CS246 Final Project Watopoly - Demo

By Martin Kong, Zhaocheng Che, Xiaoyu Zhu

1 Command line arguments:

- -load <filename> : loads a saved game from the document <filename>.
- -testing : enables the players to decide on the result of their each movement roll, and the player's input can be any non-negative integer.
- -vText and -vGraph: the -vText option turns on the text display of the game board, and the -vGraph option turns on the graphic display. The default display is set to the text display, and if both arguments are included, both the text display and the graphic display will be drawn.
- -rule1 -rule2 -rule3 -rule4: enables some custom house rules.

Note 1: When starting a new game, we need to make sure that the file "initial_information.txt" is in the same directory as the executable "watopoly", otherwise the initial grid will not be set up properly.
Note 2: In testing mode, the users should still input only "roll" when they want to roll. Then, the program will ask you about the dice results if the program needs them. Specifically, the command "roll num1 num2" in one line will still work if the player is not in DC Tims Line, but if player is in DC Tims Line, then the user should input "roll" without numbers, then choose their getting-out option, which might ask about dice results.

2 Demo examples:

For the following examples, please run by
"./watopoly -load demo/example.txt -testing -vText -vGraph" or
"./watopoly -load demo/example.txt -testing < demo/example.in"

2.1 Ordinary game process under testing mode

For this section, we have placed all the necessary input to the watopoly game in the file "ordinary.in". You can use input redirection or copy/paste to input the commands specified in "ordinary.in". Or alternatively, if you prefer an interactive demonstration, you can look at "ordinary.in" and type each command accordingly.
For this first demo, we will start with a new game, so run "./watopoly -testing -vText -vGraph" for an interactive demo, or run "./watopoly -testing < demo/ordinary.in" if you just want to see the outputs.
We start the game by trying to have 1 player for this game, which is rejected by the program, then we tried to have 9 players for this game, which is again rejected, next we tried to have 8 players for this game and succeeded. Then we input the symbol and name of each player, which are:
Player 1: symbol - 1, name - p1
Player 1: symbol - 2, name - p2
Player 1: symbol - 3, name - p3
Player 1: symbol - 4, name - p4
Player 1: symbol - 5, name - p5
Player 1: symbol - 6, name - p6
Player 1: symbol - 7, name - p7
Player 1: symbol - 8, name - p8

Next, following the commands in "ordinary.in", the game process is:

p1 roll (1,1) go to SLC, give up cup, go back 1 step, buy AL,
roll (1,1) proceed 2 step, buy ML, improve ML to level 5, check assets,
roll (0,1) go to Tuition, check cannot see assets, choose option 2, lose $150, next

p2 roll (1,1) go to SLC, give up cup, go back 1 step to AL, pay double amount of standard tuition,

roll (1,1) proceed 2 step to ML, pay level 5 tuition, check assets,
roll (4,0) go to NH, give up cup, gain $200, next

p3 roll (1,1) go to SLC, get a cup,
roll (5,5) go to PAC, buy PAC,
roll (16,0) go to CIF, buy CIF, next

p4 roll (1,1) go to SLC, give up cup, go back 3, collect OSAP on way, to DC, auction, p8 get it with $3,
roll (13,0) go to PAC, collect OSAP on way, pay 10 times of newly rolled dice, next,

p5 roll (1,1) go to SLC, give up cup, choose to go OSAP, collect OSAP, "all", "trade p3 100 CIF - yes",
roll (13,0) go to DWE, auction building, no one offer, next

p6 roll (6,6) go to PAC, pay 4 times of newly rolled dice,
roll (13,13) go to Coop Fee, pay $150,
roll (2,0) land on OSAP, collect $200, next

p7 roll (20,0) go to Goose Nesting

Now we will type "save ordinaryDemo.txt" to save the current state of the game into "ordinaryDemo.txt".
Now we will make a copy of it. Call the new copy "cheatDemo.txt". And then we can cheat by assigning MKV, UWP and V1 to player p7. That is, we can find the line corresponding to MKV, UWP, V1 in "cheatDemo.txt", then change the second entry from "BANK" to "p7".
Now we will load the game from "cheatDemo.txt". Just like before, you can either run "./watopoly -load demo/cheatDemo.txt -testing -vText -vGraph" or "./watopoly -load demo/cheatDemo.txt -testing < demo/ordinary2.in" (Note that this time, we have a new input file called ordinary2.in)

Following the commands in "ordinary2.in", the game process is:
p7 roll (0,5) go to V1, his own building does not charge rent, next
p8 roll (2,3) go to MKV, pay $100 to p7

(The relevant files: ordinary.in, ordinary2.in, ordinaryDemo.txt, cheatDemo.txt can be found in demo.zip.)

