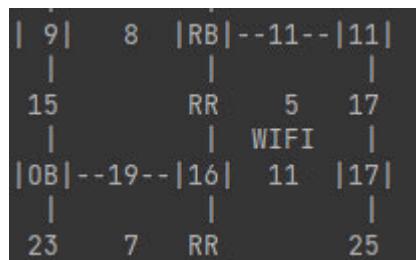
(before)

```
i status
Blue has 2 building points, 0 brick, 0 energy, 0 glass, 0 heat, and 0 WiFi.
Red has 2 building points, 0 brick, 0 energy, 0 glass, 2 heat, and 1 WiFi.
Orange has 2 building points, 0 brick, 0 energy, 0 glass, 0 heat, and 1 WiFi.
Yellow has 2 building points, 0 brick, 0 energy, 0 glass, 0 heat, and 0 WiFi.
```

(after)

```
i status
Blue has 2 building points, 0 brick, 0 energy, 0 glass, 0 heat, and 0 WiFi.
Red has 2 building points, 0 brick, 0 energy, 0 glass, 1 heat, and 0 WiFi.
Orange has 2 building points, 0 brick, 1 energy, 0 glass, 0 heat, and 1 WiFi.
Yellow has 2 building points, 0 brick, 0 energy, 0 glass, 1 heat, and 0 WiFi.
```

Also, a road can be built adjacent to another road without any building next to it.



If the edge number is outside the scale or invalid (does not connect to a residence owned by this builder), the system will report you cannot build there.

```
i build-road 100
You cannot build here.
i build-road 1
You cannot build here.
i ■
```

Also, the builder will report "You do not have enough resources" when the resource is not enough to build a road.

```
i build-road 17
You do not have enough resources.
i ■
```

If invalid input is detected, the program will report it as well.

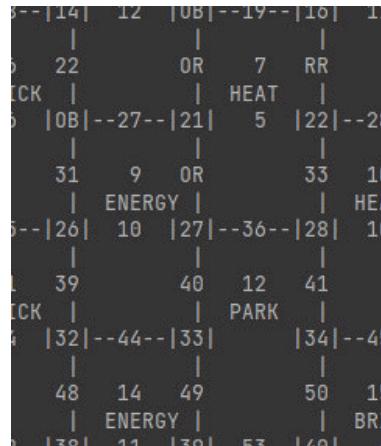
```
i build-road uw
Invalid input.
i ■
```

Command 6# build-res + (vertex_num):

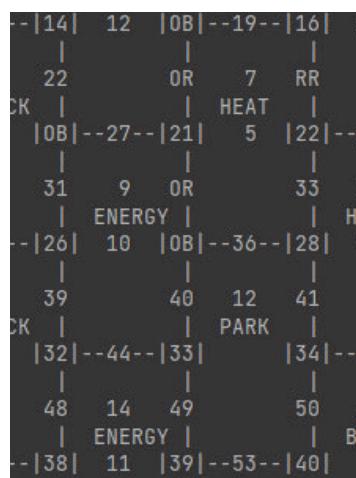
Function: This command will help the builder to build a basement. The base cannot be built with an adjacent residence, and there has to be at least one road connect to it. The cost of a basement is one Heat, one Brick, one Glass, one Wifi. The vertex will be replaced by the first character of the builder and followed by "B", indicating it is a base.

Using "build-res 27" in Orange's turn:

(Before)



(After)



Also, the corresponding resource will be deducted from the builder, and the building point will increase one since one more basement is created:

```

i status
Blue has 2 building points, 5 brick, 0 energy, 2 glass, 0 heat, and 0 WiFi.
Red has 2 building points, 1 brick, 0 energy, 0 glass, 3 heat, and 3 WiFi.
Orange has 2 building points, 3 brick, 1 energy, 2 glass, 1 heat, and 2 WiFi.
Yellow has 2 building points, 0 brick, 0 energy, 4 glass, 2 heat, and 0 WiFi.
  
```

(after)

```

i status
Blue has 2 building points, 5 brick, 0 energy, 2 glass, 0 heat, and 0 WiFi.
Red has 2 building points, 1 brick, 0 energy, 0 glass, 3 heat, and 3 WiFi.
Orange has 3 building points, 2 brick, 0 energy, 1 glass, 1 heat, and 1 WiFi.
Yellow has 2 building points, 0 brick, 0 energy, 4 glass, 2 heat, and 0 WiFi.
  
