6. Conclusion

From the systematic way in which the next trial arrangement is constructed it is clear that the number of Cows must be 0, 1, or 2. Simple arguments show that the conclusions drawn in each case are correct. It is interesting to find the worst number of trials for a given M (number of colors) and N (number of positions), called Mastersize. We have tried a large number of examples on our program and it has never taken more than 12 trials for M = 8 and N = 5. It appears that, in the worst case, the algorithm takes less than (M + N) attempts. It is quite pleasant to observe that repetitions in the secret code actually reduce the number of attempts.

References

[1] Nilsson, N., *Problem Solving Methods in Artificial Intelligence*, McGraw-Hill (1971).

[2] Winston, P.H., *Artificial Intelligence*, Addison-Wesley (1977).

[3] Winston, P.H. and Horn, B.K.P., *Lisp*, Addison-Wesley (1981).

TURTLES AND DEFENSE

The following letter, which was the response to a request for information about the use of AI hardware for defense purposes, arrived through the ARPAnet after passing through many different sites (more than 4). We received permission from the authors to include it in the newsletter. They said that the time (3:05 AM) was the actual time the letter was written - *Ed.*

3:05am Tuesday, 19 January 1982

William Schubert
Center for Defense Analysis
SRI International
333 Ravenswood Avenue
Menlo Park, CA 94025

Dear sir:

We must admit to some initial puzzlement at receipt of your communique of 6 January regarding possible applications of our robotics and artificial intelligence products to military functions.

However, always eager to contribute to the defense of our country from the ever-present threat, we put our best minds right to work on the problem and put together the enclosed report. We hope it will aid in your analysis.

Please note that the information we are providing is to be used only in your analytical studies, and is *not* to be considered an official offer by Terrapin to supply the systems at the quoted prices.

Please call us at (617) 492-8816 if you have any questions.

Sincerely yours,

Patrick G. Sobalvarro
Leigh L. Klotz
Senior Software Engineers
Terrapin, Inc.

Introduction

At Terrapin, we feel that our two main products, the Terrapin Turtle®, and the Terrapin Logo Language[1] for the Apple II, bring together the fields of robotics and AI to provide hours of entertainment for the whole family. We are sure that an enlightened application of our products can uniquely impact the electronic battlefield of the future.

The Terrapin Turtle® is a small, versatile robot that can perform any number of complex tasks under computer control. A powerful AI programming language is necessary to realize the full potential of this advanced device.

The Terrapin Logo Language, developed at the Massachusetts Institute of Technology Artificial Intelligence Laboratory's Logo Group, is ideal for this application. The Logo language is a close relative of Lisp, the language used most widely in AI research. It fills the bill quite handily!

Descriptive Information

1. Functions the system might perform.

While somewhat limited in range,[2] Turtles show great promise as all-terrain, high-accuracy system with excellent survivability in hostile environments. The Turtle's low observability, low vulnerability to ECM, and its multidirectional sensing capabilities make it an ideal reconaissance vehicle. Its ability to perform complex terminal maneuvering while pushing a heavy payload makes it a superb delivery vehicle as well.

Survivability

The Turtle enjoys very low observability, due to a minimal radar cross-section and an almost non-existent infra-red signature.

[1]Logo trademark under license from Bolt, Beranek, and Newman, Inc.

[2]Extended-range Turtle Mark I variants are currently in testing, however. See the section on *Range*.

In addition, its ground-hugging characteristics maximize terrain masking, resulting in lowered target acquisition by most classes of SSM and ASM threats. Its sophisticated sensors generate virtually no radar signature, so antiradiation missiles are useless against it. Furthermore, the Turtle is completely invulnerable to high-altitude SAM attacks.

With its powerful motors, the Turtle has very high initial acceleration. It climbs 45-degree grades with ease. Its maneuverability more than compensates for its somewhat low cruising speed. *The Turtle can make 180 degree turn in less space than any military vehicle currently in use by US forces, ground, air, or sea.*

With minor modifications, a Turtle could be constructed that could double its cruising speed for a terminal "dash" capability that would greatly enhance survivability in the endgame.

Because of the Turtle's compactness, with proper camouflage, weapons that depend on human vision are ineffective against it, as ground troops will find it hard to spot even at very close range.

