说明文档

一、 主题说明

本项目实现的是一个基于自然语言指令的文字冒险游戏——主题为剧情类密室逃脱。与一般的基于决策树的用选项推动剧情发展的文字冒险游戏不同，该游戏模拟了密室环境，可用自然语言指令（如“look” “pick up sth”）来执行一系列动作以探索房间，触发剧情，解谜最后逃脱或者‘死亡’。游戏中还按照现实世界的时间设置了时间限制，如第二个房间内超过五分钟未成功逃出则触发 bad ending。 游戏共四个房间，每个房间都有剧情线索和谜题，且前面的操作会影响最后的结局。

二、 使用说明

运行 play.pl，输入 play. 即可开始游戏，输入动作指令进行操作（游戏指令做了优化可以不加结尾的‘.’），输入 end 或 exit 退出游戏。界面指令如下：

```
?- play.
[Welcome to Thriller Paradise!]
[Script loading.....]

[Your name is Arthur seager.
You are a famous photojournalist. You have a good family, good job and good social status.
Until one Christmas night, you wake up from a coma and find yourself in a strange warehouse.
The last thing you can think of is to pick up your car in the parking lot after work...]

[GAME INSTRUCTIONS]
use 'help' to get instructions.
use 'look' to look around.
use 'inspect sth' to inspect for more information about the thing.
use 'get sth' to pick up sth and put it to your inventory.
use 'put sth' to take out sth from your inventory and put it on the ground.
use 'inventory' to list things in your inventory.
use 'open sth' to open the door or sth else if its key is in your inventory.
use 'get out' to get out the door if the door is open.
use 'state' to see your health point(HP) and physical point(FP).
use 'hint' to get some hints if you are stucked.
use 'end' to exit this game.
You should use natural language to execute some other actions in the game.

[dxl@Thriller Paradise]# look
This room seems to be a warehouse and its roof is about 12 meters or more above the ground.
The walls are solid, with a metal surface, so it's clear that you're unable to force its way out.
There is an unlocked door, which may be the only way out...
[Hint: You can use 'get out' to walk out the door. Use 'get sth' to pick up things.]
```

三、 实现说明

文件名	功能
Play.pl	游戏入口，定义了游戏中的主要事实，规则和逻辑。
Parse_input.pl	处理用户指令，采用了句法分析对用户的输入进行解析，以转化为标准的游戏指令。
Plot.pl	剧情事实，定义了游戏中所使用的剧情，提示信息等。
Util.pl	定义了 list 操作，规范化输出等一些操作规则。

游戏中的主要事实如下：

room 为游戏基本的四个房间（不可更改）；

door 定义了房间间的关系；

objects 定义了游戏中的物品；

position 定义了游戏中的物品(由于物品可拾取故该事实可更改)；

Locate 标识玩家所处位置；

inventory 为一个列表，为玩家的背包，可存储物品；

hp/fp 表示生命值/体能值，降为 0 时游戏结束。

Game_over 标识游戏状态；

Not_pickable 定义物品是否可拾取；

```
:-dynamic
door/3,
position/2,
locate/1,
inventory/1,
hp/1,
fp/1,
total_weight/1,
game_over/1,
on_chair/1,
not_pickable/1,
contain/2.
```

游戏运行的整体逻辑为：

```
play:-start_info,
      help_info,
      loop.

loop:-repeat,
      request_input(InputList),
      parse_input(InputList,CommandList),
      execute(CommandList),
      end_condition(CommandList),!.
```

基本规则及功能有（以下为基本操作，还有每个房间的特定操作等未列出）：

```
use 'help' to get instructions.
use 'look' to look around.
use 'inspect sth' to inspect for more information about the thing.
use 'get sth' to pick up sth and put it to your inventory.
use 'put sth' to take out sth from your inventory and put it on the ground.
use 'inventory' to list things in your inventory.
use 'open sth' to open the door or sth else if its key is in your inventory.
use 'get out' to get out the door if the door is open.
use 'state' to see your health point(HP) and physical point(FP).
use 'hint' to get some hints if you are stucked.
use 'end' to exit this game.
```

句法分析中，定义了一系列名动冠词，逻辑实现为：

```
translate([Verb,NP]) --> verb(Verb),np(NP).
translate([Verb]) --> verb(Verb).

np(NP)-->det,noun(NP).
np(NP)-->det,noun(NP),prep,det,noun(_).
np(NP)-->det,noun(NP),prep,noun(_).
np(NP)-->noun(NP).

det --> [the].
det --> [a].
det --> [this].
det --> [that].

prep --> [to].
prep --> [into].
```

工具文件中定义了列表的增删，反转等规则：

```
append([],X,X).
append([H|T1],X,[H|T2]):-append(T1,X,T2).

remove_list([],X,X).
remove_list([H|T1],X,T2):-remove(H,X,T3), remove_list(T1,T3,T2).

remove(ITEM,[ITEM|T],T).
remove(ITEM,[H|T1],[H|T2]):-dif(ITEM,H), remove(ITEM,T1,T2).
remove(_,List,List).

list_reverse(List, NewList):- list_reverse(List, [], NewList).
list_reverse([], X, X).
list_reverse([H|T], X, NewList):- list_reverse(T, [H|X], NewList).

output([]):-nl,!.
output([H|T]):- write(H),nl,output(T).

output_list([]):-write("."),nl,!.
output_list([H|T]):-write(H),write(" "),output_list(T).
```