```

Also, the program will print "you cannot build there.", if the number of vertex input is larger than the scale, and "invalid input.", if undefined input is detected.

```
i build-res 100
You cannot build here.
i build-res uw
Invalid input.
i
```

If the resource of the builder is not enough, the system will report “You do not have enough resources”, when the resources are not enough.

```
i build-res 22
You do not have enough resources.
i
```

Command 7# improve + (vertex_num):

Function: This command will allow the builder to upgrade the residence. There are 3 types of residences: Base, House, Tower. The build has to be upgraded in the order Base->House->Tower. From a basement to a house, the builder will spend **two** Glass, and **three** Heat. From a house to a tower, the builder will spend **three** Brick, **two** Energy, **two** Glass, **one** Wifi and **two** Heat.

When improving a basement to a house by using “improve 29” in Yellow’s turn:

(before)

33	10	34
	HEAT	
- 28	10	YB
41		42
34	--45--	35

(after)

33	10	34
	HEAT	
-- 28	10	YH
41		42
K		
34	--45--	35

Also, the corresponding resources will be deducted as well, and the type of the building will be changed in residences as well:

(before)

```
i status
Blue has 2 building points, 4 brick, 0 energy, 4 glass, 0 heat, and 0 WiFi.
Red has 2 building points, 2 brick, 0 energy, 1 glass, 3 heat, and 5 WiFi.
Orange has 3 building points, 4 brick, 14 energy, 1 glass, 4 heat, and 1 WiFi.
Yellow has 2 building points, 0 brick, 0 energy, 7 glass, 9 heat, and 0 WiFi.
i residences
Yellow has built:
29 B
30 B
i improve 29
```

(after)

```

i status
Blue has 2 building points, 4 brick, 0 energy, 4 glass, 0 heat, and 0 WiFi.
Red has 2 building points, 2 brick, 0 energy, 1 glass, 3 heat, and 5 WiFi.
Orange has 3 building points, 4 brick, 14 energy, 1 glass, 4 heat, and 1 WiFi.
Yellow has 3 building points, 0 brick, 0 energy, 5 glass, 6 heat, and 0 WiFi.
i residences
Yellow has built:
29 H
30 B

```

When improving a house to a tower (done in Orange's turn):

(before)

21	6	22		OR	
	BRICK				
19	6	OH	--27--	21	
30		31	9	OR	
			ENERGY		

(after)

```
i improve 20
```

The display of the house will be changed to a tower

10	10	10	10	10	10
21	6	22		OR	
	BRICK				
19	6	OT	--27--	21	
30		31	9	OR	
			ENERGY		

The building point will increase one, the corresponding resources will be deducted, and the building type to change from house to tower.

(before)

```

i status
Blue has 2 building points, 4 brick, 0 energy, 4 glass, 0 heat, and 0 WiFi.
Red has 2 building points, 0 brick, 8 energy, 0 glass, 3 heat, and 0 WiFi.
Orange has 4 building points, 6 brick, 6 energy, 3 glass, 4 heat, and 6 WiFi.
Yellow has 3 building points, 0 brick, 6 energy, 2 glass, 9 heat, and 0 WiFi.
i residences
Orange has built:
15 B
20 H
27 B

```

(after)

```

i status
Blue has 2 building points, 4 brick, 0 energy, 4 glass, 0 heat, and 0 WiFi.
Red has 2 building points, 0 brick, 8 energy, 0 glass, 3 heat, and 0 WiFi.
Orange has 5 building points, 3 brick, 4 energy, 1 glass, 2 heat, and 5 WiFi.
Yellow has 3 building points, 0 brick, 6 energy, 2 glass, 9 heat, and 0 WiFi.
i residences
Orange has built:
15 B
20 T
27 B
.

```

If resources are not enough, the program will report just as what it does in build-road and build-res. If a residence that does not belong to the current builder or an invalid vertex is improved, the program will report "you cannot build there". The system will also report "invalid input" if the input is not valid.

```

i improve 11
You do not have enough resources.
i improve 16
You cannot build here.
i improve 999
You cannot build here.
i improve uw
Invalid input.
i ■

```

Command 8# trade + (builder's name) + (provide resource_type) + (receive resource_type):

Function: This command allows the builder to exchange resources one at a time, and another builder has the right to choose whether or not to accept the offer given by the current builder by typing "yes/no". The resource type has to be typed in capital form. The first resource type represents what kind of resources the builder will offer in return for the second resource type. If type "yes", the builder will be able to exchange resources.

