

War of the Language Worms

A logic puzzle
Age group: 7-adult

From:

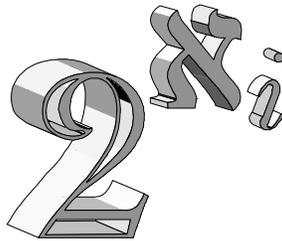
Jim Andrews
1806 8th Ave, #313
Seattle WA, 98101
USA

Jim@vispo.com
(206) 223-7377

Vispo ~ Langu(im)age:
<http://www.vispo.com>

Dr. Michael Fellows
Dept. of Computer Science
P.O. Box 3025
University of Victoria
Victoria, B.C.
Canada, V8W 3P2

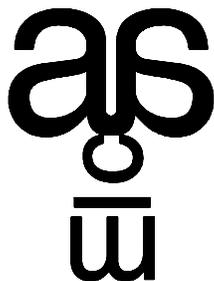
mfellows@csr.uvic.ca



WAR OF THE LANGUAGE WORMS

TABLE OF CONTENTS

THE STORY AND RULES 3
The Story 3
The Rules 3
 Graph Logic Rule 1 3
 Graph Logic Rule 2 3
 Graph Logic Rule 3 4
 Language Worm Weapons and Other Game Rules 4
 PLAYER CONTROLS AND LANGUAGE WORMS 5
Mouse and Keyboard 5
Language Worms 5
Conventional Weapons 6
 FUNDAMENTAL INFERENCES AND LOGIC CHECKING 7
 SCORING SYSTEM 9
General Principles 9
Awarding Valid Inferences 9
Tuning the Assessment System 10
Rewarding Quick Play 12
 MULTI-PLAY AT HOME, IN THE CLASSROOM, AND ON THE INTERNET 13
The Puzzle/Game Exists Naturally at Home, in the Classroom, over the Modem, and on the Internet ... 13
Play Solo or Play the Computer or Play a Friend Simultaneously or Sequentially 13
On the Internet 14
 SAMPLE WORDSCAPES 15
 SAMPLE PLAY-THROUGH WAR OF THE LANGUAGE WORMS 17
 GENERATING AN INSTANCE (I) 18
Step 1: Determine the Number of Satellites and label each satellite with a group name. Also, pick a satellite to be the Widgetian base. 18
Step 2: Connect the Widgetian headquarters to satellites of one group 18
Step 3: Form "Legal" links with probability p 19
Step 4: If any satellites (except the Widgetian headquarters) are not connected to each of the other groups, create links. 19
The Finished Graph 19
 GENERATING AN INSTANCE (II) 20
 THE LOOK OF THE GAME AND PROGRAMMING 21
 FOOTNOTES 22



Who's Home at *Widgetian headquarters*?

THE STORY AND RULES

THE STORY

War of the Language Worms is a game of logic and of Information War in which you attempt to score Transmission power (or media) points by freeing network communications channels controlled by the Widgetians (W). The Widgetians have commandeered communications channels between the Edu (E), Mil (M), and Gov (G) satellites. Your job is to locate the Widgetian satellite and disable it, thereby freeing the Edus, Mils, and Govs from Widgetian tyranny/reruns.

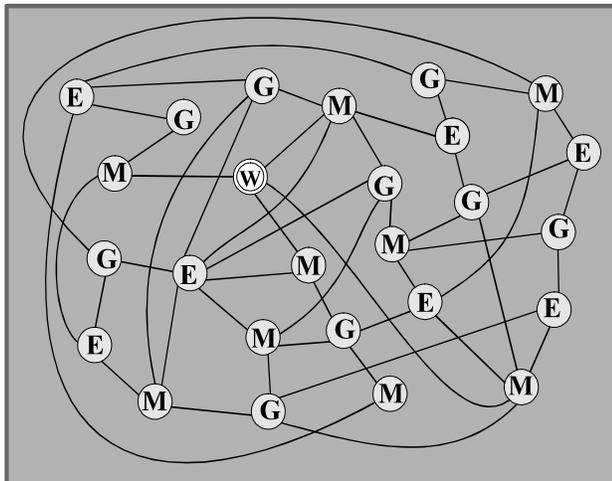


Figure 1: An instance of the game

The story and its language play cartoonish fantasy against strange reality. The “language worms” you wield as weapons to locate the Widgetians appear as cartoonish, comically designed figures. Yet “language worms” or “spiders” or “web crawlers” travel the Internet constantly, automatically gathering information and reporting back to their owners. The situation of the game is not far removed from types of conflict that could exist on the Internet. The player’s goal is to free the Edu, Mil, and Gov satellites whose bandwidths have been filled with Widgetian signals. “Divide and conquer” has always involved dominating or jamming communications.

THE RULES

Widgetian communications propagate between the headquarters satellite and lesser satellites along net links. The Widgetians have captured the communications system *between* groups.

Graph Logic Rule 1

Each satellite (not including Widgetian headquarters) is connected to at least one satellite of each of the other two groups.

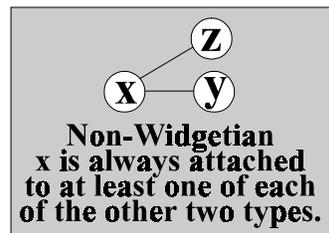


Figure 2: Graph logic rule 1

Graph Logic Rule 2

No satellites are connected to other satellites of the same type.¹

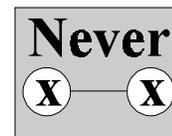


Figure 3: Graph Logic Rule 2

Graph Logic Rule 3

You don't know which satellite is actually the Widgetian base. But unlike every other satellite, it's only connected to satellites of one group.

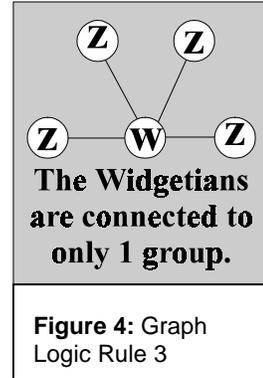


Figure 4: Graph Logic Rule 3

Language Worm Weapons and Other Game Rules

Your arsenal of language worms is fourfold. You may, for instance, trigger the Triangular language worm, i.e. open a channel for it, open the bay doors—loose it on a satellite. You also have the Fork and the Foursome weapons. And, of course, you have the dreaded Anti-Widgetian language worm to polish them off when you locate them. You must pick among worms carefully when zapping a satellite or the Widgetian headquarters. Choose the right worms for the right targets. If you do, you can visit more than one satellite per worm and score extra transmission power points. When your language worm successfully visits a satellite, it dominates some of the info channels into the satellite and uncloaks it, i.e. retrieves the type of the satellite (Edu, Mil, or Gov).

All satellites are cloaked at the beginning (unlike the above diagrams).

All the Widgetian's influence is exercised first through certain key satellites all of the same group. But which group? That changes from game to game. If you sting it with the Fork or Triangle or Foursome then it behaves as any other satellite does when you infiltrate it improperly: it shoots at you. If you zap it with the Cloak Snapper then you're toast. But if you worm it with the Anti-Widgetian language worm then you've done well, the game is over, you get a bonus, and the player with the most points wins. Don't miss with that worm: you're soup if you do. The game ends, you get penalised, and the player with the most points wins.