Even if a suitable counter were found to all these properties of the Turtle, it seems dubious that an enemy could afford to deploy counterweapons in sufficient numbers to nullify the possibility of defense saturation in the event of an all-out Turtle attack. The low system cost of the Terrapin Turtle® make it an economic way to defeat defensive systems of increasing technological sophistication.

Range

The Mark I, Mod 0 Turtle has an effective range of some 3-4 meters, depending on its winding count. Range is most severely limited by the Turtle's C3[3], but this limitation is trivial by comparison with the inherent advantages of wire-guidance. These advantages are discussed further below (see section *Guidance*).

Furthermore, our research department is currently engaged in the testing of a 100-mile C3 for the Turtle. The thrust of this research is towards the development of an Extended-Range Turtle II®. While this does result in a shorter tooth-to-tail ratio, we feel it could significantly enhance the battlefield capabilities of Turtle installations.

Guidance

The Terrapin Turtle®, like many missile systems in use today, is wire-guided. It has the wire-guided missile's robustness with respect to ECM, and, unlike beam-riding missiles, or most active-homing systems, it has no radar signature to invite enemy missiles to home in on it or its launch platform.

However, the Turtle does not suffer from that bugaboo of wire-guided missiles, i.e., the lack of a fire-and-forget capability. Often ground troops are reluctant to use wire-guided antitank weapons because of the need for line-of-sight contact with the target until interception is accomplished. The Turtle requires no such human guidance; once the computer controlling it has been programmed, the Turtle performs its mission without the need of human intervention. Ground troops are left free to scramble for cover.

[3]Computer Control Cable, often familiarly called the Turtle's "tail" in civilian applications.

But why stop there? Even more interesting scenarios are readily envisionable.

Because the Terrapin Turtle® is computer-controlled, military data processing technicians can write arbitrarily baroque programs that will cause it to do pretty much unpredictable things. Even if an enemy had access to the programs that guided a Turtle Task Team®, it is quite likely that they would find them impossible to understand, especially if they were written in ADA. In addition, with judicious use of the Turtle's touch sensors, one could, theoretically, program a large group of turtles to simulate Brownian motion. The enemy would hardly attempt to predict the paths of some 10,000 turtles bumping into each other more or less randomly on their way to performing their mission. Furthermore, we believe that the spectacle would have a demoralizing effect on enemy ground troops.

Since the Terrapin Logo Language incorporates list structure, it is conceivable that someone will someday write an intelligent general problem-solving program in it. If this happened, truly intelligent guidance of Turtles would be possible, and since everyone knows that "smart" weapons are superior to "dumb" weapons, the Terrapin Turtle ® is superior to every other weapon developed to date.

Munitions

The Terrapin Turtle ® does not currently incorporate any munitions, but even civilian versions have a downward-defense capability. The Turtle can be programmed to attempt to run over enemy forces on recognizing them, and by raising and lowering its pen at about 10 cycles per second, puncture them to death.

Turtles can easily be programmed to push objects in a preferred direction. Given this capability, one can easily envision a Turtle discreetly nudging a hand grenade into an enemy camp, and then accelerating quickly away. With the development of ever smaller fission devices, it does not seem unlikely that the Turtle could be used for delivery of tactical nuclear weapons.

General

When controlled from a properly programmed computer, the Turtle can solve mazes, push and follow blocks, and draw complex geometric figures. In military terms, this means it can travel through mine fields and repel advancing troops. Large quantities of Turtles could conceivably be used to simulate battle plans. (Terrapin, Inc. does not supply software for these applications.)

The Terrapin Turtle performs the following functions:

Forward <centimeters>
Back <centimeters>
Left <degrees>[4]
Right <degrees>
Penup
Pendown
Eyeson
Eyesoff
Hornoff
Hornhi
Hornlow

[4]Convertible to mils for military applications.