Builder's name: Blue, Red, Orange or Yellow. The spelling and capitalization should be exact.

Resource Type: BRICK, ENERGY, GLASS, HEAT or WIFI. The spelling and capitalization should be exact.

(The following example is done in builder Orange's turn)

```

i status
Blue has 2 building points, 0 brick, 0 energy, 2 glass, 0 heat, and 0 WiFi.
Red has 2 building points, 1 brick, 0 energy, 1 glass, 0 heat, and 0 WiFi.
Orange has 2 building points, 1 brick, 1 energy, 0 glass, 2 heat, and 1 WiFi.
Yellow has 2 building points, 0 brick, 0 energy, 3 glass, 1 heat, and 0 WiFi.
i trade Yellow HEAT GLASS
Orange offers Yellow for one HEAT for one GLASS.
Does Yellow accept this offer?
i yes
i status
Blue has 2 building points, 0 brick, 0 energy, 2 glass, 0 heat, and 0 WiFi.
Red has 2 building points, 1 brick, 0 energy, 1 glass, 0 heat, and 0 WiFi.
Orange has 2 building points, 1 brick, 1 energy, 1 glass, 1 heat, and 1 WiFi.
Yellow has 2 building points, 0 brick, 0 energy, 2 glass, 2 heat, and 0 WiFi.

```

If type "no", the exchange will not be implemented.

```

i status
Blue has 2 building points, 4 brick, 0 energy, 4 glass, 0 heat, and 0 WiFi.
Red has 2 building points, 0 brick, 8 energy, 0 glass, 3 heat, and 0 WiFi.
Orange has 5 building points, 3 brick, 4 energy, 1 glass, 2 heat, and 5 WiFi.
Yellow has 3 building points, 0 brick, 6 energy, 2 glass, 9 heat, and 0 WiFi.
i trade Yellow WIFI GLASS
Orange offers Yellow for one WIFI for one GLASS.
Does Yellow accept this offer?
i no
i status
Blue has 2 building points, 4 brick, 0 energy, 4 glass, 0 heat, and 0 WiFi.
Red has 2 building points, 0 brick, 8 energy, 0 glass, 3 heat, and 0 WiFi.
Orange has 5 building points, 3 brick, 4 energy, 1 glass, 2 heat, and 5 WiFi.
Yellow has 3 building points, 0 brick, 6 energy, 2 glass, 9 heat, and 0 WiFi.
i |

```

If another builder does not have enough resources, the program will print "The other builder does not have enough resources"

```

i status
Blue has 2 building points, 4 brick, 0 energy, 4 glass, 0 heat, and 0 WiFi.
Red has 2 building points, 0 brick, 8 energy, 0 glass, 3 heat, and 0 WiFi.
Orange has 2 building points, 3 brick, 4 energy, 1 glass, 2 heat, and 5 WiFi.
Yellow has 2 building points, 0 brick, 6 energy, 2 glass, 9 heat, and 0 WiFi.
i trade Red ENERGY BRICK
The other builder does not have enough resources.
i |

```

(The following example is done in builder Yellow's turn)

If the current builder does not have enough resources to act exchange, the program will print "You do not have enough resources."

```

Builder Yellow's turn.
i load
i roll
Input a roll between 2 and 12:
i 3
No builders gained resources.
i status
Blue has 2 building points, 4 brick, 0 energy, 4 glass, 0 heat, and 0 WiFi.
Red has 2 building points, 0 brick, 8 energy, 0 glass, 3 heat, and 0 WiFi.
Orange has 2 building points, 3 brick, 4 energy, 1 glass, 2 heat, and 5 WiFi.
Yellow has 2 building points, 0 brick, 6 energy, 2 glass, 9 heat, and 0 WiFi.
i trade Red BRICK ENERGY
You do not have enough resources.
i |

```

The program will print "invalid input" for all other types of input:

```

i trade Red BBQ MONEY
Invalid input.
i |

```

Command 9# next:

Function: This command allows the current builder to stop their turn and shift to the next builder.

```

Builder Blue's turn.
i fair roll
i Builder Orange gained:
2 HEAT
i next
Builder Red's turn.
i |

```

Command 10# save + (File_name):

Function: This command allows the player to save the current game and board layout into specific documents.