This game does not have to involve homicide. It is, instead, an info war. You score transmission power points according to how many cloaked satellites you hit per shot. Aim the weapons carefully. If your language worm cannot travel through the channels it must (according to its pattern) to uncloak at least one satellite and free channels, then the attacked satellite detects the intrusion and shoots at you. You lose 2 points of transmission power for each hit. It's beginning to take up your bandwidth. To get it off your bandwidth and begin controlling its bandwidth, you can zap it with the Cloak Snapper. The Cloak Snapper uncloaks the satellite it snaps and stops the satellite from firing. You can use this twice without penalty. Or you can zap an attacker with the Satellite Weapon Deflector, which stops it firing.

The Widgetians have one big weakness: they're almost *defenceless* against sound reasoning. It is the clarity of your thought that defeats the Widgetian conspiracy. Your language worms only work when *you can be sure* that the satellite you're zapping isn't the Widgetian headquarters. The worms search for their assigned patterns (Fork, Triangle, Foursome) among the zapped satellites and their neighbours. The Fork, Triangle, and Foursome uncloak only satellites that *cannot* be the headquarters, that are safe to visit—given current data and the patterns worms search.

This way, the Widgetians are not alerted of your activity. Don't alert them to your presence by worming headquarters (except with the Anti-Widgetian language worm). You'll be in for a worse battle then, at some point—or you'll be soup if you use the Cloak Snapper on headquarters.

A satellite is capable of firing at you unless you free all of its channels from Widgetian control. But they don't fire at you very much unless you make mistakes using the language worms or conventional weapons. They also start shooting at you if you take too long. But if you learn to use the buttons, you'll complete the game and receive less zapping from the satellites and the Widgetians.

In solo play, you win if you locate the Widgetian headquarters and zap it with the Anti-Widgetian language worm. In group play or play against the computer, the player with the most points wins. You start out with 20 transmission power points. If you ever reach 0, then your communications channels have been dominated by Widgetian signals and you are rendered a satellite of the Widgetians.

PLAYER CONTROLS AND LANGUAGE WORMS

For more information concerning the controls, see *Fundamental Inferences and Logic Checking*, which explains why certain patterns are safe to reveal, and *Scoring System*, which explains the point scoring system for revealing various patterns.

MOUSE AND KEYBOARD

MOUSIE: The mouse is used to click on satellites, Widgetian Headquarters, and the other buttons appearing below. Speedplay options might include having the right mouse button cycle among the available buttons.

KEYBOARD PLAY: The above buttons describe a mouse driven interface. The program should be written to accommodate mouse *or* keyboard control. Keyboard control is necessary if two players are to play against one another on a PC. Instead of a mouse, you have a 9 key directional pad to navigate the town. The middle key among the 9 is the "enter key". You use other keys instead of the mouse to control the above buttons. To accommodate coherent keyboard navigation, a limit of 8 links per satellite must be imposed. This limit does not hinder the entertainment value: it's probably undesirable to have more than 6 links connected to a satellite not only because of aesthetic considerations (too cluttered) but also because the more links there are, the easier the game is.

LANGUAGE WORMS



THE FORK: You hit this button before clicking on the cloaked middle satellite of a fork *where the two prongs are uncloaked and of two different groups*. You uncloak the satellite and score 2 points. Improper use of the worm results in the satellite detecting your intrusion. It begins shooting at you. To get it off your back you can either use a worm *successfully* anywhere (which stops all shooting), use the *Cloak Snapper*

on the firing satellite to both uncloak it and stop it firing, or use the *Satellite Signal Deflector* which also stops further shooting from the clicked satellite.



THE TRIANGLE: You hit this switch before clicking on any of the cloaked satellites of a triangle. All the cloaked satellites in the triangle are then uncloaked.

The Triangle is used improperly if the clicked satellite is not part of any triangle. If you use it improperly then the clicked satellite commences firing (firing speed should be adjustable in a **prototype**. Firing speed can also be a ramping factor). Attack is parried by methods described in improper use of the Fork (above).



THE FOURSOME: You hit this before clicking on any cloaked satellites among the middle two. The two end satellites *must* be of the same group and must both be uncloaked. Improper use of the worm involves no such pattern being part of the satellite's communication links. The satellite then fires (as above) and is parried (as above).



THE ANTI-WIDGETIAN LANGUAGE WORM: You hit this before clicking on the cloaked Widgetian headquarters. One way to be sure you've got the right target for this worm is to be sure that all the satellites linked to the assumed headquarters are all of one type. It can only be used once. If you guess headquarters incorrectly while using this worm you're stardust.

If you worm headquarters with the Fork, Triangle, or Foursome, then headquarters simply shoots at you—it behaves as though you've made a bad move (you have).

CONVENTIONAL WEAPONS.



CLOAK SNAPPER: Use this button to zap satellites when you don't know what you're doing or when you've got no safe moves. You sometimes find yourself in a situation where there are no satellites to safely zap (they all *could* be the Widgetians). When using language worms, satellites that cannot *safely* be uncloaked, given current data, refuse to be uncloaked *at all*². So the Cloak Snapper allows you to uncloak a satellite that cannot safely be uncloaked. But don't use it more than twice or you'll be penalised 2 points each additional time. And don't zap Widgetian headquarters with it or you lose and the game's over. You become a communications satellite of the Widgetians. The Cloak Snapper also stops the clicked satellite from firing at you (if it's firing).



SATELLITE SIGNAL DEFLECTOR: If a satellite shoots at you, this weapon will stop it firing for a while. You can fire the signal deflector whenever you want.

You can permanently stop satellites from firing at you if you free all the info channels connected to the satellite. Info channels are freed when language worms traverse them successfully (find their pattern among adjacent info channels).

FUNDAMENTAL INFERENCE AND LOGIC CHECKING³

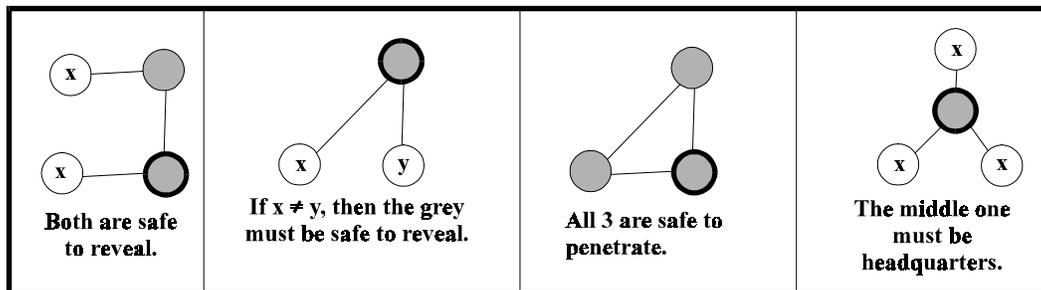


Figure 5: The grey satellites are cloaked. The white satellites are uncloaked. The heavy outline indicates the satellite clicked by the player.

