CS100 HW4 Hints

How can I get started?

- Look through the PDF document.
- Try to play this game first, and then you can realize more about game rules.
- Try to read the code and comments in the skeleton.

Recommended implement order

- First, directly compile the given skeleton and run it.

```
wuty@xxx ~/hw4package$ ./a.out
..........
..........
..........
You have 1 lives, and your score is 0.
make a move (a, b, c, u, d, g):
a



[1]    33736 segmentation fault  ./a.out
```

  - It will print the grids and let you to make a choice, but after you choose an operation, It will crushed immediately. On Unix like systems (like Linux or MacOS), you'll receive a `segmentation fault`, and on Windows, it will also crushed and be killed immediately by your Windows system.

- Second, Implement `Fmart::Fmart()` and `Fmart::~Fmart()`.

```
wuty@xxx ~/hw4package$ ./a.out
..........
..........
..........
You have 1 lives, and your score is 0.
make a move (a, b, c, u, d, g):
a



..........
..........
..........
You have 1 lives, and your score is 0.
make a move (a, b, c, u, d, g):
b
```

```
..........
..........
..........
You have 1 lives, and your score is 0.
make a move (a, b, c, u, d, g):
u



..........
..........
..........
```

  - As shown above, it will be an infinite loop if your implementation is correct.
  - Whatever you choose, the same empty grid will be printed.

- Then, modify the `display` part. By now, you can just set the player's position to be a character like `P`, regardless of what meal in his/her hand. And implement `Player::Player(Fmart* fp)`, `Player::~Player()`, `Player::row()`, `Player::col()` and `Player::move(int dir)` .

  - After you done these functions, add some code in `Fmart::play()` to make the `Player` know your option and move as you want.

  - Congratulations! If your implementation is correct, your player can move correctly by now.

```
wuty@xxx ~/hw4package$ ./a.out
..........
P.........
..........
You have 1 lives, and your score is 0.
make a move (a, b, c, u, d, g):
u



P.........
..........
..........
You have 1 lives, and your score is 0.
make a move (a, b, c, u, d, g):
d



..........
P.........
..........
```

```
You have 1 lives, and your score is 0.
make a move (a, b, c, u, d, g):
d



..........
..........
P.........
```

- Then, finish the `Player` part, and modify function `Fmart::play()` to make the program interacts correctly with game player.
- THen, finish `Student` part, and finish the game logics.
- Finally, debugging until it performs the same as the given sample, have fun with it!

> How much code will I need to write?

- Here's a summary of our reference solution, produced by the `diffstat` program. The final row gives total lines inserted and deleted; a changed line counts as both an insertion and a deletion.
- The reference solution represents just one possible solution. Many other solutions are also possible and many of those differ greatly from the reference solution.

```
hw4package/hw4_2_skeleton.cpp     |   151 ++++++++++++++++++++++++++
1 files changed, 151 insertions(+), 0 deletions(-)
```