HÁSKÓLINN Í REYKJAVÍK
REYKJAVÍK UNIVERSITY

Murder She Programmed: Dynamic Plot Generating Engine for Murder Mystery Games

María Arinbjarnar

Computer Science

Thesis Advisor:
Yngvi Björnsson

School of Science and Engineering

Autumn 2005

Abstract

There is an increased demand for more interactive and dynamic single-player computer games. Most currently proposed solutions usually focus on multi-linear narrations. The problem with those solutions is that they do not actually create a dynamic game, they only increase player choices; the games themselves continue to follow a fairly strict storyline. In this article a new game engine is proposed that dynamically creates new game plots for a murder mystery based game. The game engine uses Bayesian networks to create a new plot based on a probability map of a typical murder mystery novel. To abstract plot elements for the construction of the Bayesian network a unique morphology similar to the morphology of the Russian structuralist Vladimir Propp is created. A directed graph, which is inferred from the resulting Bayesian network, is used as a knowledge base for a murder mystery game. This game sets up a complete and consistent murder mystery that is solvable with logical inference. Each new game generates a unique plot, a murder mystery with all the attributes that are needed for such a plot to be logically consistent, coherent, and complete, and make sense to the human perspective.

This game engine is responsive to preset constraints rather than a pre-authored narration, which opens up numerous possibilities for integration in computer games where the constraints can be manipulated by interactions from game events and player interaction.

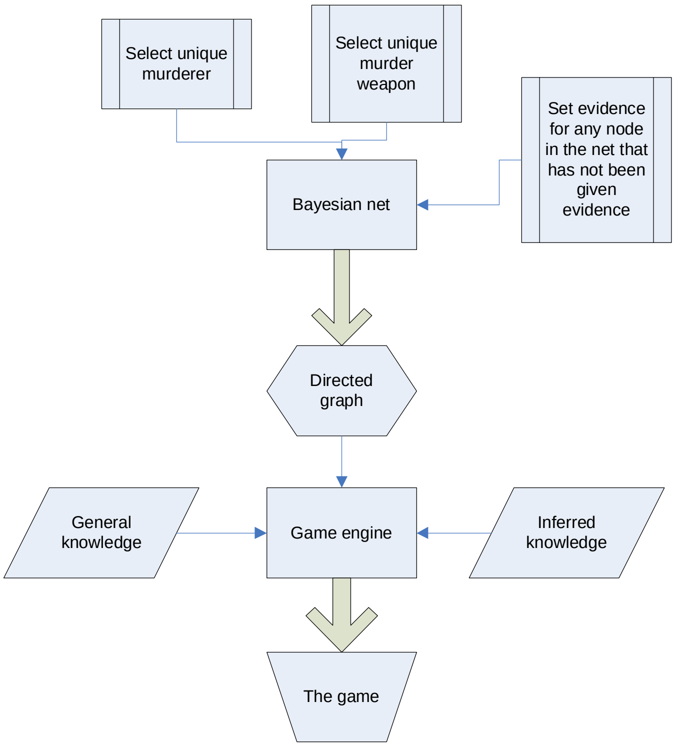
1. Introduction

Quite a large portion of the game market focuses on producing games with a high degree of narration. These games, especially Role Play Games (RPG), are highly interactive with very advanced game engines offering all the best in combat, graphics and sounds that is available on the market. These games are specifically trying to meet a demand for non-linear game interaction. Their answer has been to offer increasingly multi-linear narrations. The problem with that is that the games are still linear: The player is still forced to follow a set line or lines of preset prerequisites to advance or complete the game. This increases the danger of the player becoming stuck because he did not follow the set storyline. More importantly, it ruins the replayability of the games. It is also not responsive to player actions or role development in the game.

I propose a dynamic plot generating engine as a first step in solving this problem. The engine utilizes a Bayesian network, which is a probability-driven causally connected network, to generate a new murder mystery plot for each new game initialized. The Bayesian network is based on a unique morphology similar to the morphology of the Russian structuralist Vladimir Propp (Propp 1968). This morphology abstracts necessary events and actions to create complete and consistent murder mystery plots as they appear in typical murder mystery novels. The Bayesian network uses the morphology to abstract necessary events and actions to create a complete plot. As can be seen from the following diagram, the only functions implemented now are those that ensure a unique murderer and unique murder weapon.

A murder mystery game is used as a test base for this plot generating engine. The game uses a directed graph as its knowledge base. This directed graph results in a one on one relation to the Bayesian network where every node has been given evidence. When a node is given evidence in a Bayesian network then one of its attributes has been given 100% value. This murder mystery game sets up a complete and consistent murder mystery that is solvable with logical inference. Completeness means that the game has a logical solution whereas consistency ensures that there is no ambiguity, that is that the mystery has at most one solution. The game reads the directed graph and first it applies to it the general knowledge that is read in from a text file. The general knowledge describes objects, actions and persons in the game. At the start of the game the player has access to this knowledge at will. Next the game engine reads the inferred knowledge from a file and applies it to the graph. The inferred knowledge details all inference that can be drawn from the initial Bayesian net. The general knowledge and inferred knowledge are in direct one to one relationship to the

Bayesian net, meaning that for each attribute there is only one knowledge rule, either general or inferred. It is then a matter of game implementation how the knowledge base is shown to the player



Further I identify 18 rules that are used as constraints to ensure the consistency and completeness of each plot generated. These rules are adapted from rules that murder mystery writers themselves have put forth as guidelines to making good mysteries. These rules come into the functionality applied to the Bayesian net and to the implementation of the game itself, both in what is shown to the player at each stage and to the actual content of the knowledge base.

This plot generating engine solves the issue of replayability as it guaranties that the player always gets a new plot for each new game. It also tackles the problem of linearity as it is responsive to preset constraints instead of pre-authored narrations. This makes the plot generating engine adaptable to all kinds of game settings that require a plot to structure game play and game interaction. Simply by adapting the morphology to the respective story type and making the rules constraining the net at least partially responsive to the respective character played by the player the game engine is able to create new plots for the player to tackle.

The paper is organized a follows. In section 2 I discuss related work, how it contributes to this research and what I am adding and doing differently than others. In section 3 I form the actual problem statement and research question. In section 4 I describe in some detail my contribution and implementation of each of the components used, the Bayesian net, the mystery plot with its 18 rules to define it and the knowledge base. In section 5 I present the result and follow with discussion of this work and future work in section 6 and conclusions in section 7.

2. Related Work

There are numerous experiments dealing with automatic plot generations, they nearly always try to construct a story or narration of some sort. An example is the Open ended Proppian Interactive Adaptive Tale Engine (OPIATE) (Fairclough and Cunningham 2004). The engine creates interactive narrations and is in a large part based on Propp's morphology (Propp 1968). The most interesting part is a gossip system that connects the

Non-Player Characters (NPCs) together and spreads news and opinions about the current player character in respect to his actions. Another experiment is The Story Engine also based on Propp's morphology (Schneider et al 2003). It is basically an engine to assist in writing multi linear stories for RPG. It uses the morphology to avoid deadlocks and to assist in creating a literary sound narration. A third system that is worth mentioning is a Character Focused Narrative Generation (Riedl and Young 2003). There the authors have a set of actor-agents that each play out a character. The actors plot a few moves onwards with assistance of a director agent and a "blackboard". Blackboard is a message board to post messages between agents. The actor's movements are causally dependent to each other and this restricts the plot development and creates seemingly unnecessary complications.