Let's agree to limit the domain of "valid inferences" to a smaller domain than may really be the case (in the multiple-moves-from-here sphere). Let's say a "valid inference" consists of detecting one of the Fundamental Inferences diagrammed above. We can determine more or less instantaneously whether a particular move represents a valid inference (in the limited sense) that the satellite is safe to uncloak. This follows from the fact that when a move is made, only a few calculations need to be done by the computer to determine whether a move is a valid inference (again, in the limited sense). The number of possible situations the computer has to check for is not prohibitive.

A Fundamental Inference Pattern is a connected configuration of satellites (some cloaked—but perhaps some are uncloaked) such that it's logically impossible for *any* of the cloaked satellites to be the Widgetian base. A F.I.P is *not* a concatenation of other F.I.Ps.

I haven't been able to find any more Fundamental Inference Patterns. Are there any more? Mike says that there *must* be. There are certainly no more with four satellites. And probably none with 5 satellites. Beyond that, I'm not sure. But even if there are some with 6 or more satellites, the search time for such patterns would probably disturb real-time play. And only superb players would find them. By the time a player becomes superb, she can play at the highest level of play where all the logic-checker training wheels come off (more on that later).

The first three above are situations where you may validly infer it's safe to uncloak the satellite or satellites. The fourth is a situation where you may validly infer the location of the Widgetian headquarters.



The first situation above implies that both middle satellites are safe to reveal (even if they're both cloaked). Why? An *x* cannot be connected to an *x*. Therefore neither of the middle satellites are of type *x*. So they are each of type *y* or *z*. But the middle satellites are connected, so $y \neq z$. Hence each of the middle satellites is connected to two different groups (*x* and (*y* or *z*)). So they are both safe to reveal.

Searching for the Foursome pattern among the adjacent satellites is the most time consuming task the program does. But if we limit the number of channels per satellite to a maximum of

6 then, at worst, the program searches for $(6*5*5)=125$ possible foursomes. This will only cause a small ripple in real-time play. Most often, it will search for far fewer (usually $36=4*3*3$).



The second situation implies that the cloaked middle satellite is safe since it's connected to two different groups and is, therefore, not the Widgetian headquarters.

If we limit the number of channels per satellite to a maximum of 6, then the program will only have to search for 15 possible forks (at worst).



The point of the third diagram is that if 3 satellites form a triangle, they're all safe regardless of whether any of them is cloaked. Why? All three groups are represented (or two satellites of the same group are connected—which never happens). Therefore each satellite is connected to at least two different groups. Hence none of them is the Widgetian headquarters. So they are all safe to reveal.

If we limit the number of channels per satellite to 6 then, at most, the program will only have to search for 15 triangles.



The point of the fourth diagram is that if all the satellites linked to a cloaked satellite are uncloaked and of the same type, then you may validly infer the location of the Widgetian headquarters to be the middle satellite. The satellites adjacent to the headquarters do not need to be uncloaked before you can successfully use the Anti-Widgetian language worm.

Given the limited number of situations the computer has to check to determine whether the player has made a valid inference (even if I've missed one or two), you can see that the logic checking option is feasible for real-time play. I've analysed the running time for such pattern searches and they're not unruly.

In some game-play modes, when you click on a satellite to uncloak it, the satellite will *not* be uncloaked unless a valid inference has been made (a worm has been used correctly). This provides both good pedagogic reinforcement and well-defined, entertaining competitive play. It integrates pedagogy and fun. You don't have to understand logic to play at the lowest levels. You can learn as you go. At first it's a shoot-em-up game involving pattern recognition.

Current educational research strongly suggests it's unrealistic to expect most young children to be able to comprehend a logic game without considerable ramping and drawing attention carefully to the relevant data and concepts. Kids often require considerable tutoring when approaching *Minesweeper*, for instance. Even many grade 11 students.

SCORING SYSTEM

GENERAL PRINCIPLES

The guiding principles of a point system would take into account

- Guesses versus valid inferences (which are awarded far more points).
- Time taken (the shorter the time the better).
- Awarding short, valid, winning routes more than valid and winning but less than brilliant routes.

Whether you are playing solo, against the computer, against a friend, or on the Internet, the game should store lots of information about your standing, scores, and history.

The scoring keys on the number of cloaked satellites you uncloak in one pop, not on the number of valid inferences involved in one click. It also keys on the difficulty of the inference. Some patterns are easier to spot than others.

If you uncloak more than 1 satellite with one worm, then you score 3 points per uncloak satellite. You score 2 points if you uncloak just 1 satellite. The satellite detects the intrusion and commences firing if you uncloak none.

AWARDING VALID INFERENCE

When you hit one of the language worm triggers and then click on a cloaked satellite, the program searches the immediate vicinity for all possible occurrences of the pattern.

The game momentarily changes the colour of the forked channel lines to red when it finds the Fork used successfully. The player sees, then, the pattern that the worm is searching for. Same with successful use of the Triangle and Foursome.



The Fork: You're awarded 2 points for using this switch successfully. You uncloak the middle satellite. Even if multiple forks emanate from a satellite (as in Figure 6) you're awarded only a total of 2 points. All channels involved in a valid fork move are freed from the Widgetians (all 3 in the diagram to the right because there are two valid forks). The satellite fires at you if you muck up with this worm. Same with the Triangle and Foursome.

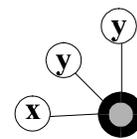


Figure 6



The Triangle: Hit this and then a cloaked satellite that's in a triangle. The worm searches for all triangles connected to the satellite. If there's more than 1 cloaked satellite among the triangles it finds, then you get 3 points per satellite. You get two points if you uncloak just one satellite. If there's no triangles, you get shot at.



The Foursome: If you use this weapon but the two end satellites aren't both uncloak and of the same group, you're under fire. If the pattern is present and the two middle satellites are both cloaked, you score 6 points and uncloak both. If 3 (rather than 2) of the satellites in the pattern are uncloak and you click on the cloaked

satellite, you get 2 points. The pattern is present but scores no more than a fork. You can uncloak intertwined Foursomes at once if you click the right satellite.



The Widgetian headquarters: If you guess the Widgetian headquarters incorrectly while using this button, you lose and the game is over. The players who didn't make the mistake finish with the number of points they had when the error was committed. The player who made the mistake finishes with 10 points less than the player with the least points. The player with the most points wins.

If you use this button successfully, then you score 6 points, the game is over, and the player with the most points wins.



The Cloak Snapper: No points are awarded or docked except if you use this conventional weapon more than twice in a game, in which case there's a 2 point penalty for each use over twice. If you use this weapon on headquarters, you lose, you're an info-bite, buster.



Satellite Signal Deflector: This scores no points and earns no punishment. When the satellites are firing at you, you can stop them taking up your bandwidth by zapping them with this conventional (non logic) weapon. It's possible to make the player earn satellite deflector capabilities, but that isn't part of the design currently.

