

CPSC 312

Functional and Logic Programming

November 7, 2013

Administrative stuff

Midterm exam marks have been posted on Connect

Assignment 4 due 6:00pm tomorrow, Friday,
November 8

Project 1 due noon Tuesday, November 12

Administrative stuff review

There will be several homework assignments. They're not scheduled. They just happen when I decide to make them happen.

- 0/1/2/3 marking
- they all count - nothing is dropped
- each assignment carries the same weight
- we don't mark late assignments
(emphasis added)

Administrative stuff review

There will be two *team-based* programming projects...more details in the weeks to come, but...

Projects authored by only one person will be given a mark of zero.

We don't mark late projects, either.

(ditto)

Administrative stuff review

Tentative approach to computing final grades

Assignments	10%
Projects	20%
Midterm exam	30%
Final exam	40%

You have to earn at least 50% of the possible marks on the final exam, at least 50% of the possible marks on the midterm exam, and **at least 50% of the possible marks on each of the projects to obtain a passing grade in the course.**

(emphasis added yet again)

Administrative stuff review

Being late with your homework assignment costs you 2% of your final grade

Being late with your project is fatal

Do not be late submitting your project through handin

“My partner forgot”, “handin didn’t work”, “I thought it was due some other time”, are examples of the many explanations that will not help you

Do not email your project to me or the TAs...we lose things

From the previous lecture

Unification

Database manipulation

Database manipulation

```
?- assert(<some clause>).
```

Adds the clause to the corresponding procedure. Where?
In SWI-Prolog, the clause is added to the end of the procedure.

```
?- asserta(<some clause>).
```

Adds the clause to the beginning of the corresponding procedure.

```
?- assertz(<some clause>).
```

Adds the clause to the end of the corresponding procedure.

Database manipulation

```
?- retract(<some clause>).
```

Finds the first clause in the database that matches (i.e., unifies with) `<some clause>` and removes it from the database. (How that's implemented may vary from one Prolog implementation to another.)

Database manipulation

`assert` and `retract` work only on dynamic predicates. So far, everything you've created is a static predicate. Built-in predicates are also static. If you want a predicate to be modifiable, you need to declare it as such:

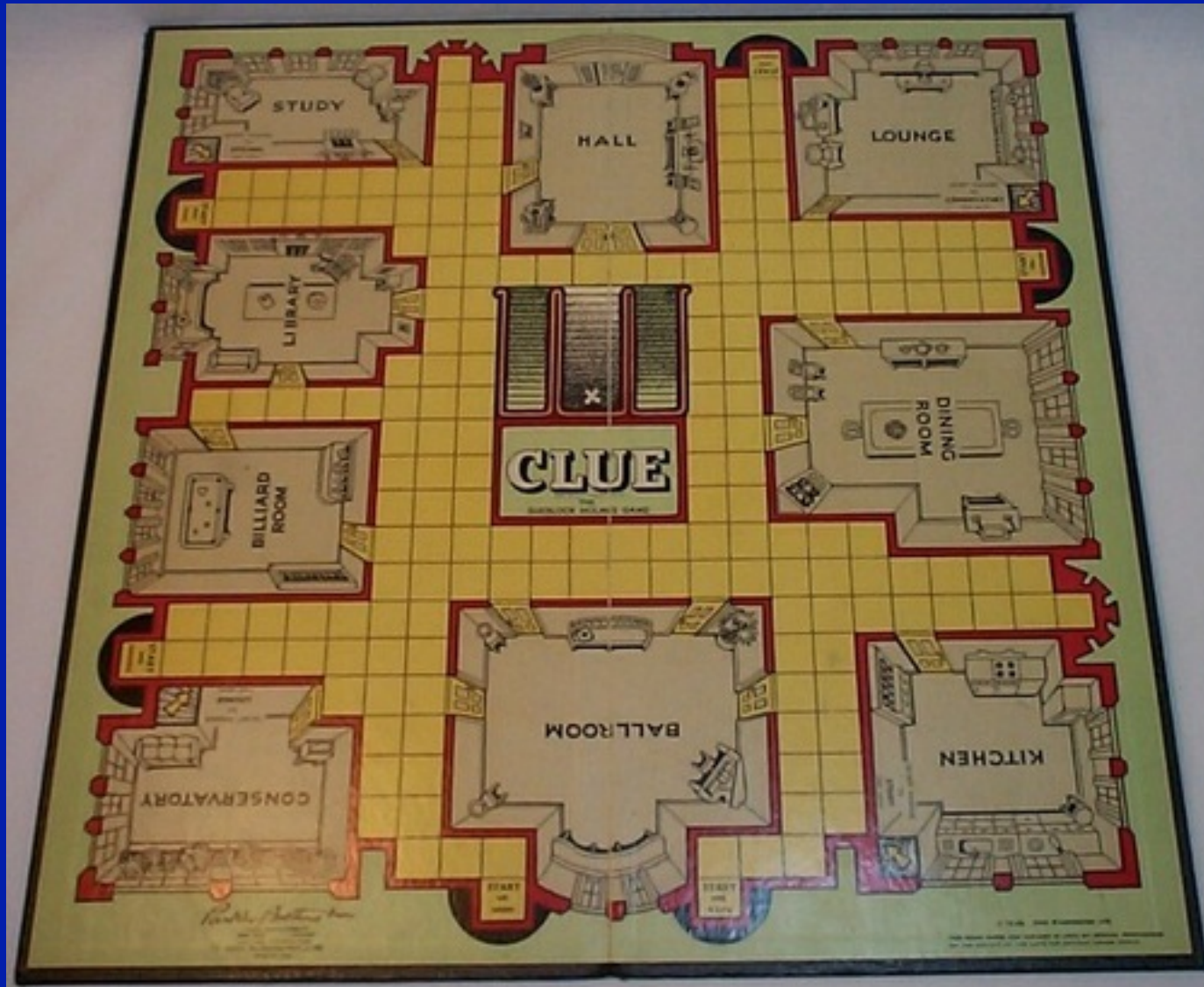
```
:- dynamic <pred_name>/<arity>, ... ,  
      <pred_name>/<arity>.
```