The problem with such forced narration is that the chance of a dead end or some other major logical conflict is very high and in some cases practically unavoidable. This problem is widely recognized and some solutions have been offered. Most involve a net of some sort. Verbrugge (Verbrugge 2002) proposes a narrative flow graph that is based on a Petri net that enforces constraints that he claims are necessary for a Narrative game. Of course he has a perfectly valid point if the aim is to enforce a predetermined narration but it is not appropriate in an open-ended dynamic game world. (Lindley and Eldahari 2002) also discuss this problem in some detail. They propose a couple of solutions. The former is to apply Boyce-Codd Normal Form (BCNF) to a causal net and the latter is to apply object orientation and loose coupling. Their ideas are very sound but as with (Verbrugge 2002) they enforce a preset narration.

By forcing such a predetermined narration onto the game world and individual players the game is essentially trying to enforce a different past on the player than the one actually played out. It is as if a student would sign in to computer science department at a University and happily go about his studies until he finds out that he can not graduate because he took a course in Calculus before entering the CS department. This would hardly happen simply because causality is not reversed in real life. To gain a degree from a University you need to pass certain sets of tests and exercises to prove sufficient knowledge. There is seldom an issue of how much knowledge you gain from outside the University. This example seems far-fetched but is essentially an accurate description of what many game developers are enforcing.

For example let's imagine a wizard having a vital key to enter some important place. Frequently a game would have a wizard give a quest like killing a beast and retrieving a vial of its blood. As discussed by Lindley and Eldahari (2002) many games have a problem if you happen to kill the beast before you meet the wizard for the first time, as then he can not give you his quest. The problem here is with perspectives and backwards causality. The player is supposed to have met the wizard before killing the beast. Which is equivalent to the student registering at the University before taking a course in Calculus. When put into perspective it becomes obvious that this is backwards logic. If the player kills the beast then the beast is dead and if he meets the wizard afterwards and can give him the vial of blood as proof, then he should be rewarded, just as it does not matter when and where you learnt calculus. If you pass the tests and have mastered the other subjects at a specific University then you should be able to graduate from that University.

The following two experiments do not attempt to force a specific storyline onto the game or make actions of characters in the game causally dependent to each other.

The non-linear interactive storytelling game engine (NOLIST) (Bangsø et al. 2004) is the one that most closely resembles my proposal. There they utilize a Bayesian network to determine the culprit of a murder mystery, the Bayesian network is dynamically changing in response to actions and observations made by the player. It is not preset. It uses player's moves and logical inference of the net to determine the culprit. For example if the player finds a body and a gun lying beside the body then the probability that the murder victim was shot with the gun increases. They do this to be able to construct an engine that will create a dynamic storyline in respect to player actions and choices. The plot and the culprit are not known by the game engine in the beginning but are determined in the course of the game. Thus NOLIST creates the past in reaction to player interaction. Although NOLIST is highly adaptive to player interaction it might actually be too reactive. Players are likely to play games in similar manners each time and are thus not good at surprising themselves.

The Virtual Story Teller (Theune et al. 2004, Theune et al. 2002) deploys a director agent and character agents. The director agent's job is to direct the character agents towards meaningful goals to enforce an emerging storyline, but the character agents are very independent and their actions are controlled by their feelings and by the priority of their goals, both of whom are continually changing in respect to actions and input from their surroundings. The director agent can forbid or accept an action chosen by a character but he can not command the character, which essentially keeps the character actions coherent and sensible for that character. The characters are given individual characteristics that have a direct influence on their feelings, reactions and response to input from other characters and the environment. They are also given goals, primary and secondary, that are the driving force of their actions. Thus a storyline emerges by the NPC's reactions to in-game events as they attempt to fulfil their goals. This has not been implemented in an actual game and it appears that the player would be hard pressed to be at the right place at the right time to interact intelligently with an NPC's and participate in the storyline.

Although these are both excellent steps towards solving the problem they still lack a connection to literature. Connection to literature is necessary to implement some conflict between NPC's that calls for interaction by the user.

Many of those who implement some part of Propp's morphology (Propp 1968) have adapted it to their needs. Propp observed that the Russian fairy tales, classified by Aarne index 300 to 749 (Aarne 1911), are constructed of repetitive themes where actions and functions stay the same but the names and attributes of the persons change. In this case a function is defined as an act of character and viewed for its significance in the course of an action. This means that each function is independent from the rest of the narration and is only recognised for its effect on characters each time and for its part in completing an action. From his careful observations Propp was able to propose the following 4 rules for the makeup of the set of Russian fairy tales:

1. Functions of characters serve as stable, constant elements in a tale, independent of how and by whom they are fulfilled. They constitute the fundamental components of a tale.
2. The number of functions known to a fairy tale is limited.
3. The sequence of functions is always identical.
4. All fairy tales are of one type in regard to their structure.

The temptation here is to apply Propp's morphology to computer games in an attempt to make a more coherent narration. This idea is very sound especially if the functions and roles that Propp laid out are adapted to the intended narrative. What you get is a coherent storyline that a player could follow but you do not get an open-ended game play unless it is ensured that none of the functions deployed is causally dependent upon another. This is a very important point in Propp's morphology. Propp numbers the functions and states correctly that no function may happen after a function of a higher number has happened. This is to ensure a coherent time line. Propp managed in this way to map a sort of unintentional logical timeline for the folk tales. This does not mean that the functions are causally dependent in any way. Instead they are not and Propp stresses this point repeatedly in his writings (Propp 1968). However the functions need to be coherent and logically related to create a narration.

My Dynamic Plot Generating Engine implements a morphology similar to Propp's morphology discussed previously and thus satisfies the need for a literary connection and a call for conflict. Additionally it can supply a new plot at each new game or, if it were a larger continuous game, at each new scene. This makes it highly responsive to changes in the game and to player interaction without losing connection to literature and the essentials of a good story.

3. Problem statement

The first step to solve the problem of linearity and lack of replayability in computer games is to be able to generate new plots in response to player actions and role development. The plot generated needs to be coherent, consistent and complete so that it meets the expectations of the human players. If there are obvious inconsistencies, logical or otherwise, then the player is likely to notice.

There are many problems that need to be address for this to be possible. In this research I develop a novel approach for automatically generating story plots based on a probabilistic representation of game knowledge in the form of a Bayesian net.

The research question addressed in this report is thus whether it is possible to use Bayesian nets as proposed to automatically generate coherent story plots for a murder mystery game.

4. Problem solution

A Bayesian network is primarily used for assisting in decision making; it is a causally connected probability network (Jensen 2001). To construct a sensible plot one needs to make a series of calculated decisions which is why I believe that a Bayesian network is perfect for plot construction. The plot that the Bayesian net constructs follows 18 rules (see "The Mystery Plot" below) to enforce coherence and to fulfil the usual expectations of a mystery plot. I chose the murder mystery as a testing ground because it is easy to set clear rules that can be used to measure the success of the experiment.