2.System Unit Procurement Cost

Item	Unit Cost	Quantity
Turtle, Assembled	$599.95	1
Turtle, Assembled	$637.17	259
Turtle, Assembled	$599.95	10,000
Turtle, Kit	$399.95	1
Turtle, Kit	$424.76	259
Turtle, Kit	$399.95	10,000
Apple II Interface	$199.95	1
Apple II Interface	$212.35	259
Apple II Interface	$199.95	10,000

3.Installation Cost

The Terrapin Turtle is designed for installation at no cost by children and elementary school teachers. We feel that military installation cost should be under $10,000/unit.

4. Annual Cost of expendable supplies and spares per unit.

Ball-point Pen refills$0.59(one spare included)

In the rugged terrain of the battlefield, under rigorous load conditions, it maybe necessary to occasionally replace the Turtle Tires®. Due to fluctuations in the world rubber market, quotation of exact prices is not possible.

5. Annual-hours of labor required for operation and maintenance of one unit.

Operation costs for a Terrapin Turtle® can essentially be reduced to cost of programmer time. Terrapin, Inc., offers demonstration software with every Apple II interface. If the functions provided by this software are satisfactory for military purposes, then no man-hours of programmer time need be tallied.

In any case, we feel that is may not be fair to consider operation costs of single units, because we purposely designed the Terrapin Turtle® in such a manner that *every* Terrapin Turtle® can execute the same functions as every other Terrapin Turtle®. Thus there are no problems of software incompatibility, and a single team of programmers can, theoretically, operate *any number* of Turtles.

The Turtle's high reliability components and rugged construction minimize its life-cycle costs.[5] We feel that this should contribute to ahigh weaponsontargetperdollar potential. However, the Turtle's C3 will sometimes require maintenance, and, occasionally, rotation. Careful programming of the Turtle can reduce the need for rotation significantly. The challenge, simply stated, is to develop sophisticated, yet flexible, routing performance boundaries so as to maximize near-zero winding counts.

With the careful design of programs with provisions for evaluation of field-operation and recertification test results, together with failure/discrepancy analysis and corrective action, we feel that it should be possible to optimally extend the scheduled maintenance intervals and actually effect performance exceeding specifications. It can be expected that with a conscientious implementation of such programs very little time need be spent in C3 maintenance.

This estimate was, of course, computed taking only peacetime sortie rates into account. Under wartime conditions, we feel that upto a six-to-one increase in sorties might be experienced, and thus operationally ready (O.R.) rates can be expected to decrease accordingly.

6. System Characteristics

The Terrapin Turtle® weighs approximately 32 ounces. It measures approximately 8" in diameter, and is about 6" tall.Its power requirements are 15 volts at 2 amps. In normal trim, its power output is about 27.4 watts; however, an enhanced Turtle with"dash" capability would necessarily have higher power input and output specifications.

[5]It has long been the contention of our public relations department that the Turtle maintains itself. We feel that, if this is true, with a little work, Turtles might be made to maintain other devices, such as tanks, helicopters, and aircraft carriers. Thus we feel that a *negative* figure might even be appropriate for maintenance costs.

Impressions from the 1982 ACM Symposium on Lisp and Functional Programming

N.S. Sridharan, Rutgers University

September 1982

Dominant Impressions

I returned from this conference with three distinct impressions: firstly, having Alonzo Church, Barkley Rosser and Haskell Curry together as Honored Guests of the conference; secondly, being totally entertained by the "performer" Peter Henderson presenting a rational reconstruction of an Escher woodcut using recursive function definitions[1] ; and feeling that there were a number of astute British scientists who were thinking about foundational issues, who may soon make fundamental contributions to novel ways of computing!

Themes

Many people associate the beginning of interest in Functional Programming with Backus' Turing Award lecture titled "Can programming be liberated from the von Neumann style?". In that lecture Backus showed how to program with Functionals, that is with function producing operators, rather than with Functions, that is operators that map objects to objects. The advantages of functional programming include:

1. the ability to express parallelism naturally;
2. the prospects of turning parallelism into speed through novel architectures;
3. programming at a level close to specification;
4. the possibility of operating on programs for optimization and compilation;
5. the fact that the absence of side-effects makes program analysis easier.

There was a conference on Functional Programming and Computer Architecutre held at New Hampshire last Fall. But the roots of functional programming are older than this. McCarthy's definition of Pure Lisp, which includes function definition by abstraction,

[1]The irrelevance of his talk to the conference did not seem material!