**Wanted – A Game of Deduction**

## Presented by the group:

**Engelska parken**

## Participants:

**AthulGangadharan,**

**PriyanjaliSanyal,**

**Viktor Jönsson.**

### **Table of Contents**

|  |
| --- |
| 1. Project plan 2 |
|  |
| 1. Programming Manual 3 |
| * 1. - Program Purpose 3 |
| * 1. - Program Structure 4 |
| * 1. - User Interface 5 |
| * 1. - Program Code 6 |
| * 1. –Testing 13 |
|  |
| 1. User Manual 26 |

### **Project Plan**

### Group: Engelska parken

**NAME 19th Sept,2018 20-22nd Sept,2018 23-24th Sept,2018 25-26th Sept,2018**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Viktor Jönsson | Discussion about the Project Structure and Implementation method.  No. Of Hours:2hours | Create Murderer.  Create Person List.  No. Of Hours:5 hours | Testing Bug Fixing of respective modules.  No. Of Hours:6 hours | Documentation( Project Structure and Comments in the code)  No. Of Hours:5 hours |
| PriyanjaliSanyal | Discussion about the Project Structure and Implementation method.  **No. Of Hours:2 hours** | No of Trials Computation.  Create Suspect and Innocent List.  **No. Of Hours:6 hours** | Testing Bug Fixing of respective modules.  **No. Of Hours:6 hours** | Documentation(Project Plan, Program purpose).  **No. Of Hours: 4 hours** |
| AthulGangadharan | Discussion about the Project Structure and Implementation method.  **No. Of Hours: 2 hours** | Interface .  CleanDataBase Predicate.  **No. Of Hours: 4 hours** | Testing Bug Fixing of respective modules.  **No. Of Hours: 6 hours** | Documentation(User manual, User Interface, Test Cases).  **No. Of Hours: 6 hours** |

**PROGRAMMINGMANUAL**

**2.1 Program Purpose:**

A murder has been committed! The player’s job is to find out who the murderer is. At the start of the game, the program randomly selects a person to be the murderer, and the murderer is kept secret until the end of the game. The game begins. A new person is randomly selected and presented to the player in one of two ways: either as a “suspect” or as an “innocent.” This process is repeated until the player is ready to guess who the murderer is.

The purpose of the program is to deduce the person who has committed the murder

**2.2 Program Structure:**

Exit

Game

interface

****

End

**2.3 User Interface:**

The user interface of this game has 8 different options which are shown below along with the inputs & outputs.

1. Please type ”**game.**” To start the game i.e clears the database, generates the murderer and displays content of the menu.
2. To query the program for a new person, type the following: "**1.**" After yougive this input a new person is created and the combinations of the person are displayed to you.
3. To check the latest randomly selected person and to which category it was assigned, use the following: "**2.**" After you’ve given this input the last created person’s combination and his category is displayed.
4. To check the innocents so far, use the following: "**3.**". This input will display you the list of innocents created until now.
5. To check the suspects so far, use the following: "**4.**". This input will display you the list of suspects created until now.
6. To guess the combination, please use the following: "**5.**". After you give this input, the program will first ask you what is the age& gender. After this, the program will ask you what is the color and finally what was the weapon used to commit murder. After all the three inputs are given, you’ll know if the murderer you guessed is a murderer or not.
7. In case you do not wish to continue guessing and you want to exit the game, please enter **“6.”**
8. Debugging. Please use “**test.**” to debug the entire code.

**2.4 Program Code:**

:- dynamic murderer/3.

:- dynamic person/3.

:- dynamic suspect/3.

:- dynamic innocent/3.

:-use\_module(library(random)).

:-use\_module(library(lists)).