4.1. Constructing the Bayesian net

There are three main questions in the murder mystery:

1. Who is the murderer?
2. What did he use?
3. Why did he do it?

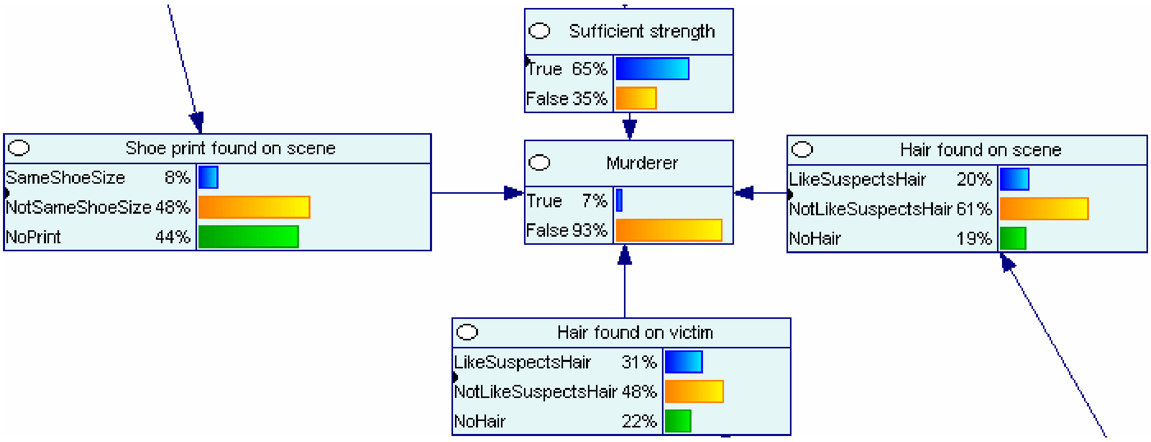
This solution only tackles the "who" and "how" part and leaves the motive implementation for a better time. The net is constructed around the characters, objects and clues with the murder itself as its center point. The program runs through the Bayesian net at the start of a new game. First the program needs to ensure that there is only one murderer and one murder weapon. This is done in the following manner.

The program randomly generates a name for each of the suspects and for each of the weapons. Sets the first suspect encountered in the Bayesian network and the first weapon as the murderer and murder weapon respectively, since the names are randomly generated it has been ensured that the murderer and murder weapon is randomly generated.

Then the program ensures that at least one element for each of the other suspects that is necessary to qualify him as a possible murderer is set to false and thus ensuring that no other suspects qualifies as a murderer. The same is done for the weapons. This function is based on the following famous words of Conan Doyle's famous detective Sherlock Holmes himself:

> *"When you have eliminated all which is impossible, then whatever remains, however improbable, must be the truth"* (Doyle 1960).

In the following example the *murderer* node will only be true if the attributes strength of suspect, hair found on victim, hair found on scene or shoe print found on scene if any match the suspect's hair or shoe print respectively. If any of these is false then *murderer* node will be false



A more detailed layout of the Bayesian net is in Appendix A.

Finally the program runs through the rest of the net, making decisions in relation to the probabilities that the net gives at each point.

The Bayesian net is used for the plot generation only. When the plot has been set for each game the following game logic for playing out the game does not need a Bayesian network, although it is handy to implement the game logic and this is what is done here.

To draw the net and implement it into a program I use GeNIe (Graphical Network Interface) and SMILE (Structural Modelling, Inference, and Learning Engine), http://genie.sis.pitt.edu/ .

4.2. The mystery plot

Those who enjoy mysteries know that a key element of a good mystery is a good plot. So what makes a good plot?

Many have written and discussed what makes a good mystery and what makes a good plot. Most agree that the plot needs to be logically sound, that there may never be any tricks played on the reader. A good mystery will keep the suspense high throughout the book but at the end when the reader is told the solution he should be able to read it again and see for himself that he could have discovered the answer early on, if he had correctly evaluated the clues.

Turning the detective story into a computer game or a purely logical puzzle is indeed not a new idea. As early as 1924 Austin Freeman recognized, in writing this essential makeup of the mystery:

"…, that a completely executed detective story is a very difficult and highly technical work, …, it is a work of ratiocination, demanding the power of logical analysis and subtle acute reasoning" (Freeman 1924).

This clearly shows the recognition of the algorithmic quality of the murder plot. It has also been identified that the plot does not need to be overly complicated or twisted; in fact some of the most famous plots had very simple solutions with clues readily apparent early on in the story. For instance the very famous incident of the "dog that did not bark", there Sherlock Holmes explains his conclusions with reference to a dog that does not bark when expected:

> "The Simpson incident had shown me that a dog was kept in the stables, and yet, though someone had been in and had fetched out a horse, he had not barked enough to arouse the two lads in the loft. Obviously the midnight visitor was someone whom the dog knew well." (Doyle 1960).

It seems to be that it is the readers themselves that disregard the obvious solution and thus create a large part of the suspense themselves.

The plot that the engine will generate should strictly follow the expectations that people have come to expect of a good mystery, and to that extent I have identified the following 18 rules that the plot must follow to be considered coherent to the human perspective. These rules are adapted and updated to modern times from the twenty rules of Van Dine 1928, the detection club oath from the detection club founded in 1928 by Anthony Berkeley (Berkeley 1928) and Knox's Ten Commandments published in 1928 (Knox 1928).

Rules of the plot:

1. All clues must be plainly stated and described and they should be accessible early on in the game.
2. No wilful tricks or deceptions must be placed on the player other than those played legitimately by the criminal.
3. There should always be fair play and no purposely hidden items or clues that will taka a long systematic search to find. No spoilsport.
4. No new technology or invention that needs a thorough introduction and no advanced physics or psychology.
5. The player may never turn out to be the culprit as it would be false pretences
6. The culprit must be determinable by logical deductions.
7. There absolutely must be a corpse to make the game worth the player's trouble
8. The corpse must be the intended victim. No "oops accidentally murdered the wrong human".
9. The problem of the crime must be solvable by strictly naturalistic means. There should never be any divine intervention, sixth sense, more than 3 dimensions or a trip through the spirit world and absolutely no extra terrestrial beings, green or otherwise, are allowed.
10. The player is the only detective; there will be no other in the game that will solve the problem for him. The player may get assistance, a sort of Watson if he prefers but the Watson will never solve the problem or make any deduction for the player.
11. The murderer must be readily accessible at an early time in the game and must be one or more of the following:
    - have a clear connection to the victim
    - be a prominent character
    - be an interesting character
12. The murderer may not be a servant, have a prior known criminal record, be an immigrant or in any other way be a typical culprit for a prejudicial mind. The murderer should be someone who would normally not be a likely suspect of a criminal investigation.
13. There must be only one culprit, no matter how many murders are committed. The culprit may have an assistant or a black mailer but the actual dead is done by a single human.
14. No secret society or organized crime syndicate. It ruins the fun of the deduction game and the culprit's self respect.
15. The crime should never turn out to be an accident or a suicide. It is a crime plain and simple; the player needs to be able to trust this.
16. The motive should be personal to the murderer and it should be plausible. No insanity or bad social background that makes the motive something less than rational personal hate. This means that a bad hair day is not a just cause for murder but simple greed is.
17. There must be one and only one solution to the game. There must be an absolute impossibility to deduct correctly a wrong culprit, murder weapon or motive. This means that if the player puts forth an incorrect solution the game engine should be able to point out a logical fault in the player's deduction.
18. Finally the murderer must be a human being of average or greater intelligence

4.3. Knowledge base

The game plot is read from the Bayesian network into a knowledge base that the game uses to construct an in-game logic. The game exists in a closed world that is defined by a set of general knowledge and inference knowledge. There are five abstract types in the world: victim, scene, suspect, weapon and object. Each has a name, set of attributes and a description of itself.