TUNING THE ASSESSMENT SYSTEM

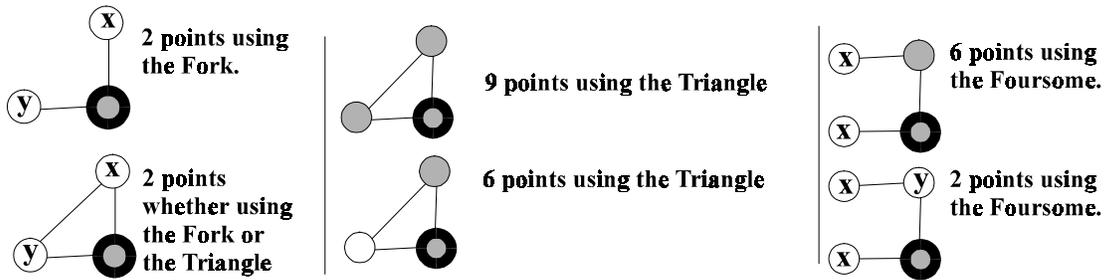
Tuning the assessment system will be important not just to make the game as lively, fair, and wisely rewarding as possible, but also so that the information teachers can get from the game about student performance is useful. Teachers need to be able to *use* the information if the game is to stand a chance of widespread success within the schools.

The game needs to keep track of individual and group performance. Has the student successfully completed several puzzles? What level are they at? Have they gotten past the level where all they use is the Fork? Have they managed to construct any graphs themselves? What level of graph? Where are they among their peers in points, wins, losses, etc.

Of course the \$64,000 question is: DO THEY UNDERSTAND LOGICAL INFERENCE ANY BETTER? It's conceivable that if the pedagogical element of the game wasn't well thought out, players could come away from the game being able to spot Fundamental Inference patterns, and even combinations of them very well and quickly but still not understand, for instance, that the satellites in a triangle **MUST** be safe to reveal. They should also understand **WHY** they're safe to reveal.

The Help and Tutoring (and possibly testing) system of the program needs to be designed with these questions in mind. If the program detects that the player is repeatedly using, for instance, the fork button without success, then the program could switch into tutor mode concerning the logic of the fork. Not too many tutorials would have to be included. One for each button. One or two for the construction of graphs. And one on the story.

Fundamental Situations (scoring patterns)



If your language worm can uncloak at least two satellites, then you're awarded 3 points per satellite. If your worm only uncloaks one, then you're rewarded 2 points. If you mess up, i.e. your language worm cannot propagate along links at all, the satellite detects the intrusion and begins firing on you. Each shot makes you 2 points poorer in transmission power.

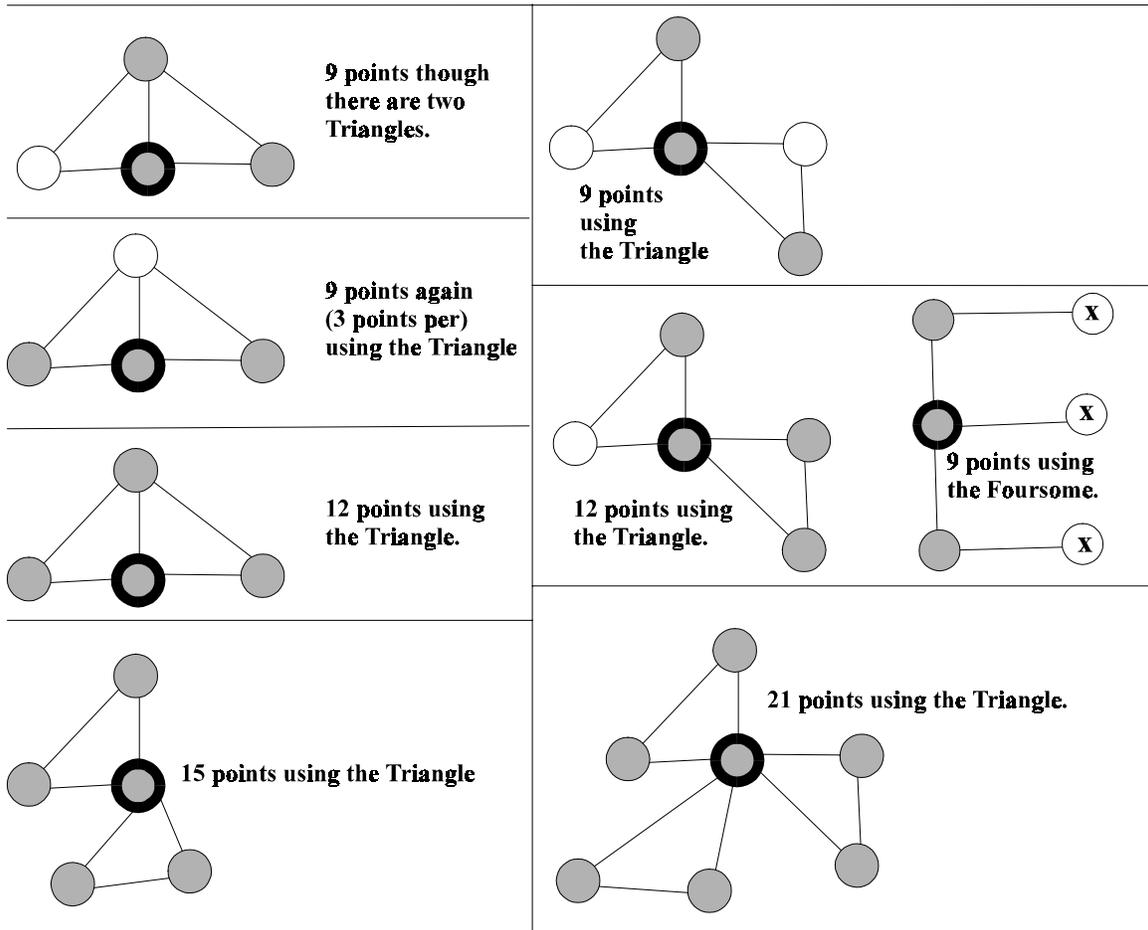


Figure 7: Scoring Fundamental Inference Patterns

REWARDING QUICK PLAY

Time is also a consideration. Bonus points should be awarded for quick play. If we say that you should reveal one satellite every 5 seconds (obviously this ramping parameter should be adjustable in a **prototype**), then *time bonus* = $5 * (\text{total \# of satellites}) - (\text{total seconds played})$. This can be a negative value. That's fine too. This bonus is awarded to all players if someone finishes the game successfully. This bonus doesn't determine a winner at all (since it's added to each player's score) but it does reward fast players in tournament play where total points over many games is a factor.

MULTI-PLAY AT HOME, IN THE CLASSROOM, AND ON THE INTERNET

THE PUZZLE/GAME EXISTS NATURALLY AT HOME, IN THE CLASSROOM, OVER THE MODEM, AND ON THE INTERNET

The design should take into account the possibility of use within the classroom, at home, over the modem, and on the Internet. Scoring and real-time competition (as well as pedagogic acumen) are important factors in these environments. This puzzle presents natural solutions to all four situations in addition to a well-defined scoring system which will correlate well with the level of the students' understanding of logical inference.⁴

PLAY SOLO OR PLAY THE COMPUTER OR PLAY A FRIEND SIMULTANEOUSLY OR SEQUENTIALLY

Whether at home, in the classroom, or on the Internet, you can play alone or against a computer or against someone else.