%Function to print information about the game.

game:- % starts the game i.e clears the database, generates the murderer and displays content of the menu.

cleanDataBase,

wanted,

go.

go:-

write('Choose an option from the below menu.'),

nl,nl,

write('------------MENU------------'),

nl,

write('1. Generate a new clue. '),

nl,

write('2. Check for the latest entry created. '),

nl,

write('3. Display the list of innocents so far. '),

nl,

write('4. Display the list of suspects so far?'),

nl,

write('5. Guess the murderer'),

nl,

write('6. Exit (In case you do not wish to continue guessing and want to exit )'),

nl, nl, nl,

write('Input your choice: '),

nl, nl,

read(Number),

check\_if\_valid\_input(Number).

check\_if\_valid\_input(Number):- % To check if the input is within the expected range i.e 1-6, 42 is cheat code.

integer(Number),

((Number>0,Number<7);(Number == 42)),

play\_game(Number).

check\_if\_valid\_input(Number):- % To check if the input is within the expected range i.e 1-6, if not go back to the menu and try again.

(atom(Number);

(Number<1;Number>6));

write('Invalid Input, try again!'),

go.

member\_of\_age\_gender(X):- % To check for valid input of age while guessing the murderer.

age\_gender\_list(Y),

member(X,Y).

member\_of\_age\_gender(X):- % To check for invalid input of age while guessing the murderer.

age\_gender\_list(Y),

\+member(X,Y), nl,

write('Incorrect input!'), nl,

write('Try again: '), nl,

read(Z), nl,

member\_of\_age\_gender(Z).

member\_of\_color(X):- % To check for valid input of color while guessing the murderer.

color\_list(A),

member(X,A).

member\_of\_color(X):- % To check for invalid input of color while guessing the murderer.

color\_list(A),

\+member(X,A), nl,

write('Incorrect input!'), nl,

write('Try again: '), nl,

read(Z), nl,

member\_of\_color(Z).

member\_of\_weapon(X):- % To check for valid input of weapon while guessing the murderer.

weapon\_list(A),

member(X,A).

member\_of\_weapon(X):- % To check for invalid input of weapon while guessing the murderer.

weapon\_list(A),

\+member(X,A), nl,

write('Incorrect input!'), nl,

write('Try again: '), nl,

read(Z), nl,

member\_of\_weapon(Z).

play\_game(1):- % Loop to run the gane as per user selected choices.

get\_person, nl,

go.

play\_game(2):-

latest\_person, nl,

go.

play\_game(3):-

list\_innocents, nl,

go.

play\_game(4):-

list\_suspects, nl,

go.

play\_game(5):-

write('Time to guess who the murderer is!'), nl,

write('The different options to choose from are: '),nl,

listing(age\_gender\_list/1),

write('What age and gender does the murderer have?'), nl,

read(X), nl,

member\_of\_age\_gender(X),

listing(color\_list/1),

write('What color of clothing does the murderer have?'), nl,

read(Y),

member\_of\_color(Y),

listing(weapon\_list/1),

write('What weapon did the murderer use?'),

read(Z),

member\_of\_weapon(Z),

guess\_murderer(X,Y,Z),

write('The game will now start over from the beginning.'), nl, nl, nl, nl,

game, nl.

play\_game(6):-

write('Good bye, the game ends now!'),

cleanDataBase.

play\_game(42):-

listing(murderer/3),

go.

%The different age\_gender attributes.

age\_gender\_list([youngman,oldman,youngwoman,oldwoman]).

%The different color attributes.

color\_list([brown,yellow,blue]).

%The different weapon attributes

weapon\_list([knife,poison,gun]).

%Predicate to get all innocents.

list\_innocents:-

listing(innocent/3).

%Predicate to det all the suspects

list\_suspects:-

listing(suspect/3).

%Predicate to get the last person called and if its a suspect or innocent.

latest\_person:-

murderer(A,B,C),

person(X,Y,Z),

(X \== A,Y \== B, Z \== C),

write('Innocent: '),

listing(person/3), nl.

%Predicate to get the last person called and if its a suspect or innocent.

latest\_person:-

murderer(A,B,C),

person(X,Y,Z),

(X == A; Y == B; Z == C),

write('Suspect: '),

listing(person/3), nl.

%Predicate to call the game to create a murderer to start the game.

wanted:-

create\_murderer,

murderer(A,B,C).

%Predicate to call the game to create a new person.

get\_person:-

create\_person,

latest\_person.