The general knowledge describes the attributes of the abstract types and states their logical significance. This is done by assigning values to the attributes. As the following example shows:

- v_hair_on_victim = True $ There is alien hair on the victim
- ms1_shoe_size = Nr_37 $ shoe size nr 37

The prefix "v_" indicates that this node is part of the victim's sub-node and the "ms1_" prefix indicates the first murder suspect. If the directed graph resulting from the Bayesian net has node "v_hair_on_victim" set to true then the game will include the description in its knowledge base and write it out when appropriate

The inference knowledge is the set if inference rules that defines the closed world of the game; it is in a direct one to one relation to the actual Bayesian net it describes every inference possible from the net. Here is an example of such inference knowledge:

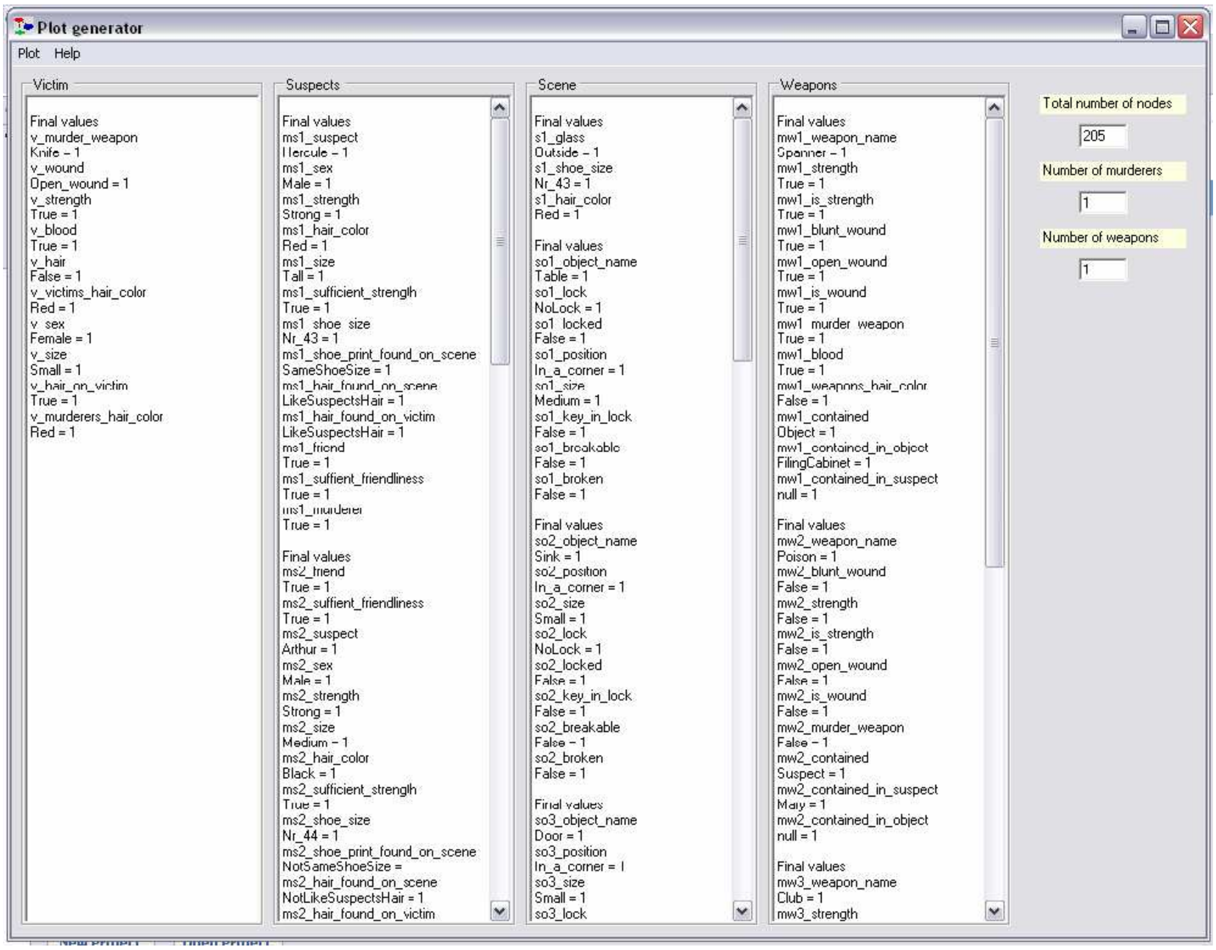
- ms1_shoe_print_found_on_scene = NoPrint$<*suspects name*>'s shoe print was not found on scene
- ms1_hair_found_on_scene = LikeSuspectsHair$<*suspects name*>'s hair colour matches the one found on scene

Notice again the prefix "ms1_" to indicate that these nodes belong to the sub-net describing murder suspect nr 1. The game can read from the Bayesian net what value each node is set to and if it matches one of these inference rules then that rule is given to the player as a possible conclusion.

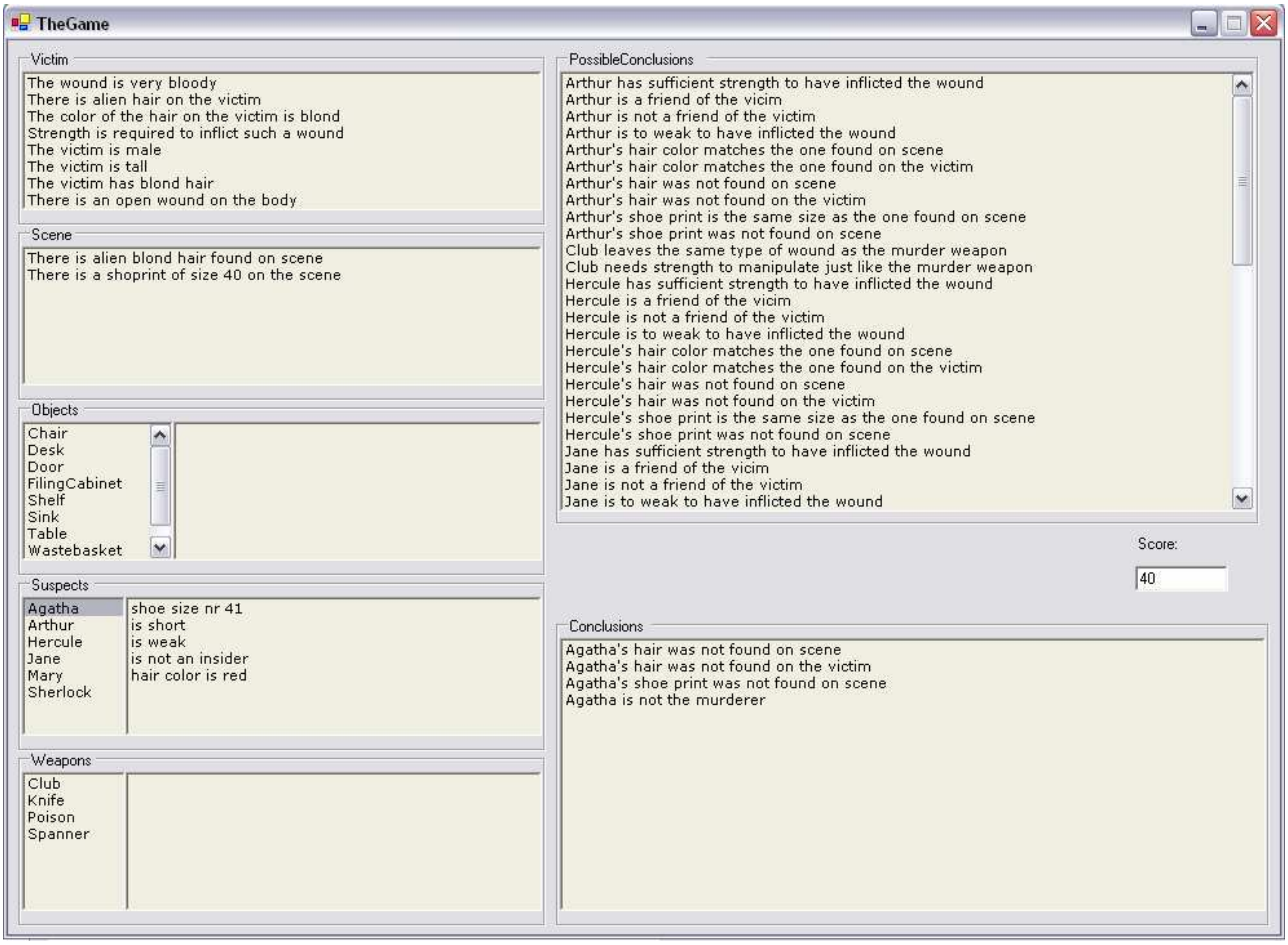
Both the general knowledge and the inferred knowledge are listed in full in Appendix B