At home, the player should also be able to play someone else via the keyboard.

A game between players could take any of the following forms:

1. Each player has their own graph (two on the screen at once) and each tries to complete their own graph/score more points before the other. A potential limitation of this mode is the size of the graph each player can use.
2. Each player plays sequentially, takes turns solving graphs.
3. There's only one graph on screen and the two (or 3 or 4) players are competing to reveal the most satellites/score most points.
4. The player could play against the computer. The speed and 'intelligence' of the computer's play is easily adjustable depending on how much use the robot makes of the logic checker and which construction method we use in making the graph. Various parameters can be adjusted to vary the intelligence: the speed with which it makes moves; the location it searches for language worm patterns (these are best searched in the vicinity of what's just been unclocked). If the graph is constructed according to *Generating An Instance (II)*, then not much searching will usually be required because the construction will have been from Fundamental Inference Patterns. The computer will already know where most of the key inferences are located. If it's search time is exhausted, it uses the Cloak Snapper (just like a player does).
5. Each player constructs a graph. Each player has the same resources to construct the graph. At easier levels, the resources available for graph construction are Fundamental Inference patterns. They link together only 'legally' and the player puts a set of them together Mechano-like. The construction must not only be 'legal' but capable of yielding a certain score when played. This encourages the player to understand the point scoring system and the ways, for instance, in which triangles or foursomes may link together to form high-point patterns. They play using the constructed graphs. The player with the

most points at the end wins. You should definitely be allowed to play your own graph. There's an important lesson or two there.

At higher levels of play, the resources the players have to construct their graphs are not Fundamental Inference patterns, but just individual links and satellites. Such graphs are harder to construct in a 'legal' fashion. A large educational objective of the game is to impart understanding about logic. By having to construct graphs, the players are playfully led into deeper understanding of the logical structure of the game.

6. At the **highest level**, all the training wheels come off. The logic checker is turned off. The disadvantage of the logic checker is that it does not allow complicated inferences not included among the Fundamental Inference Patterns.⁵ At this high level of play, you can uncloak satellites even if they're not safe to uncloak according to the logic of Fundamental Inference Patterns. The object of the game becomes identifying the Widgetian headquarters in the fewest moves. The Widgetians only start shooting at you if you take too long. This is the deluxe version in terms of challenge between players. Simply because it's the fastest in feel. It is also fastest of mind as the players begin to link sequences of Fundamental Inference Patterns quickly. They are more free, less brutalised by satellites, and may leap about like ghazals (or gazelles) in the fastest pace. The game can appear extraordinarily simple at this point in its graphics. More like a graphically simple chess game.

The **prototype** should attain this working level first before adding the logic checker or the player's Worm Based Satellite and Channel Construction Set (graph constructor) or much graphical sophistication or firing satellites. Making the logic checker and the player's graph constructor will be an interesting programming challenge. How does a worm search through its pattern and all possibilities? And if the player can construct graphs, how *do* those pieces fit together so magically? And how can the computer play a fast, good game against a player? These are questions that key on understanding the method in *Generating An Instance (II)* and Fundamental Inference Patterns.

ON THE INTERNET

What has been outlined above concerning 2-4 player games applies also to play on the Internet. Players could take part in tournaments or just hook up together briefly with a stranger to play a game. Scores should be kept of player performance and rating. You'd be updated concerning how many satellites the competitors still have to reveal before finishing and how many points each of the opponents has scored thus far.

War of the Language Worms, like many of the other puzzle/games we have devised, is unusually well suited to being played on the Web wordscape. The speed of the Internet currently precludes graphically intensive multi-player games. Our puzzles are not graphically intensive. Moreover, they do not rely so much on speedy eye-hand co-ordination as upon deductive skill (though there is an element of quickness, of speed play also). This series of puzzle/games could be on the vanguard of computer games used within schools and on the Internet.

SAMPLE WORDSCAPES

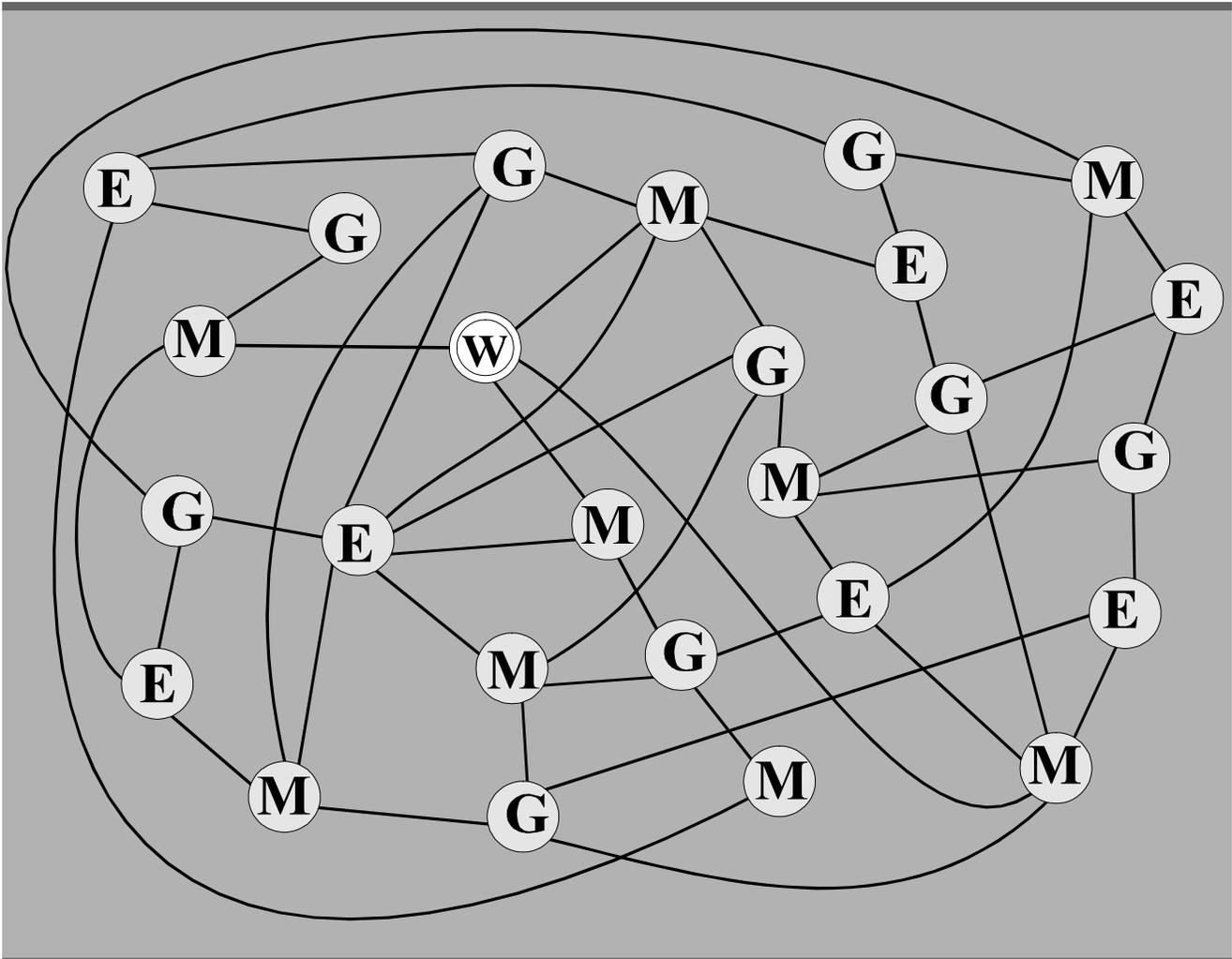
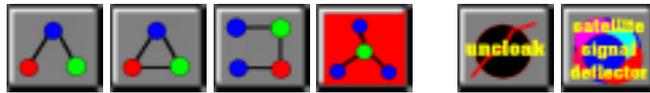