%Predicate to reset the game.

cleanDataBase:-

retractall(murderer(A,B,C)),

retractall(suspect(X,Y,Z)),

retractall(innocent(X,Y,Z)),

retractall(person(X,Y,Z)).

%Function to calculate how many persons youve generated.

no\_of\_tries(Trials):-

findall(suspect(X,Y,Z), suspect(X,Y,Z), Suspect\_list),

length(Suspect\_list,Snum),

findall(innocent(X,Y,Z), innocent(X,Y,Z), Innocent\_list),

length(Innocent\_list,Inum),

Trials is Snum+Inum.

%The function to guess the murderer and end the game. Tells you if you won or lost and how many persons you generated.

guess\_murderer(A,B,C):-

murderer(X,Y,Z),

(A==X, B==Y, C==Z),

write('Murderer caught, good job!'),

nl,

no\_of\_tries(Trials),

write('You guessed it right within: '),

write(Trials),

write(' tries.'), nl,

cleanDataBase.

%The function to guess the murderer and end the game. Tells you if you won or lost and how many persons you generated.

guess\_murderer(A,B,C):-

murderer(X,Y,Z),

(A\==X; B\==Y; C\==Z),

write('Wrong suspect, game over!'),nl,

write('The murderer was: '),

write(murderer(X,Y,Z)),

no\_of\_tries(Trials),

write('You exhausted: '),

write(Trials),

write(' trial.'), nl,

cleanDataBase.

%Function to get a random element from a list.

randomElement(List,Element):-

length(List,Length),

Max is Length + 1,

random(1,Max,Z),

nth1(Z,List,Element).

%Function to figure out if a person should be considered to be innocent or suspect, and then assert said person to respective innocent/suspect list.

check\_for\_suspect\_or\_innocent(Element1,Element2,Element3):-

murderer(A,B,C),

(Element1 \== A,Element2 \== B, Element3 \== C),

findall(innocent(X,Y,Z), innocent(X,Y,Z), Innocent\_list),

\+member(innocent(Element1,Element2,Element3),Innocent\_list),

assert(innocent(Element1,Element2,Element3)).

%Function to figure out if a person should be considered to be innocent or suspect, and then assert said person to respective innocent/suspect list.

check\_for\_suspect\_or\_innocent(Element1,Element2,Element3):-

murderer(A,B,C),

(Element1 == A; Element2 == B; Element3 == C),

findall(suspect(X,Y,Z), suspect(X,Y,Z), Suspect\_list),

\+member(suspect(Element1,Element2,Element3), Suspect\_list),

assert(suspect(Element1,Element2,Element3)).

%The function to create the murderer called from the "wanted" predicate.

create\_murderer:-

age\_gender\_list(A\_list),

color\_list(C\_list),

weapon\_list(W\_list),

randomElement(A\_list, A),

randomElement(C\_list, B),

randomElement(W\_list, C),

assert(murderer(A,B,C)).

%The function to create a person called from the "get\_person" predicate.

create\_person:-

retractall(person(F,G,H)),

maximum\_combinations(B),

no\_of\_tries(Trials),

Trials < B,

age\_gender\_list(A1\_list),

color\_list(C1\_list),

weapon\_list(W1\_list),

randomElement(A1\_list, A1),

randomElement(C1\_list, B1),

randomElement(W1\_list, C1),

\+murderer(A1,B1,C1),

check\_for\_suspect\_or\_innocent(A1,B1,C1),

assert(person(A1,B1,C1)). % Push it to the create\_person\_List.

%The function to create a person called from the "get\_person" predicate.

create\_person:-

%I will check the combination if all possible combinations are reached.

maximum\_combinations(B),

no\_of\_tries(Trials),

Trials < B,

create\_person.

%The function to create a person called from the "get\_person" predicate.

create\_person:-

%I will check the combination if all possible combinations are reached.

maximum\_combinations(B),

no\_of\_tries(Trials),

Trials >= B,

write('No more available suspects').