5. Results

The result is a dynamic plot generating engine. This engine creates a logically coherent plot with a unique murderer and a unique murder weapon. It follows those of the 18 rules put forth that are not about motive or opportunity. Each new plot can be seen in detail in a window that shows each node of the Bayesian net and the value it is set to.



From each plot generated the game engine reads the initial facts to put forth a new unique murder mystery that is solvable with simple inference, with only one possible murderer and one possible murder weapon. Here we see the initial steps of such a game. Agatha is not the murderess but since the shoeprint found on scene is nr 40, and men rarely fit that shoe size, it is clear that a woman is the murderer. On the left side are all the initial facts that the player gets to draw conclusions from. On the right side there are all the possible conclusions, both correct and incorrect conclusions, to choose from. If a correct conclusion is chosen then the player gets plus 10 points and it goes into the conclusion box. If the conclusion chosen is incorrect then he gets -25 points. To win the player needs to find both the killer and the murder weapon. He gains 20 points for each.



6. Discussion

As previously discussed the research question addressed in this report is whether it is possible to use Bayesian nets to automatically generate coherent story plots for a murder mystery game. The results show that this can be done at least to the extent of setting up a murderer, murder weapon, victim and scene. The intention was to include the motive and opportunity also but there was not enough time to do so. This would have involved the use of a Proppian like morphology as described in the related works section. To implement motive into the plot generation it is necessary to get some kind of narration into the causality so that the player would see the plot as plausible, meaning that it may not lack some elementary functionality or behavioural patterns that we expect in humans and human interactions. This functionality should be implemented in a similar way as the implementation of a unique murderer and unique murder weapon. This means that it needs to be a series of simple and completely independent functions, although they may and perhaps should be executed in a specific order. This is necessary to ensure that they can be applied to any size of Bayesian network as long as the network fits a specific set of rules. In the future the net needs to be able to grow with out worrying about the plot coherency too much. Also it is the ultimate goal to have the Bayesian net changeable at execution time so that the constraints themselves can be changed in respect to game play and player interaction.

When implementing the elementary causality and the knowledge base it became clear that to try to write out every logical constraint or every inference rule would be impossible. To explain this we can imagine dropping a glass and immediately freezing time. Now we have all the time in the world and our mission is to determine whether the glass will break when it hits the floor. If we would try to write down all the logical clauses that will contribute in determining this, then we would easily need to write down billions of clauses since all the elements that could affect whether the glass will break are infinitely many. If on the other hand we would use probabilities to determine whether the glass will break, then we only need the most significant contributors to get a very reasonable and exact estimate. This is exactly the strength of using a probabilistic decision net it makes it possible to draw up in simple terms, fairly complicated scenarios of multiple causalities. For example instead of counting every logical clause that is influential in plot construction, the net only draws out the connections between essential contributors and calculates the respective probabilities. This also reduces the knowledge base into a simple set of descriptive sentences for each node.

7. Conclusion

As seen by the 18 rules the basic structure holds, sometimes by triviality as other options than those described by the rules are simply not included in the knowledge base. It is clear that the experiment is successful, the game engine manages to create a logically consistent plot and it appears to be coherent as it meets the 18 constraints put forth. Further testing needs to be done to see whether it fulfils human expectations of a murder mystery plot or not. For such an experiment to be successful it will be necessary to create a more appealing graphical user interface so that the experiment does not fail due to lack of user friendliness.

The overall results are very exciting to my mind. It is clear that it is possible to dynamically generate new plots. The next steps should be to implement motive with the Proppian functionality as discussed and to have the constraints influencing the plot generating dynamically changeable so that they can be manipulated over the course of a game. Although this experiment was successful, it is so somewhat by triviality both because of a very selective knowledge base and also because of the lack of time in the game. There is of course no linearity where there is no passing of time. The next logical steps are thus to implement time into the game and to create a more diverse game knowledge base than was used in this experiment. My work has shown that it is possible to dynamically generate plots and thus the first step in solving the problem of linearity and lack of replayability in computer games has been tackled and solved.

8. Bibliography

Aarne, A. (1911). Verzeichnis der Märchentypen. *Folklore Fellows Communications*. No. 3, Helsinki

Bangsø, O., Jensen, O. G., Jensen, F. V., Andersen, P. B. and Kocka, T. (2004). Non-Linear Interactive Storytelling Using Object-Oriented Bayesian Networks. *Proceedings of the International Conference on Computer Games: Artificial Intelligence, Design and Education (CGAIDE 2004)*, University of Wolverhampton.

Doyle, A. C. (1960). Silver Blaze. *The Complete Sherlock Holmes*, London, Doubleday, pages 335-350

Doyle, A. C. (1960). The Adventure of the Blanched Soldier. *The Complete Sherlock Holmes*, London, Doubleday, pages 1000-1012

Fairclough, C. R. and Cunningham P. (2004). AI Structuralist Storytelling In Computer Games. *Proceedings of the International Conference on Computer Games: Artificial Intelligence, Design and Education (CGAIDE 2004),* University of Wolverhampton.

Freeman, R. A. (1924). The Art of the Detective Story, *The Art of the Mystery Story (1992)*, New York, Carroll & Graf Publishers, pages 7-18.

Jensen, F. V. (2001). *Bayesian Networks and Decision Graphs*, New York. Springer Verlag.