After running these tests, readers can try any creative commands to continue the game. The following sections are intended to clarify our decisions on some unspecified program behaviours, upon which more tests on more complicated program's behaviours are based.

2.2 Trade, mortgage and improvement limitations

In this section, we will demonstrate the limitation on the players when they try to trade/mortgage/improve a building. In detail: a player cannot trade a mortgaged building; a player cannot mortgage an improved building; a player can improve an academic building only if the player controls the monopoly block of that building; moreover, when a player own a monopoly block, if any building of the monopoly has improvement, then the player cannot trade or mortgage any of the the monopoly's building, on the other hand, if any building of the monopoly is mortgaged, then the player cannot improve any of the monopoly's building.

For this example, run "./watopoly -load demo/limitation.txt -testing -vText -vGraph" and input commands listed in "demo/limitation.in".
You can also type "./watopoly -load demo/limitation.txt -testing < demo/limitation.in" to see the results directly.

At the start of this demo, p1 owns the academic buildings AL, ML, MC, and all are unmortgaged and unimproved. p2 owns the monopoly blocks Arts2, Eng, Gym, and a single academic building DC, in which the Arts2 block has two buildings with improvements, the Eng block's buildings are neither mortgaged nor improved, the Gym block as a mortgaged building, and DC is not mortgaged.
Following the commands given in limitation.in, p1 in turn tries:

1. Mortgage and unmortgage MC, which are both successful; buy and sell improvements of MC, which both fail since p1 does not own the entire monopoly block of MC.
2. Improve AL, which is successful since p1 owns the monopoly block of AL - Art1; mortgage AL and ML, both attempts fail since AL now have improvements and ML is in the same block with AL. Then p1 sells AL's improvement, and successfully mortgages AL. Now p1 tries to improve AL and ML, and both attempts fail.
3. Next p1 tries to trade $1600/MKV with p2 for DC, and both cases fail because p1 does not own that much money/p1 does not own MKV. p1 then trades $100 with p2 for DC and succeeds. p1 again tries to trade $100 with p2 for PAC/ECH, and both cases fail since PAC is mortgaged, and the block of ECH has improved buildings. At the end, p1 tries to trade $100/MC with p2 for RCH/CIF, and the message is successfully sent since the monopoly block of RCH owned by p2 does not have any mortgaged or improved building, and CIF itself is not mortgaged, then p2 rejects the first trade and accepts the send trade.

2.3 Bankrupt, Auction, and Game ending

The indicator of whether a player can bankrupt is if he is in debt. A player becomes in debt when he is obligated to pay fee more than his cash. After one is in debt after some move, one will automatically pay off the debt once one raised enough money. But before one pays off the debt, one cannot next, roll, or do anything that does not raise money. After some player declares bankruptcy, if there is only one player left, then the game should end.

In this section, we only deal with debt caused by landing on a square. Debt caused by paying prison fine and receiving mortgaged property are explained in the last two sections. Normal debt is caused either landing on a non-property square, or paying rent to the building owner. We are going to test both scenarios here.

If a player bankrupted owing money to the bank, then its properties with improvements and cash is auctioned. Its cups are destroyed. If there is a max offer in this auction, then all properties with improvements and cash are transferred to the offeror, with all mortgaged properties unmortgaged automatically, and the original debt to the bank is waived. If no one makes an offer, then cash disappears and all properties are unmortgaged and become unimproved as a fresh game starts.

If a player bankrupted owing money to another player, then its properties with improvements, cash, and cups are transferred to the owner. The owner must pay a transfer fee according to specification. (Note that the receiver might own money to the bank after this process, more at next Double Kill section.)