%Sets the maximum number of combinations for persons based on the number of attributes. Real number is 36 but the list starts at 0 so we set max number to 35 here.

maximum\_combinations(A):-

color\_list(X),

length(X,Cnum),

age\_gender\_list(Y),

length(Y,Anum),

weapon\_list(Z),

length(Z,Wnum),

A is (Cnum\*Anum\*Wnum)-1.

%Debugging tool.

test:-

cleanDataBase,

create\_murderer,

create\_person,

create\_person,

create\_person,

create\_person,

create\_person,

create\_person,

create\_person,

create\_person,

create\_person,

create\_person,

create\_person,

create\_person,

create\_person,

create\_person,

create\_person,

create\_person,

create\_person,

create\_person,

create\_person,

create\_person,

create\_person,

create\_person,

create\_person,

create\_person,

create\_person,

create\_person,

create\_person,

create\_person,

create\_person,

create\_person,

create\_person,

create\_person,

create\_person,

create\_person,

create\_person,

create\_person,

create\_person,

create\_person,

create\_person,

create\_person,

create\_person,

create\_person,

create\_person,

create\_person,

listing(person/3),

listing(innocent/3),

listing(suspect/3),

listing(murderer/3).

**2.5 Testing:**

|  |  |
| --- | --- |
| Test Case\_ID | 01 |
| Date | 24/09/2018 |
| Tester | Athul |
| Test Description | To Check if the interface menu works fine. |
| User Input | game. |
| Expected Result | It should trigger list of available options of the same, clean the database & generate and hide the murderer from the player. |
| Actual Result | Query the program for a new person,  Check the latest randomly selected person and to which category it was assigned,  Check the innocents so far,  Check the suspects so far,  Guess the combination. |

|  |  |
| --- | --- |
| Test Case\_ID | 02 |
| Date | 23/09/2018 |
| Tester | Viktor |
| Test Description | To check if the murderer is created. |
| User Input | 1. |
| Expected Result | Create a murderer . |
| Actual Result | For debugging purpose we printed the murderer. Now a murderer gets created in memory. |

|  |  |
| --- | --- |
| Test Case\_ID | 03 |
| Date | 23/09/2018 |
| Tester | Viktor |
| Test Description | Check the last person generated in the game. |
| User Input | 2. |
| Expected Result | Person(Age\_gender,Color,Weapon). |
| Actual Result | Person(Age\_gender,Color,Weapon). |

|  |  |
| --- | --- |
| Test Case\_ID | 04 |
| Date | 24/09/2018 |
| Tester | Priyanjali |
| Test Description | Generate a new Person. |
| User Input | 1. |
| Expected Result | A new entry is created in Person List. |
| Actual Result | Person(age\_gender,Color,Weapon). |

|  |  |
| --- | --- |
| Test Case\_ID | 05 |
| Date | 24/09/2018 |
| Tester | Priyanjali |
| Test Description | List all the innocents generated so for in the game. |
| User Input | 3. |
| Expected Result | List of all the persons which are innocent. |
| Actual Result | Person(age\_gender,Color,Weapon).  .  N Persons which are innocent |

|  |  |
| --- | --- |
| Test Case\_ID | 06 |
| Date | 24/09/2018 |
| Tester | Athul |
| Test Description | List all the suspects generated so for in the game. |
| User Input | 4. |
| Expected Result | List of all the persons which are innocent. |
| Actual Result | Person(age\_gender,Color,Weapon).  N Persons which are suspects. |

|  |  |
| --- | --- |
| Test Case\_ID | 07 |
| Date | 24/09/2018 |
| Tester | Athul |
| Test Description | Guess the murderer |
| User Input | 4. |
| Expected Result | Match the attributes given by the user with the actual murderer |
| Actual Result | Match the attributes given by the user with the actual murderer and decide if the user has won or lost the game. |