Knox, R. A. (1928). Detective Story Decalogue, *The Art of the Mystery Story (1992)*, New York, Carroll & Graf Publishers, pages 194-197.

Lindley, C. A. and Eladhari, M. (2002). Causal Normalisation: A Methodology for Coherent Story Logic Design in Computer Role-Playing Games. *Proceedings of the Computers and Games (CG 2002), Third International Conference*, Edmonton, Canada, pages 292-307.

Propp, V. A. (1968). *Morphology of the Folktale (2nd edition)*. Texas, University of Texas Press.

Riedl, M. O. and Young, R. M. (2003). Character-Focused Narrative Generation for Execution in Virtual Worlds. *Proceedings of the International Conference on Virtual Storytelling 2003*, Toulouse, France, pages 47-56.

Schneider, O., Braun, N. and Habinger, G. (2003) Storylining Suspense: An Authoring Environment for Structuring Non-Linear Interactive Narratives. *The 11-th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG 2003)*, Plzen, Czech Republic.

Berkley, Anthony. (1928). The Detection Club Oath. *The Art of the Mystery Story (1992)*, New York, Carroll & Graf Publishers, pages 197-203.

Theune, M., Faas, S., Nijholt A. and Heylen. D. (2002). The Virtual Storyteller: Story creation by intelligent agents. *Workshop on Storytelling in Collaborative Virtual environments at ACM Collaborative Virtual Environments (CVE 2002),* Bonn, Germany.

Theune, M., Rensen, S., op den Akker, R., Heylen, D. and Nijholt. A. (2004). Emotional Characters for Automatic Plot Creation, *Proceedings of the Technologies for Interactive Digital Storytelling and Entertainment (TIDSE 2004) Conference,* Darmstadt, Germany, pages 95-100.

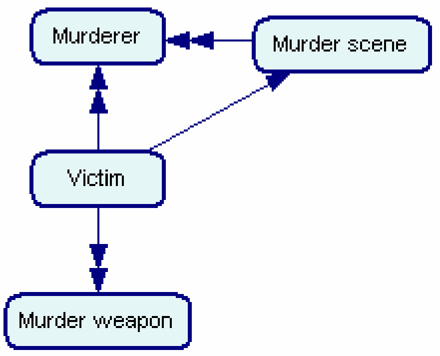
Van Dine, S. S. (1928). Twenty Rules for Writing Detective Stories, *The Art of the Mystery Story (1992)*, New York, Carroll & Graf Publishers, pages 189-194.

Verbrugge. C. (2002). A Structure For Modern Computer Narratives, *Proceedings of the Computers and Games(CG 2002), Third International Conference*, Edmonton, Canada, pages 308-325.

Appendix A

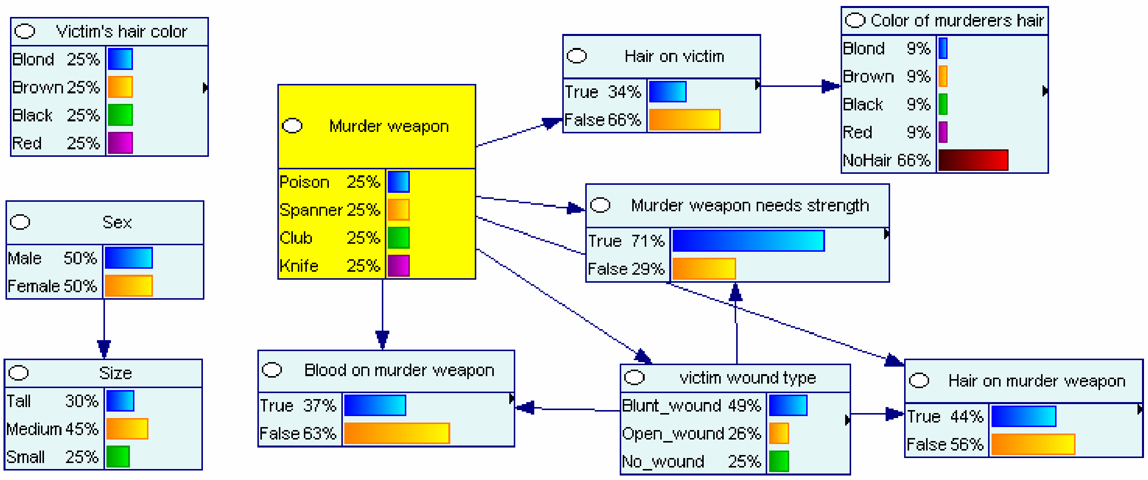
Overview

The net is organized into subnets that have some causal connections between each others.



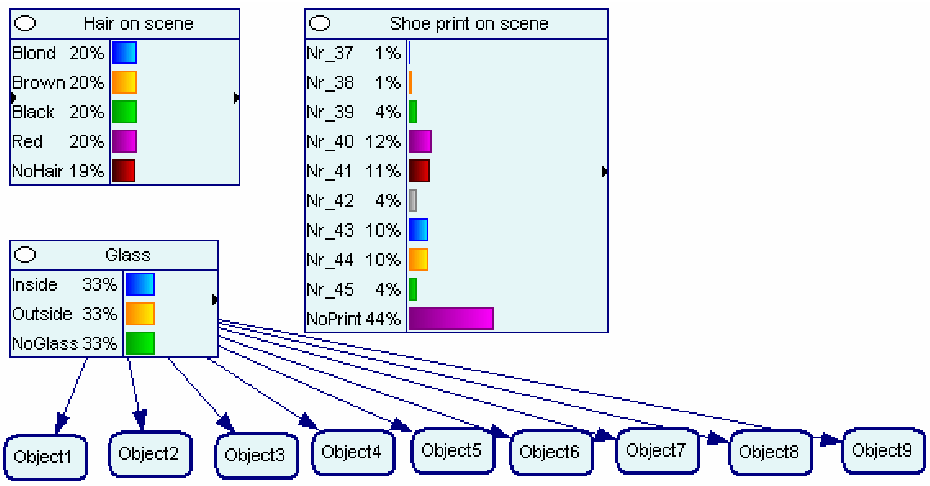
Victim

Details all that can be found or known about the victim.



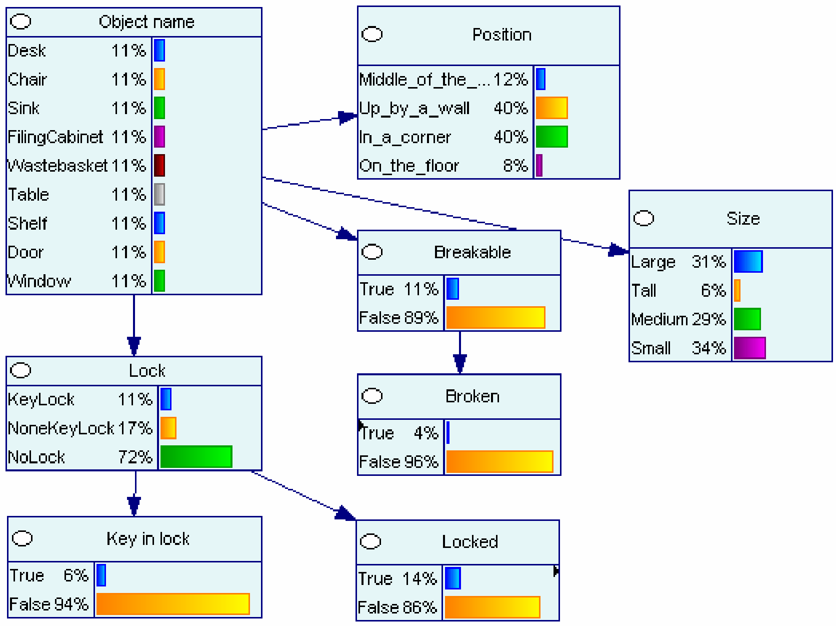
Scene

Details all that can be detected about the scene and all the objects that can be found there.