Figure 8: A sample board



Cover the satellites with dimes or big diamonds and have at it.

It's possible to score 78 points.

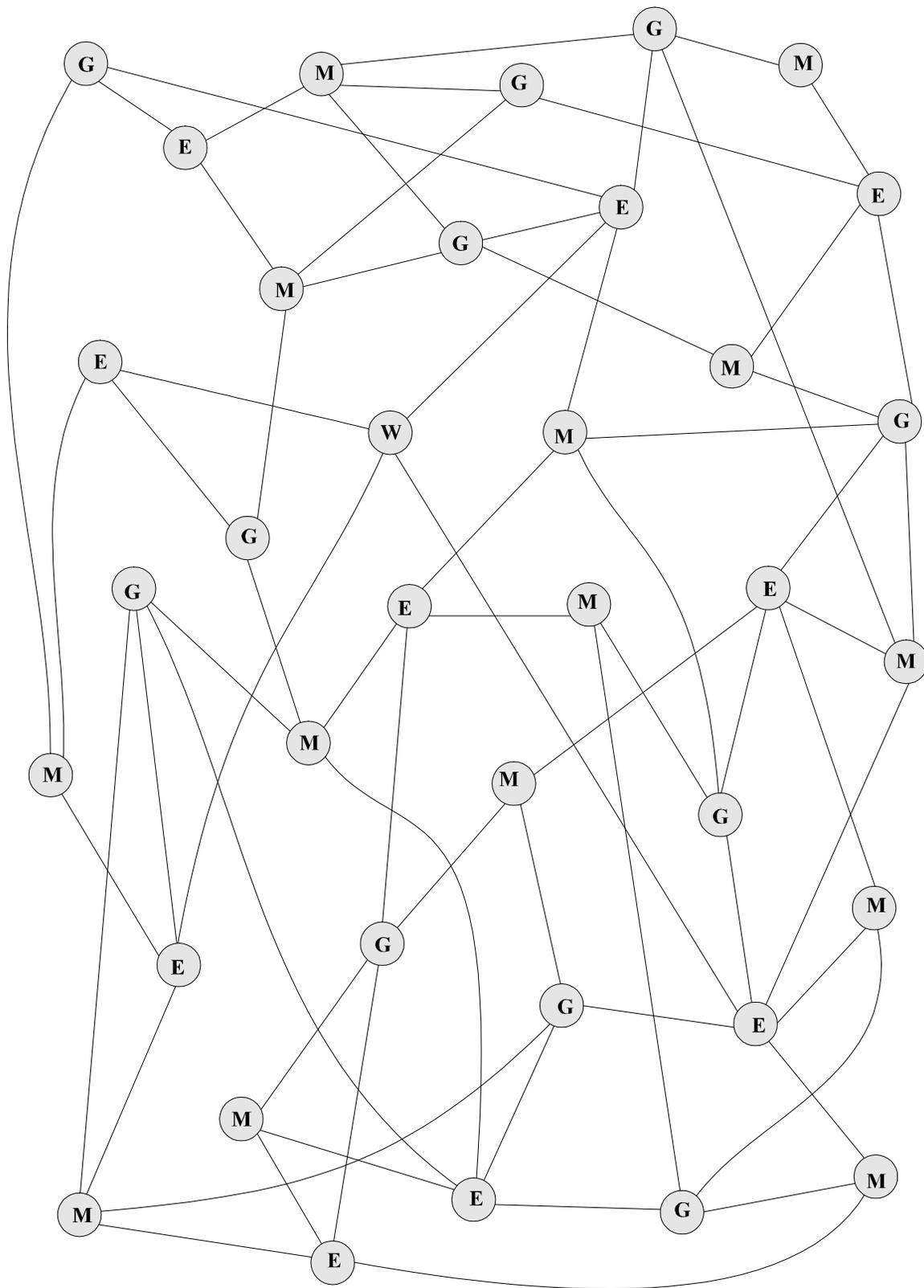
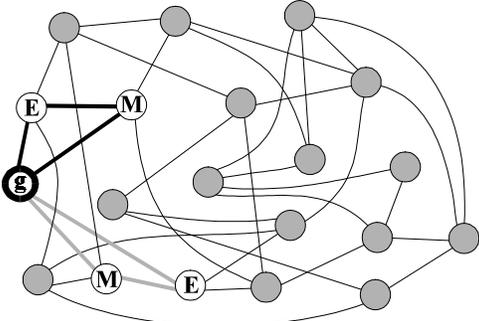


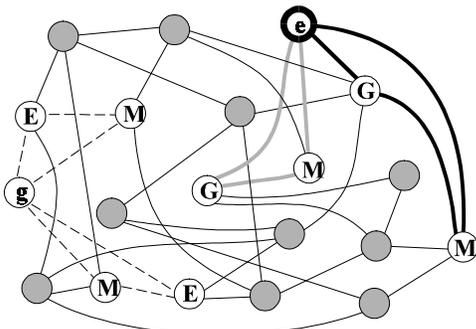
Figure 9: No guessing required.

SAMPLE PLAY-THROUGH WAR OF THE LANGUAGE WORMS

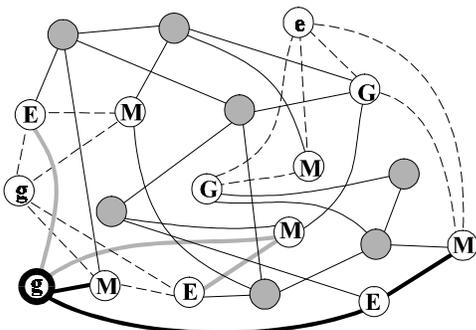
Trigger the Triangular language worm.