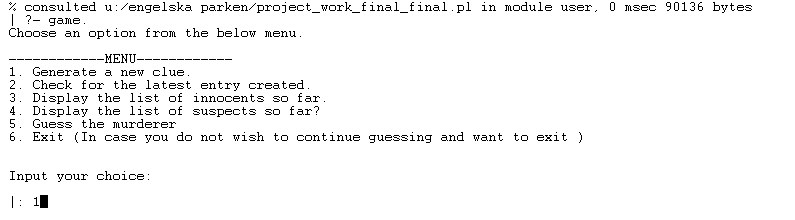
|  |  |
| --- | --- |
| Test Case\_ID | 08 |
| Date | 25/09/2018 |
| Tester | Viktor |
| Test Description | Enter a person and check if according to its attributes , it is inserted properly in either suspect or innocent list. |
| User Input | Wanted with debug prints for murderer.  create\_person.  And then list\_suspect or list\_innocent according the attributes match. |
| Expected Result | If two attributes of the person matches with murderer push it to suspect list, else push it to innocent list. |
| Actual Result | Irrespective of the no. of attributes matching with the murderer the code was always populating innocent list. |
| SEVERITY OF DEFECT | Major |
| COUNTER MEASURE | The predicate **check\_for\_suspect\_or\_innocent** had logical error. Correcting it made the logic work. |

|  |  |
| --- | --- |
| Test Case\_ID | 09 |
| Date | 05/10/2018 |
| Tester | Priyanjali |
| Test Description | Exit |
| User Input | 6. |
| Expected Result | Prints the attributes of a murderer , clear the database and exit the game. |
| Actual Result | Prints the attributes of a murderer , clear the database and exit the game. |

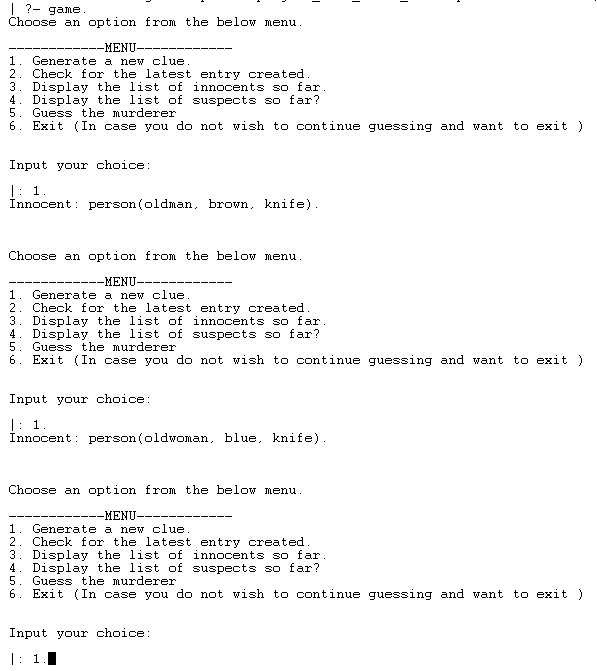
### **USEr manual**

This game has 8 different options which are shown below along with the screenshots of inputs & outputs.

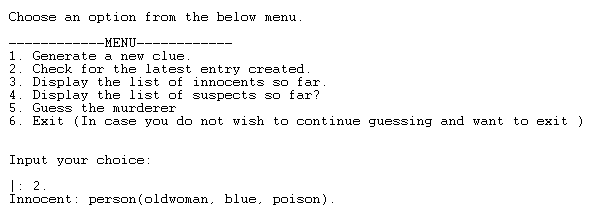
1. Please type ”game**.**” To display information about the game and the available options in the menu. After you’ve given the input you’ll be displayed with 6 options along with instructions in the menu.



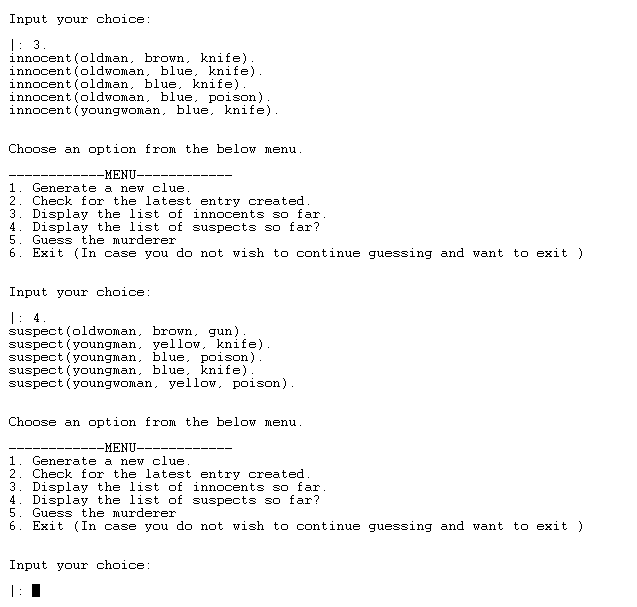
1. To query the program for a new person, type the following: "**1.**" After you give this input a new person is created and the combinations of the person are displayed to you.