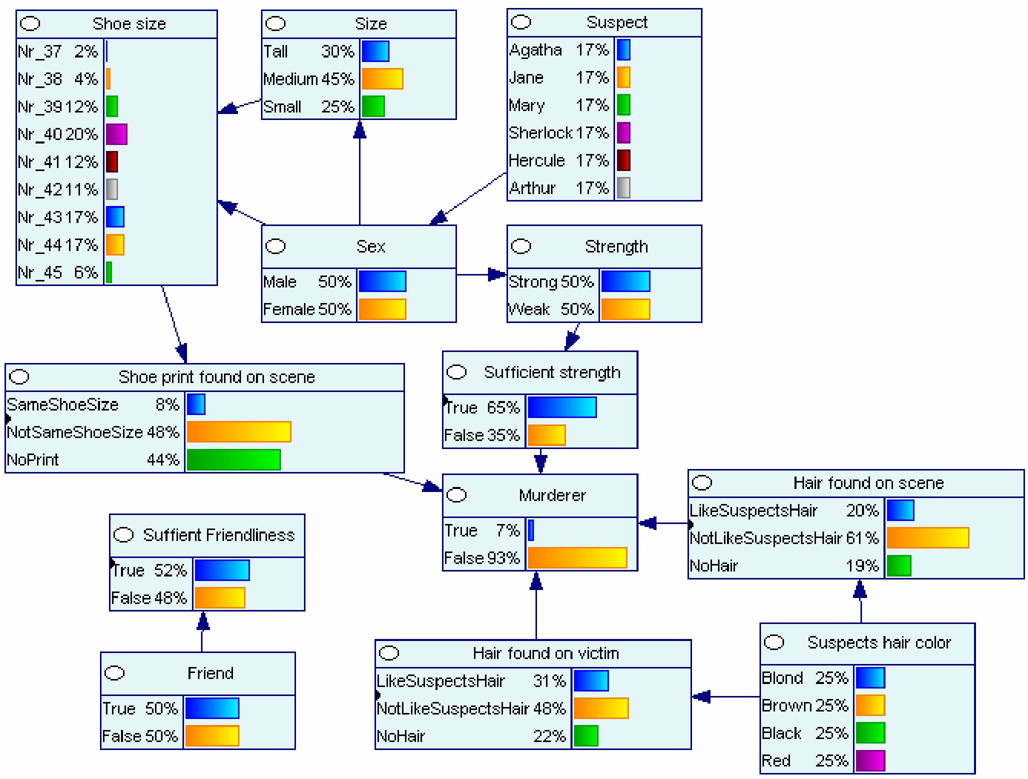
Object

There can be any number of objects; each object is in a subnet that details all there is to know about that object.



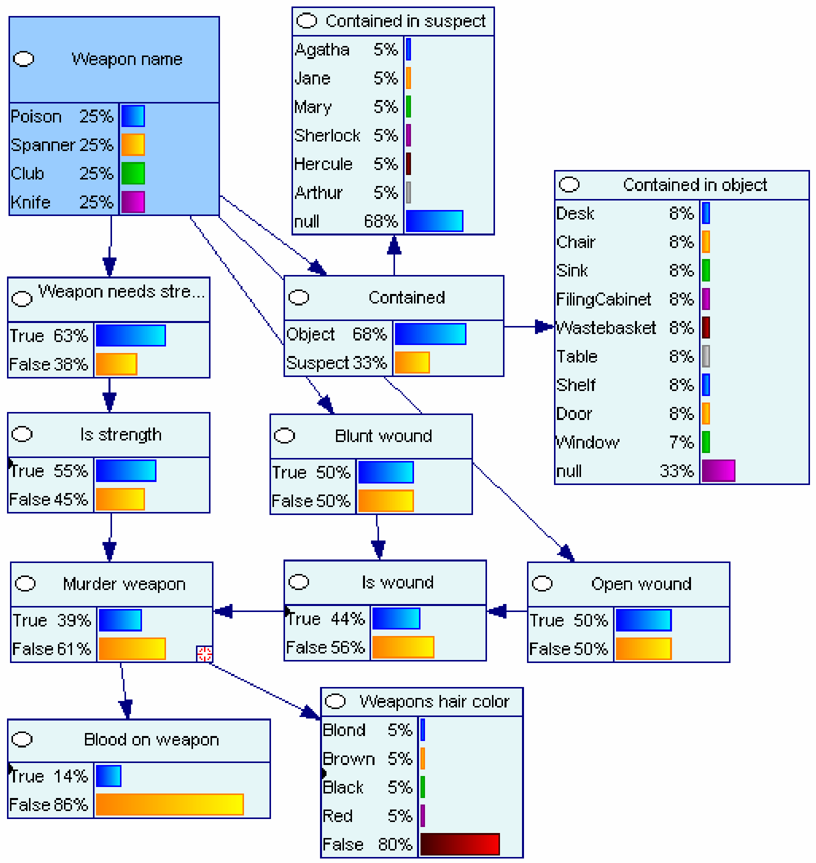
Suspect

There can be any number of suspects; each suspect is in a subnet that details all there is to know about that suspect.



Weapon

There can be any number of weapons; each weapon is in a subnet that details all there is to know about that weapon.



Appendix B

To translate the Bayesian net to a player the game uses a knowledge base, one for general knowledge and one for inferred knowledge this knowledge base is static, that is it stays the same for every plot generated and is read in from text files.