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

The document is saved into a file called test.sv, and the test.sv has the correct form of layout that recording the location of the buildings, roads and resources.

```
$ cat test.sv
0
0 3 1 0 0 r h 0 B 27 B
0 0 0 0 0 r h 4 B 24 B
1 1 0 0 1 r h 8 B 20 B
1 1 0 0 0 r h 12 B 16 B
1 11 4 4 1 12 1 8 5 7 0 6 0 10 0 9 1 9 2 5 2 10 3 3 2 11 3 2 2 6 0 5 4 3 4 8 3 4
0
```

It is clear to see that the document has been saved in the directory. Also, the program allows users to load the game with a specific document that records the data.

```

$ ./ctor.exe -load test.sv
Builder Blue's turn.
↳ fair roll
↳ No builders gained resources.
↳ board
          |BB|-- 0--| 1|
          |      |
          |      0   2|
          |      ENERGY |
          | 2|-- 3--| 3| 11 |RB|-- 4--| 5|
          |      GEESE |
          |      5   1   6   7   2   8|
          |      WIFI    ENERGY |
          | 6|-- 9--| 7| 4 |OB|--10--| 9| 12 | 10|--11--|11|
          |      ENERGY   PARK   15   16   5   17|
          |YB| 8 |13|--18--|14| 14| 4 | 15| 16| YB| 6 |17|
          |      ENERGY   PARK   15| 15| 19| 16| YB| 6 |17|
          |      BRICK   22| 23| 7 | 24| 25|
          | 20| 21 | 6 | 22| 23| 7 | 24| 25|
          |      BRICK   22| 23| 7 | 24| 25|
          |18|--26--|19| 10 |OB|--27--|21| 9 | 22|--28--|23|
          |      ENERGY   GLASS   31| 32| 33| 34|
          |RB| 9 |25|--35--|26| 5 |BB|--36--|28| 10 |29|
          |      ENERGY   GLASS   31| 32| 33| 34|
          |      HEAT   39| 40| 12| 41| 42|
          | 37| 38 | 11| 39| 40| 12| 41| 42|
          |      HEAT   39| 40| 12| 41| 42|
          |30|--43--|31| 3 |32|--44--|33| 11 |34|--45--|35|
          |      HEAT   37| 38| 48| 49| 50| 15| 51|
          |46| 13 | 47| 48| 14| 49| 50| 15| 51|
          |      HEAT   37| 38| 48| 49| 50| 15| 51|
          |36| 2 |37|--52--|38| 6 |39|--53--|40| 5 |41|
          |      WiFi   16| 56| 57| 17| 58| 59|
          |54| 55 | 16| 56| 57| 17| 58| 59|
          |      WiFi   16| 56| 57| 17| 58| 59|
          |42|--60--|43| 3 |44|--61--|45| 8 |46|--62--|47|
          |      WiFi   18| 65| 66|
          |63| 64 | 18| 65| 66|
          |      HEAT   49| 50| 51|
          |48|--67--|49| 4 |50|--68--|51|
          |      HEAT   49| 50| 51|
          |69| 70 |
          |52|--71--|53|

```

EOF:

The program only expects an EOF at the point of waiting for the above ten commands. We cannot guarantee the behaviour of the program if an EOF is given at any other position.

An EOF entered while the program is waiting for the above ten commands will generate a document called backup.sv, which is used to document the data during the turn in order to prevent unexpected exit. The current game data will be recorded in the backup.sv, which can be used to reload the game.

```

$ cat backup.sv
0
0 3 1 0 0 r h 0 B 27 B
0 0 0 0 r h 4 B 24 B
1 1 0 0 1 r h 8 B 20 B
1 1 0 0 0 r h 12 B 16 B
1 11 4 4 1 12 1 8 5 7 0 6 0 10 0 9 1 9 2 5 2 10 3 3 2 11 3 2 2 6 0 5 4 3 4 8 3 4
0

```

Ending Session of the Game:

When any builder has achieved ten building points during the turn, the program will enter the ending session of the game. The program will ask whether the user wants to start another game.

```

status
Blue has 2 building points, 4 brick, 4 energy, 0 glass, 0 heat, and 0 WiFi.
Red has 2 building points, 2 brick, 8 energy, 0 glass, 3 heat, and 1 WiFi.
Orange has 9 building points, 1 brick, 1 energy, 3 glass, 12 heat, and 0 WiFi.
Yellow has 3 building points, 2 brick, 11 energy, 0 glass, 8 heat, and 1 WiFi.
↳ residences
Orange has built:
4 B
8 B
15 H
20 T
27 H
↳ improve 4
Would you like to play again?