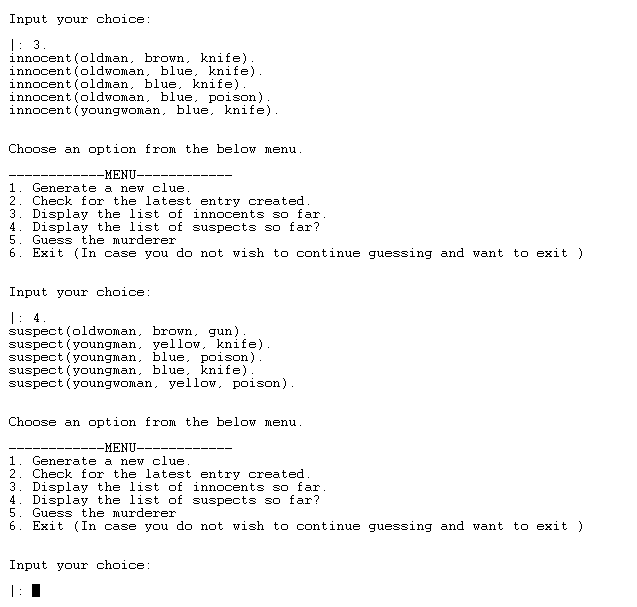
1. To check the latest randomly selected person and to which category it was assigned, use the following: "**2.**" After you’ve given this input the last created person’s combination and his category is displayed.



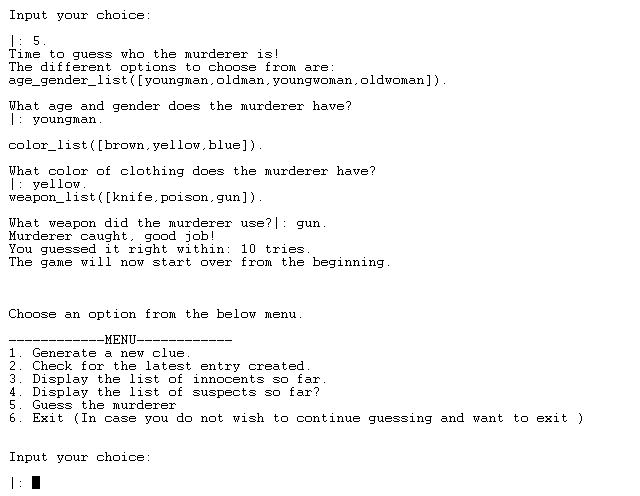
1. To check the innocents so far, use the following: "**3.**". This input will display you the list of innocents created until now.



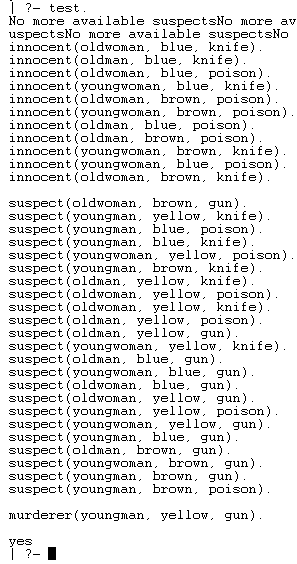
1. To check the suspects so far, use the following: "**4.**". This input will display you the list of suspects created until now.



1. To guess the combination, please use the following example "**5.**". After you give this input, it ‘ll show you if the murderer you guessed is a murderer or not.



1. Debugging. Please use “**test.**” to debug the entire code.



Tack!