General knowledge

v_blood=True$The wound is very bloody
v_hair_on_victim=True$There is alien hair on the victim
v_murderers_hair_color=Blond$The color of the hair on the victim is blond
v_murderers_hair_color=Brown$The color of the hair on the victim is brown
v_murderers_hair_color=Black$The color of the hair on the victim is black
v_murderers_hair_color=Red$The color of the hair on the victim is red
v_hair=True$This is a head wound
v_strength=True$Strength is required to inflict such a wound
v_sex=Male$The victim is male
v_sex=Female$The victim is female
v_size=Tall$The victim is tall
v_size=Medium$The victim is of average height
v_size=Small$The victim is short
v_victims_hair_color=Blond$The victim has blond hair
v_victims_hair_color=Brown$The victim has brown hair
v_victims_hair_color=Black$The victim has black hair
v_victims_hair_color=Red$The victim has red hair
v_wound=Blunt_wound$The wound on the body has been made by a blunt weapon
v_wound=Open_wound$There is an open wound on the body
v_wound=No_wound$There is no wound on the body
ms1_shoe_size=Nr_37$shoe size nr 37
ms1_shoe_size=Nr_38$shoe size nr 38
ms1_shoe_size=Nr_39$shoe size nr 39
ms1_shoe_size=Nr_40$shoe size nr 40
ms1_shoe_size=Nr_41$shoe size nr 41
ms1_shoe_size=Nr_42$shoe size nr 42
ms1_shoe_size=Nr_43$shoe size nr 43
ms1_shoe_size=Nr_44$shoe size nr 44
ms1_shoe_size=Nr_45$shoe size nr 45
ms1_size=Tall$is tall
ms1_size=Medium$is of average hight
ms1_size=Small$is short
ms1_strength=Strong$is strong
ms1_strength=Weak$is weak
ms1_friend=True$is an insider
ms1_friend=False$is not an insider
ms1_hair_color=Blond$hair color is blond
ms1_hair_color=Brown$hair color is brown
ms1_hair_color=Black$hair color is black
ms1_hair_color=Red$hair color is red
ms2_shoe_size=Nr_37$shoe size nr 37
ms2_shoe_size=Nr_38$shoe size nr 38
ms2_shoe_size=Nr_39$shoe size nr 39
ms2_shoe_size=Nr_40$shoe size nr 40
ms2_shoe_size=Nr_41$shoe size nr 41
ms2_shoe_size=Nr_42$shoe size nr 42
ms2_shoe_size=Nr_43$shoe size nr 43
ms2_shoe_size=Nr_44$shoe size nr 44
ms2_shoe_size=Nr_45$shoe size nr 45
ms2_size=Tall$is tall
ms2_size=Medium$is of average hight
ms2_size=Small$is short
ms2_strength=Strong$is strong
ms2_strength=Weak$is weak
ms2_friend=True$is an insider
ms2_friend=False$is not an insider
ms2_hair_color=Blond$hair color is blond
ms2_hair_color=Brown$hair color is brown
ms2_hair_color=Black$hair color is black

ms2_hair_color=Red$hair color is red
mw1_blunt_wound=True$This is a blunt weapon
mw1_open_wound=True$This weapon leaves an open wound
mw1_blood=True$There is blood on the weapon
mw1_weapons_hair_color=Blond$There is blond hair on the weapon
mw1_weapons_hair_color=Brown$There is brown hair on the weapon
mw1_weapons_hair_color=Black$There is black hair on the weapon
mw1_weapons_hair_color=Red$There is red hair on the weapon
mw1_strength=True$The weapon needs strength to manipulate
mw2_blunt_wound=True$This is a blunt weapon
mw2_open_wound=True$This weapon leaves an open wound
mw2_strength=True$The weapon needs strength to manipulate
s1_hair_color=Blond$There is alien blond hair found on scene
s1_hair_color=Brown$There is alien brown hair found on scene
s1_hair_color=Black$There is alien black hair found on scene
s1_hair_color=Red$There is alien red hair found on scene
s1_glass=Inside$There is glass lying on the inside of the window
s1_glass=Outside$There is glass lying on the outside of the window
s1_shoe_size=Nr_37$There is a shoprint of size 37 on the scene
s1_shoe_size=Nr_38$There is a shoprint of size 38 on the scene
s1_shoe_size=Nr_39$There is a shoprint of size 39 on the scene
s1_shoe_size=Nr_40$There is a shoprint of size 40 on the scene
s1_shoe_size=Nr_41$There is a shoprint of size 41 on the scene
s1_shoe_size=Nr_42$There is a shoprint of size 42 on the scene
s1_shoe_size=Nr_43$There is a shoprint of size 43 on the scene
s1_shoe_size=Nr_44$There is a shoprint of size 44 on the scene
s1_shoe_size=Nr_45$There is a shoprint of size 45 on the scene
so1_position=Middle_of_the_room$it is in the middle of the room
so1_position=Up_by_a_wall$it is up by a wall
so1_position=In_a_corner$it is in a corner
so1_position=On_the_floor$it is on the floor
so1_size=Large$it is large
so1_size=Tall$it is tall
so1_size=Medium$it is medium
so1_size=Small$it is small
so1_broken=True$it is broken
so1_lock=KeyLock$There is a lock on it
so1_lock=NoneKeyLock$There is a latch on it
so1_locked=True$it is locked
so1_key_in_lock=True$the key is in the lock

Inferrance knowledge

ms1_sufficient_strength=True$ has sufficient strength to have inflicted the wound
ms1_sufficient_strength=False$ is to weak to have inflicted the wound
ms1_shoe_print_found_on_scene=SameShoeSize$'s shoe print is the same size as the one found on scene
ms1_shoe_print_found_on_scene=NotSameShoeSize$'s shoe print was not found on scene
ms1_shoe_print_found_on_scene=NoPrint$'s shoe print was not found on scene
ms1_hair_found_on_scene=LikeSuspectsHair$'s hair color matches the one found on scene
ms1_hair_found_on_scene=NotLikeSuspectsHair$'s hair was not found on scene
ms1_hair_found_on_scene=NoHair$'s hair was not found on scene
ms1_hair_found_on_victim=LikeSuspectsHair$'s hair color matches the one found on the victim
ms1_hair_found_on_victim=NotLikeSuspectsHair$'s hair was not found on the victim
ms1_hair_found_on_victim=NoHair$'s hair was not found on the victim
ms1_suffient_friendliness=True$ is a friend of the vicim
ms1_suffient_friendliness=False$ is not a friend of the victim
ms1_murderer=True$ is the murderer
ms1_murderer=False$ is not the murderer
ms2_sufficient_strength=True$ has sufficient strength to have inflicted the wound
ms2_sufficient_strength=False$ is to weak to have inflicted the wound
ms2_shoe_print_found_on_scene=SameShoeSize$'s shoe print is the same size as the one found on scene
ms2_shoe_print_found_on_scene=NotSameShoeSize$'s shoe print was not found on scene
ms2_shoe_print_found_on_scene=NoPrint$'s shoe print was not found on scene
ms2_hair_found_on_scene=LikeSuspectsHair$'s hair color matches the one found on scene
ms2_hair_found_on_scene=NotLikeSuspectsHair$'s hair was not found on scene
ms2_hair_found_on_scene=NoHair$'s hair was not found on scene
ms2_hair_found_on_victim=LikeSuspectsHair$'s hair color matches the one found on the victim
ms2_hair_found_on_victim=NotLikeSuspectsHair$'s hair was not found on the victim
ms2_hair_found_on_victim=NoHair$'s hair was not found on the victim

ms2_suffient_friendliness=True$ is a friend of the vicim
ms2_suffient_friendliness=False$ is not a friend of the victim
ms2_murderer=True$ is the murderer
ms2_murderer=False$ is not the murderer
mw1_murder_weapon=True$ is the murder weapon
mw1_murder_weapon=False$ is not the murder weapon
mw1_is_strength=True$ needs strength to manipulate just like the murder weapon
mw1_is_strength=True$ does not need the same strength to manipulate as the murder weapon would
mw1_is_wound=True$ leaves the same type of wound as the murder weapon
mw1_is_wound=True$ does not leave the same type of wound as the murder weapon
mw2_murder_weapon=True$ is the murder weapon
mw2_murder_weapon=False$ is not the murder weapon
mw2_is_strength=True$ needs strength to manipulate just like the murder weapon
mw2_is_strength=True$ does not need the same strength to manipulate as the murder weapon would
mw2_is_wound=True$ leaves the same type of wound as the murder weapon
mw2_is_wound=True$ does not leave the same type of wound as the murder weapon