Auction Logic: (this actually applies to both common building auction and bankrupted player auction)
Ask potential bidders in order, remind them of the max offer and max offerer each time. If they input a number smaller or equal to the current offer, it means they decide to withdraw and will not be asked on next turn (if applicable). If they input a number larger than the current offer, the input is only allowed if the bidder can afford the price with cash, otherwise, the input is rejected. The auction will end if it is the turn for the last max bidder to bid.

The following demo examples correspond to load and input files.
For example, run "./watopoly -load demo/bankrupt0.txt -testing -vText -vGraph"
You can also type "./watopoly -load demo/bankrupt0.txt -testing < demo/bankrupt0.in"

"bankrupt0": p1 is not in debt, try bankrupt, program does not allow

"bankrupt1": p1 becomes in debt after rolling double and landing on tuition paying 300, try various invalid commands (next, roll, etc.) to be rejected; pay off debt by trading DC with p2, selling improvement on MC and mortgaging MKV, then roll to continue.

"bankrupt2": p1 becomes in debt after landing on tuition paying 300 > current money, bankrupt, auction the player, received by p4 who has enough money to cover the transaction fee, so the next player in control should be p2, check p4's money (1500 - 550 + 100), cup (0), and properties

"bankrupt3": p1 becomes in debt after landing on p4' HH, in debt after paying rent, bankrupt, transfer property to p2, check p4's money (1500 - 200 * 0.1 - 350 * 0.6 + 5), cup (1), and properties

"bankrupt4": p1 becomes in debt after landing on tuition paying 300 > current money, bankrupt, auction the player, no one makes an offer

"bankrupt5": only two players, p1 bankrupt for tuition, p2 wins
"bankrupt6": only two players, p1 bankrupt for rent to p2, p2 wins

2.4 Double kill bankrupt

This is a bizarre case: A player can go bankrupt because he/she receives mortgaged property from a bankrupted player. That is, suppose we have three players P1, P2, P3. Also suppose P1 controls a mortgaged property. Now if P1 steps on P3's property and goes bankrupt, P3 will receive P1's mortgaged property. But if P3 does not have enough money to pay for the 10% mortgage transfer fee, P3 will be in debt. If P3 willingly chooses to go bankrupt without trying to sell off its properties, then P2 will win the game.
In detail, after a player receives mortgaged property and pays for the 10% transaction fee, if he is not in debt, then the control is transferred to the next player in order. If he is in debt for paying the mortgage transfer fee, then the control is transferred directly to him, skipping all the players in between. If he cannot pay off the debt, then he goes bankrupt too.
Demo file: ./watopoly -load demo/doublekill.txt -testing < demo/doublekill.in

2.5 Horrible DC Tims Line experience

Now we will demonstrate the DC Tims Line. Run ./watopoly -load demo/tims.txt -testing -vText -vGraph
You can also type ./watopoly -load demo/tims.txt -testing < demo/tims.in
Our load file starts with 8 players. The 7th and 8th players P7 and P8 have been stuck at DC Tims Line for 2 turns already, P4, P5, P6 have just been sent to Tims Line, whereas P1, P2, P3 are not in Tims Line.

P1 will take its turn. Type "roll 10 20" to move to Go To Tims square and get sent to Tims Line. Type "next".
P2 will take its turn. Type "roll 1 1" to move to the SLC square. Type "1" to give up the cup.
Type "7" to be sent to DC Tims Line by the SLC square. Type "next".
P3 will take its turn. Type "roll 3 3" then "roll 1 1" then "roll 1 1" to roll 3 consecutive doubles.
As a result, P3 will be sent to DC Tims Line. Type "next".

P4 will take its turn. Type "roll" to see the instructions given by DC Tims Line.
(Important: Do not type "roll num1 num2" at this point. Type the word "roll" only.)
Now type "1" to try to roll a double. This is testing mode, so we can type "3 3" to guarantee a double.
P4 will move forward 3+3=6 spots and arrive at LHI owned by P2.