```

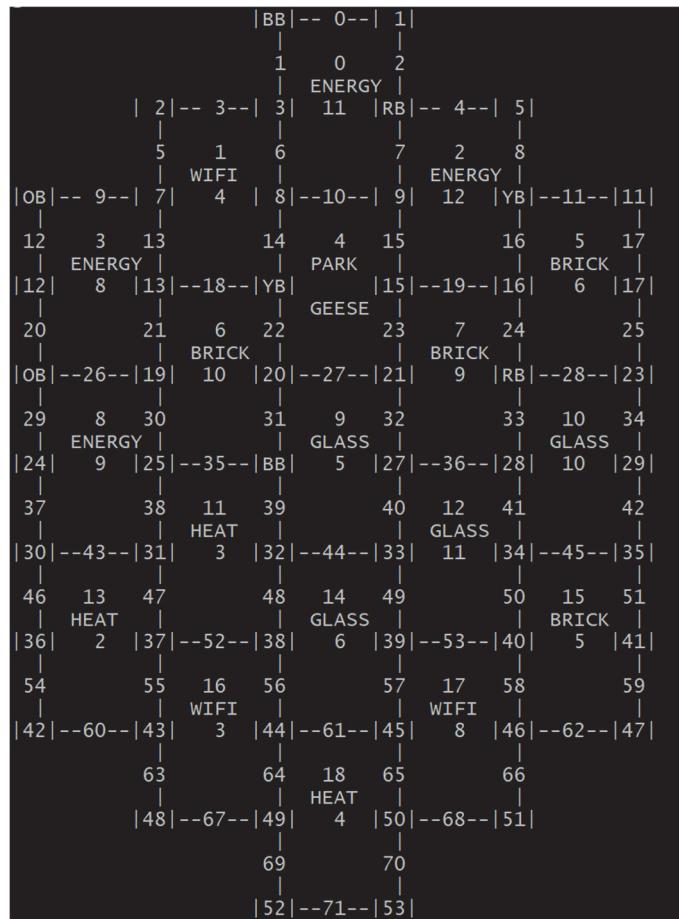
If type "yes", the board will be cleaned, and start a new game.

```
i yes  
Builder Blue, where do you want to build a basement?  
i |
```

The program will ask the build to re-write the initial basements.

```
Builder Blue, where do you want to build a basement?  
i 0  
Builder Red, where do you want to build a basement?  
i 4  
Builder Orange, where do you want to build a basement?  
i 6  
Builder Yellow, where do you want to build a basement?  
i 10  
Builder Yellow, where do you want to build a basement?  
i 14  
Builder orange, where do you want to build a basement?  
i 18  
Builder Red, where do you want to build a basement?  
i 22  
Builder Blue, where do you want to build a basement?  
i 26
```

If the program is executed with "-random-board" option, the game will be generated again randomly. It is clear to see that the board is completely new.



If the program is executed with other options (reading board layout or the entire game from file), the layout from the file will be used with all residences and roads removed and geese will be returned to park.

And all the player's data are reset.

```
Builder Blue's turn.  
i fair roll  
i Builder Blue gained:  
1 GLASS  
Builder Red gained:  
2 GLASS  
Builder Orange gained:  
1 GLASS  
i status  
Blue has 2 building points, 0 brick, 0 energy, 1 glass, 0 heat, and 0 WiFi.  
Red has 2 building points, 0 brick, 0 energy, 2 glass, 0 heat, and 0 WiFi.  
Orange has 2 building points, 0 brick, 0 energy, 1 glass, 0 heat, and 0 WiFi.  
Yellow has 2 building points, 0 brick, 0 energy, 0 glass, 0 heat, and 0 WiFi.  
i |
```

But the program will not reset the random number generator in order to introduce some variant to the new game (The game could be too predictable as the board layout and dice roll results will be certain if we reset the random number generator).

If type “no”, the program will exit.