Here's an example:

Your term project

The classic detective game! In **Clue**, players move from room to room in a mansion to solve the mystery of: who done it, with what, and where? Players are dealt character, weapon, and location cards after one card from each card type is secretly placed in the confidential file in the middle of the board. Players must move to a room and then make a suggestion against a character saying they did it in that room with a specific weapon. The player to the left must show one of any cards suggested to the player who made the suggestion if it's in that player's hand. Through deductive reasoning each player must figure out which character, weapon, and location are in the secret file. To do this, each player must uncover what cards are in other players hands by making more and more suggestions. Once a player knows what cards the other players are holding they will know what cards are in the secret file and can make their final accusation. A great game for those who enjoy reasoning and thinking things out. (adapted from www.boardgamegeek.com)

Your term project



Your term project

The premise:

I need help to play the board game Clue (known as Cluedo in Europe). Your job is to build me a Clue Player Assistant using the Prolog programming language.

The more work that your program does for me while I'm playing -- in other words, the more it makes me look like an expert Clue player -- the better your program is.

Your term project

Minimum deliverables:

The program should allow me to:

- initialize the game setup easily
 - how many players?
 - the order of play (whose turn next?)
 - which cards am I holding?
 - and so on...

Your term project

Minimum deliverables:

The program should allow me to:

- keep track of my own play
 - I tell the program my suggestion.
("I suggested Mrs. White, the rope,
and the kitchen.")
 - I tell the program what I learned
("Player X showed me the rope" or
"Nobody showed me anything")

Your term project

Minimum deliverables:

The program should allow me to:

- see the contents of the database on demand
- know when to make an accusation

"Hey, Kurt, all that's left is Mr. Green,
the knife, and the library. Go for it dude!"

Your term project

Minimum deliverables:

The program should also have:

- easy-to-understand documentation describing what the program can and can't do, how to use it, how it works
- easy-to-use interface

Your term project

Minimum deliverables:

The program just described would be an electronic note pad, and would be worth a C or maybe a low B, depending on factors such as quality of programming, extensibility, documentation, interface, and maybe other factors. This is easy.

Your term project

The next level:

In addition to the minimum deliverables, the program should:

- take advantage of what can be inferred from the suggestions of other players
"Player X suggested Miss Scarlet, the candlestick, and the conservatory.
Player Y showed her a card."

Your term project

The next level:

In addition to the minimum deliverables, the program should:

- tell me which suggestion to make next
"You should suggest Col. Mustard, the wrench, and the billiard room."

Your term project

The next level:

This program would be much more than a note pad, and would earn a B to an A, depending on the factors noted previously.

Your term project

The A+ level:

The program should go beyond what's been described so far. Some suggestions:

- build models of what the other players might know and use the models to assess how close they might be to winning
- advise me to make a suggestion that might throw other players off
- other things you might think of

Your term project

This is a team project only. No solo efforts.
Teams consist of two members.

Your team's project is due by noon, Thursday, November 28. Use handin to submit your solution. For handin purposes, the assignment name is 'project2'.

Your term project

Bring a laptop with your program to class later that same day (November 28, our last class) for the in-class Clue program showdown. Your participation in this event factors into your mark for this term project.

Questions?

So how does that Clue game work?

If you've never played Clue, you probably don't understand much of what has been said.

It might help to have a set of rules in hand so you can familiarize yourself with the game...

So how does that Clue game work?

If you've never played Clue, you probably don't understand much of what has been said.

It might help to have a set of rules in hand so you can familiarize yourself with the game...

And even if you have played Clue, it may have been a long time since the last time you played. A refresher on how it's played couldn't hurt...

So how does that Clue game work?

If you've never played Clue, you probably don't understand much of what has been said.

It might help to have a set of rules in hand so you can familiarize yourself with the game...

And even if you have played Clue, it may have been a long time since the last time you played. A refresher on how it's played couldn't hurt...

Keep track of the time. We have to shut down early to have enough time to pack up.

As you play...

Don't just get bogged down in rules and trying to win. Think about the following:

- What kinds of things could a Prolog program do for you?
- What information will you need to represent in your program? How should you represent that information?
- How will you communicate that information to your program? How will the program communicate information to you?
- Are there students here who need a team mate?

Old school Clue vs. New school Clue

Old school

knife
candlestick
revolver
rope
lead pipe
wrench

kitchen
ballroom
conservatory
billiard room
library
study
hall
lounge
dining room

New school

knife
candlestick
pistol
rope
bat
ax

kitchen
patio
spa
theatre
living room
observatory
hall
guest house
dining room

I've changed new school games to look like old school, but the board layouts still have differences.