But LHI is currently mortgaged, so no fee is paid to P2. Type “next” to end the turn.
Now P5 will take its turn. Type “roll” to see instructions. Type “2” to pay $50 fine to leave.
Now P5 can take its turn as usual. Type “roll 2 3” to go to Residence UWP owned by P3. Type “next”.
Now P6 will take its turn. Type “roll” to see instructions. Type “3” to pay a Roll Up the Rim cup to leave.
Now P6 can take its turn as usual. Type “roll 0 1” to go to RCH owned by P4. Type “next”.

Now P7 will take its turn. Type “roll” to see instructions. Type “1” to try to roll a double.
We are in testing mode, so type “1 2” to guarantee P7’s failure.
Since this is the third turn P7 is at DC, P7 has to leave. Type “2” to pay $50 fine.
P7 will then move forward 1+2=3 steps and arrive at DWE owned by P7 itself. Type “next” to end the turn.
Finally P8 will take its turn. Type “roll” to see instructions. Type “1” to try to roll double.
Now again, type “1 2” to guarantee P8’s failure. Now P8 has to leave, type “2” to pay $50 fine.
But P8 only has $30, so type “bankrupt” and wait to be auctioned.
Now you can type “-1” seven times to end the auction. Then it will be P1’s turn.
You can see that P1 was sent to Tims Line earlier, so P1 can try to leave using one of the three methods.
You can try whatever method you like, but we will stop the demo here to keep it short.

2.6 Bonus Features

We have implemented a graphical view. You can see this by running the command line argument -vGraph.
We have implemented the three house rules mentioned in watopoly.pdf:

- Money collection from Tuition, Coop Fee, and DC Tims Line goes to the center of the board. Landing on Goose Nesting gives a Player all the money in the center.
- The above except that the center starts with $500. This is replaced if the money is collected.
- Landing on “Collect OSAP” doubles the amount received

Run the program with the command line argument -rule1 if you want to activate the first house rule. Similarly, run with -rule2 or -rule3 if you want to activate the second or third house rule.
We have also implemented the “even build” rule mentioned in watopoly.pdf. Run the program with -rule4 if you want to activate it. Any combination of these house rules can be activated at the same time.

First we will demonstrate the first rule. Run ./watopoly -load demo/house1.txt -testing -rule1 -vText -vGraph
Or alternatively, run ./watopoly -load demo/house1.txt -testing -rule1 < demo/house1.in
Initially it is player P1’s turn. Type “roll 2 2” to advance to the Tuition square.
Then type “2” to pay 10% of P1’s total net worth which is $114.
Then type “roll 16 18” to go to the Coop Fee square and lose $150. Type “next” to end the turn.
Now it is P2’s turn. Type “roll” to see instructions given by DC Tims Line.
Type “2” to pay $50 fine to leave. Then type “roll 0 1” followed by “next” to end the turn.
Now it is P3’s turn. Type “roll 10 10” to advance to the Goose Nesting square.
You will see that P3 received $114 + $150 + $50 = $314 as expected.

We will show other 3 rules. Run ./watopoly -load demo/house2.txt -testing -rule2 -rule3 -rule4 -vText -vGraph.
Or alternatively, run ./watopoly -load demo/house2.txt -testing -rule2 -rule3 -rule4 < demo/house2.in
Initially P1 takes its turn. Type “roll 2 2” to go to Tuition square. Type “1” to pay $300 tuition.
Then type “roll 7 9” to go to Goose Nesting and collect $500 + $300 = $800 dollars. Type “next”.
(Notice that due to the second house rule, the centre initially starts with $500.)
P2 takes its turn. Initially P2 starts at DC. Type “roll 2 2” to pass over OSAP and collect $200.
Now type “roll 37 0” to directly land on OSAP and receive $400 as expected. Type “next”.

P3 takes its turn. P3 controls both ML and AL and thus the Arts1 monopoly.
At present ML has been improved 3 times but AL has been improved 4 times.
Type “improve AL buy” to see that P3 cannot improve AL at present due to the “even build” rule.

Type “improve ML buy”, then type “improve AL buy” to see that now P3 is allowed to improve AL.
Now type “improve ML sell” to see that P3 cannot sell ML’s improvement due to “even build” rule.
Type “improve AL sell”, then type “improve ML sell” to see that now P3 is allowed to sell ML’s improvement.
Finally type “roll 11 9” to go to Goose Nesting and see that all money has been collected by P1 already.