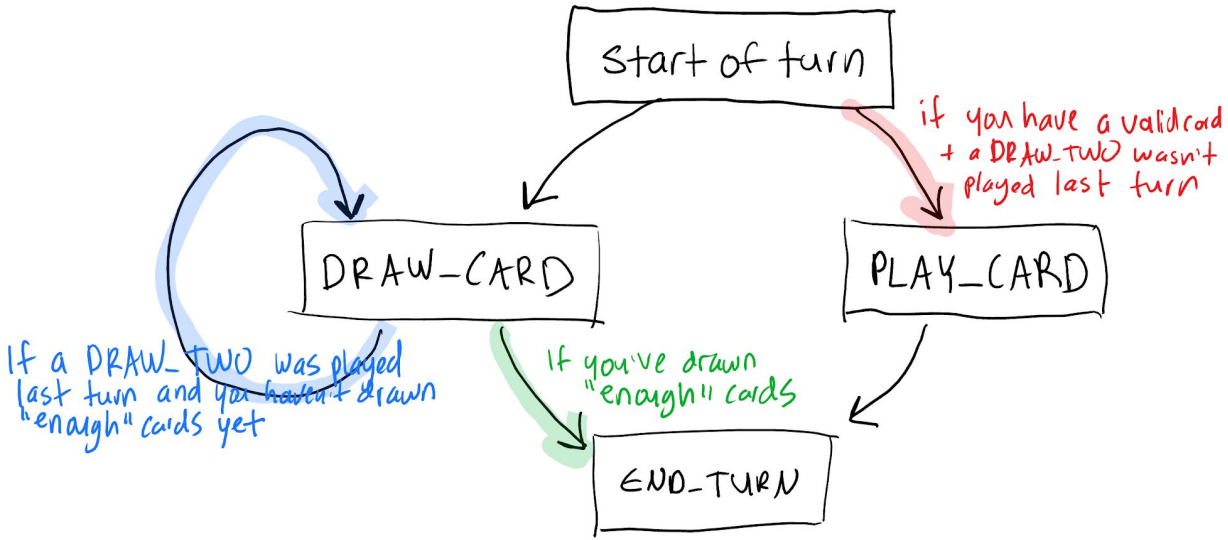
COMP1511: Assignment-2

Session 2, 2018

Valid moves, by Andrew Bennett

- the arrows with no color over them are always valid,
- the arrows with colors over them have a condition that's described in that colored text



Correction:

Earlier in one of the replies I **incorrectly** mentioned :

> you do not need to set anything, it is up to the game runner (supplied by us) to detect that there is only one card in the discard pile or that the last payer do not have any more card. I hope this clarifies your point.

You **do need to implement** the following function in your **Game.c**, that properly answers the above question.

```
// Check the game winner.
//
// Returns NOT_FINISHED if the game has not yet finished,
// 0-3, representing which player has won the game, or
// NO_WINNER if the game has ended but there was no winner.
int gameWinner(Game game);
```

Testing your player.c

- You can test your player.c against the three players (provided) by following the steps below.

- **Copy** the following files (available on Thursday 18/Oct) to your working directory:

```
% cp  /home/cs1511/public_html/18s2/assigns/ass2/runner/GameRunner.o     .
% cp  /home/cs1511/public_html/18s2/assigns/ass2/runner/Game.o    .
% cp  /home/cs1511/public_html/18s2/assigns/ass2/runner/Card.o  .
% cp  /home/cs1511/public_html/18s2/assigns/ass2/runner/player0.o     .
% cp  /home/cs1511/public_html/18s2/assigns/ass2/runner/player1.o    .
% cp  /home/cs1511/public_html/18s2/assigns/ass2/runner/player2.o    .
```

Testing your player.c

You can create a **GameRunner** by adding your player.c using the following command:

```
% dcc -o GameRunner player.c GameRunner.o
        Game.o Card.o player0.o player1.o player2.o
```

You can run GameRunner as shown below:

```
% ./GameRunner
```

If you want to **pause** moves, provide command-line argument **"--wait"** as shown below:

```
% ./GameRunner --wait
```

- If your player can successfully play a game and perform reasonably well (you don't need to win!), you will be awarded full marks for automarking of this part.

How will your `testGame.c` be marked?

- We will measure **code coverage** of your **testGame.c** that will indicate how much of the referenced Game.c code was executed while testing, using your testGame.c

- A tutor will also award subjective marks.