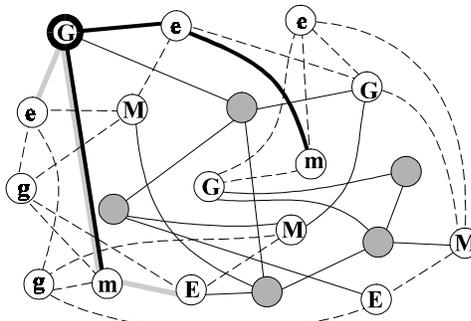
You jam 6 channels, penetrate 5 satellites (15 points) and g is now yours. The caps aren't yours yet. There are Widgetian channels through them.



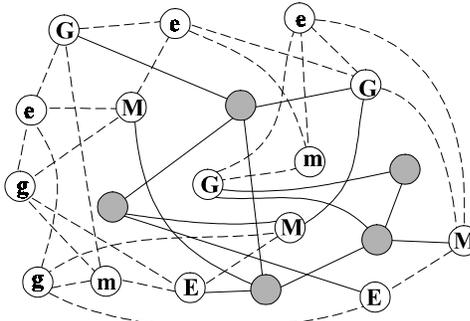
You hail a channel, open bay doors, unloose etc. the Triangular language worm. It jams 6 channels, penetrates 5 satellites (15 media power points). The heavy e satellite is yours. It won't shoot at you anymore.



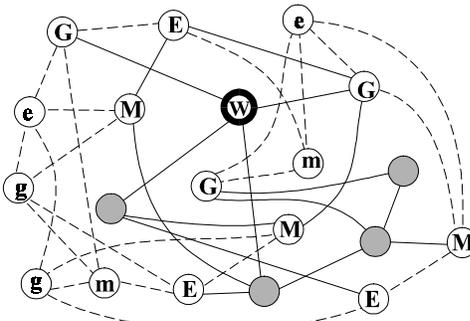
You score a double foursome (heavy black and heavy gray). 9 points. And g is yours. There's two more Foursomes connected to heavy g, but they don't uncloak any new satellites or free any unfreed info channels, so I haven't diagrammed them. But the player would see all 4 foursomes highlighted in turn.



You score with the Foursome. 6 points. Notice that 2 new m's are now yours and won't shoot anymore and so are two new e's. There are other foursomes attached to heavy G but they uncloak no new satellites nor free any new info-channels so I left them out.



Now you should know where the Widgetians are. The two cloaked satellites at far right are part of a triangle and two of the others are open to the Fork.



So you nuke the Widgetian headquarters with the Anti-Widgetian Language Worm. The game ends. You done good. You probably have the most points and therefore win.

Figure 10: Sample play-through

GENERATING AN INSTANCE (I)

STEP 1: DETERMINE THE NUMBER OF SATELLITES AND LABEL EACH SATELLITE WITH A GROUP NAME. ALSO, PICK A SATELLITE TO BE THE WIDGETIAN BASE.

Begin with n vertices (satellites). n should probably be between 8 and 40. Visit each satellite and randomly assign a group to it (Edu, Mil, or Gov). It's OK to have a different number of Edu than Mil satellites, etc., but the relative number of satellites per group shouldn't be wildly unbalanced. This will lead to a situation where the satellites of the group with the fewest satellites have *many* links connected to them. For instance, if there are only 2 Edu satellites and 10 Mil satellites and 10 Gov satellites, at least one Edu satellite will end up with 9 or 10 links attached to it (too cluttered). An easy out is to make sure that each group has the same number of satellites (± 1 or 2).

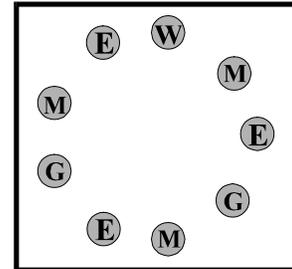


Figure 11

The satellite marked with the W is the Widgetian headquarters that will be linked to only one group.

STEP 2: CONNECT THE WIDGETIAN HEADQUARTERS TO SATELLITES OF ONE GROUP

Randomly select a group to connect to the Widgetian headquarters. How many communications links should you form? Initial investigation suggests 3 or 4 if possible. But a **prototype** is required to further investigate the matter.

During the rest of the generation process, don't connect the Widgetian headquarters to any more satellites.

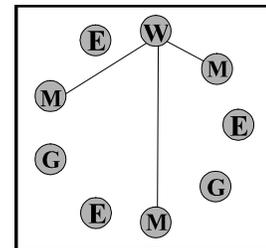


Figure 12

STEP 3: FORM "LEGAL" LINKS WITH PROBABILITY P

After we're finished with step 2, a "legal" link connects two satellites of different groups. There are no links between satellites of the same group. So for each possible "legal" link, form it with probability p .

Initial investigation suggests that relatively *challenging* games⁶ have a lot of satellites connected to 4 links, fewer satellites connected to 3 links, and fewer satellites still connected to 2 or 5 links. So a good value for p in a **prototype** would be $p=3.9/((0.66*\text{Total \# of satellites})- 1)$. This results in an expected value of approximately 3.9 links per satellite (assuming that there are approximately the same number of satellites per group).

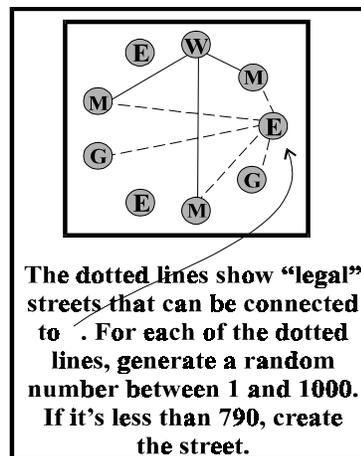


Figure 13

The higher the ratio of links to satellites, the easier the game is. So the number of links and the number of satellites and the ratio between them are definitely ramping parameters. A **prototype** where these are adjustable would be useful.

STEP 4: IF ANY SATELLITES (EXCEPT THE WIDGETIAN HEADQUARTERS) ARE NOT CONNECTED TO EACH OF THE OTHER GROUPS, CREATE LINKS.

After step 3 is finished, it's possible to end up with the diagrammed situation. Notice the bottom satellite isn't connected to at least 1 of each of the other two groups. This won't do. Every satellite (except the Widgetian headquarters) is connected to at least one satellite of each of the other two groups.

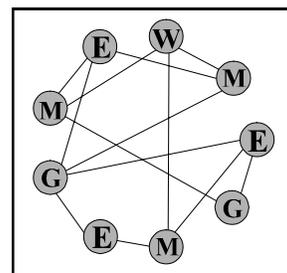


Figure 14

So if this happens, add links to get rid of the problem.

THE FINISHED GRAPH

Now you can make your own. Use pennies to cover the satellites. Good luck. You may also wish to make graphs that investigate some of the ramping parameters. And once you get really good, see if you can find any new Fundamental Inference Patterns. You'll want to bone up on those first, though. And read *Generating an Instance (II)* for a cagier method of constructing a graph.

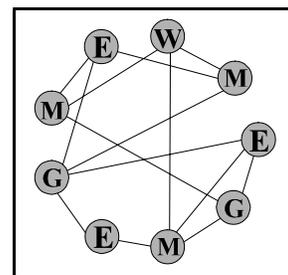


Figure 15

GENERATING AN INSTANCE (II)

The previous section outlines one method of generating an instance. There is a way, however, that may provide much more control over the difficulty of the puzzle and knowledge on the computer's part on how to play it. The idea is to construct the graph from the objects diagrammed at the beginning of *Fundamental Inferences and Logic Checking*.

The diagram to the right suggests the process. Don't add the Widgetian headquarters until the end of the construction.

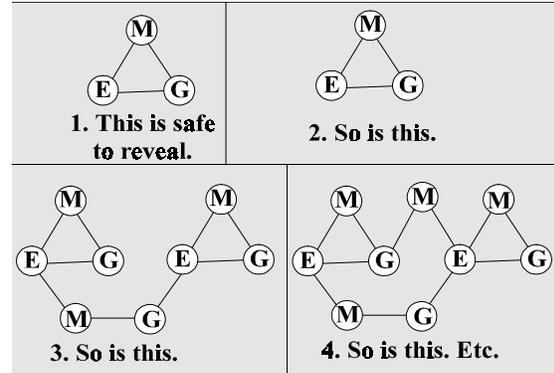


Figure 16

This method guarantees that there exists at least one sequence of deductions that doesn't involve any guessing *at all*. Unlike the diagram, the satellites within a triangle may be widely separated and not obvious (like the four interlocked triangles in the big sample game).

This method also allows the program to calculate the *typical* number of points you can score on the graph.⁷ This may be useful for assessment purposes—as well as providing players with a standard to measure their play against. “Ah, this is a 70 point Wordscape. Let's see if I can get close to it or even more points.”

The method may also be important to ramping: suitable constraints on the construction could guarantee the existence of relatively few available inferences at any stage in the game. On the other hand, at lower ramping levels, the generation algorithm could easily produce a puzzle that almost always has lots of available inferences.

The algorithm would begin with random placement of triangles. A triangle's satellites need not be close together. Then other fundamental inference shapes are added to connect 'legally' with the triangles or other fundamental inference shapes previously laid down. Inadvertent creation of new triangles or foursomes will result in the graph possibly being worth more points than the construction indicated. Finally, the Widgetian headquarters would be added and connected to satellites from only one group.

THE LOOK OF THE GAME AND PROGRAMMING

Constructing an instance abstractly is one thing. Making it look other than ugly on the screen is another. Much of this matter has to do with the graphic artists. But another part is how the programmers handle the drawing of links. If only straight lines are possible as links, problems might arise (like they don't already?).

If only straight lines are used for links, then the satellites must either be arranged in a circle (which severely limits the number of possible satellites on a screen) or ambiguity arises when links drift *over* satellites. Links should be able to connect satellites in random fashion. In other words, any two satellites housing different groups should stand at least a chance of being connected. This makes for a richer game, a more exciting game because the class of possible graphs is larger than otherwise. This may not be possible practically, but that's the ideal. Also, circuitous bezier curves of many different links provide some of the challenge (possibly at higher levels?).

The program must essentially draw elliptical or bezier arcs between many satellites. There do exist programs for quickly calculating and graphing curves given certain constraints—like *stay away from satellites you don't want to pass through*.

A more efficient solution, possibly, is to artfully examine the question of how many satellites a typical modern computer screen will accommodate. The satellites shouldn't be so small that they're a pain to click on. Consequently, the number of links must also be limited. Otherwise the screen will be too cluttered and ugly.⁸ Elsewhere in this paper a limit of 8 links per satellite was imposed. This limit of 6-8 links per satellite (maximum vertex degree) was imposed also with keyboard navigation in mind. But, typically, a satellite should be connected to 2, 3, 4, or 5 links. 3 or 4 more often than not.

Given these constraints, I suspect that the appropriate maximum number of satellites/screenful is 30-40 depending on how much room is allotted to the links.

Another possible solution to the clutter problem is to allow 'canal' routes: if a link connecting satellites A and C passes underneath satellite B so that it's immediately obvious graphically the link does not connect A and B nor B and C but only A and C, then all the better! But this will require careful co-ordination between graphic artists and programmers.

FOOTNOTES

¹ Why aren't Mil satellites connected to any Mil satellites (for instance)? Well they can't be or the logic of the game gets screwed up! But it's implausible for them not to be connected. Solution: Emphasise that the player is out to free the communications channels that have been captured by the Widgetians. They've captured the channels *between* groups. They haven't captured the channels *among* individual groups. Neither are those channels relevant to the game (though they are to the plausibility of the story).

² During the first few levels of the game, the language worm could complain to the player that its being misused and tutor the player in its proper use.

³ It is possible to have more than 3 different types of groups. But we haven't investigated this possibility yet. It's likely that the ratio of links to satellites would have to be increased. In other words, the average number of links per satellite would have to be increased to make for a situation where you don't have to guess very much. The logic checker would simply search among a new set of Fundamental Situations. But these would probably be too time consuming to search for and would disturb real-time play.

⁴ While in competitive mode, the game probably shouldn't practice too much pedagogy. Let the player's at it. Also, the sound files of characters speaking, etc., should probably be off in competitive mode except for 'sound effects.' It's also important that story not get in the way of quick, fair play. The player can concentrate on story and learning the game's subtleties when playing solo.

⁵ "Guided proofs" in High school Geometry have been found pedagogically useful. Although they admit of only one route to the proof, they give big hints to the student and teach the rules of outlining a general argument more effectively than when the student does not start out with such templates or training wheels. The logic checker outlined in this paper is advantageous over the guided proof in that it admits many routes to the solution (isolating the Widgetian headquarters).

But it also has a limitation: it's computationally intractable for the logic checker to determine whether a given, arbitrary move represents a valid inference. So the sphere of 'valid inferences' is too large for the computer to deal with. Hence, if it's to do any logic checking at all, it must check for only relatively simple inferences.

The requirements we place on the set of Fundamental Inference Patterns, then, are these:

- The set must encompass the inference patterns that all but the most advanced players will discover (advanced players can play without the logic checker being turned on).
- Each of the fundamental inference patterns must be simple enough that they do not hog too much CPU search time. If they do, then the fabric of real-time play will be badly wrinkled. Although I could be wrong, I believe the current set is sufficient. Prove me wrong. If you prove me wrong, I hope your inference pattern is tractable computationally!

⁶ A *challenging* game is one in which you don't have to guess very much yet the possible inferences are relatively few. And yet careful observation can discover a few interpenetrating triangles and/or interpenetrating foursomes. The game can be made less challenging by increasing the ratio of links to satellites and by decreasing the number of satellites. In other words, by increasing p and decreasing the number of satellites.

⁷ As long as no new triangles or foursomes are introduced inadvertently, I conjecture that the construction yields the *maximum* number of scorable points. But that's probably not helpful because its hard to see how to avoid introducing new triangles and foursomes. No matter. If nothing else, the score the construction generates is useful. I generate a graph worth 80 points. I know it's worth at least that cause I made it that way. Now I give you a handful of Fundamental Inference Patterns. The same ones I used. Make a graph out of those F.I.Ps that you can score 80 points on!

⁸ The scenario may take up more than one screen. Zooming in and zooming out is also an interesting possibility to consider.