OPPORTUNITY FIRE IN TURN-BASED PLAY

The algorithms which drive the AI in Turn-based play have never been documented, but every program must necessarily follow some logical path. I undertook this effort in an attempt to determine what this "logic" might be, specifically as it applies to ***artillery***. As a retired computer programmer myself, I approached the process as though conducting a test, not playing a game. Data collection and analysis was the first step. However, the results are only of value if they can be utilized to improve game play.

The values for the various weapons types must be set at a level which is correct for Phased-based play, which by definition – in the case of artillery at 1 hex – is too weak for Turn-based. In addition the certainty of defensive fire in a scenario played as Phased is supplanted by the random control of the AI in Turn. There are other considerations that mean little or nothing in Phased-based play, but carry implications in the outcome of Turn-based: movement in stacks and to a lesser extent the constitution (size) of artillery units.

How does the AI work?

If anyone reading this document has the definitive answer, I would be quite interested in their input. Based upon my observations during testing I can only offer a proposition. Jumping ahead, and providing a general outline of said observations, allows a better understanding of the results. Without documentation – or access to the actual program code -- one can only make an educated guess regarding the AI. Here is mine:

What is the chance of drawing fire?

If a unit moves within or into a hex, a check is made to determine if it is visible to the enemy. If not, then continue.

If visible, are any units within range as determined from PDT data? If YES, then the chance of receiving fire is based upon some specific percentage. This percentage is always the same for a given distance, but varies dependent upon the firing weapon (artillery or small arms).

Why a fixed percentage for a given distance? Unless the program is constrained in some manner, it will fire equally across all hexes within range. One could attach no value to a given hex and have the program check every unit within range applying the same percentage. This would be very inefficient and would give the same results as making the check up front.

Determining the Firing Unit(s)

At this point the AI truly becomes a "black box". Starting with the nearest unit or stack to the target, the program checks units that can fire and uses some algorithm to determine the firing unit. Secondary fire from the same stack or another hex is possible, but at a lower percentage. If a single unit or stack fails the firing check, then the program goes to another unit or stack, likely priority being proximity to the target hex. It continues this process until a unit passes the firing check or becomes the default option. If this is correct, then this explains why an attacker can receive fire from a hex other than the closest.

1. The program has no apparent limit as to how often a unit may engage in opportunity fire. Because of my testing method a defending unit would often have the chance to fire multiple times. As a practical matter, however, a unit will not be afforded as many opportunities in regular game play. Units are marked as having fired initially, but are not restrained from further action. The display of the "Has Fired" message is merely part of the normal fire routine. For defensive fire the game engine either ignores or does not record the fact.

Given that opportunity fire only occurs at half value this makes sense. Units could never render adequate defensive fire otherwise.

2. Unfired units do not appear to be given any priority when determining opportunity response. It was not unusual to have a stack of guns respond 2 or 3 times and yet have only a single section listed as having fired. I believe this to be tied to the situation described in #1. If the program does not record the fact that a defensive unit has fired, how could it establish any priority? If it does record that a unit has fired, what would it do if all units in a stack had already fired?

3. Unit size does not appear to affect selection. This is similar to the situation described in #2 and has both tactical and scenario design implications. Whereas stacking a 100-man regiment with a 400- or 500-man unit has no effect in Phase-based play, doing so in Turn-based presents the opportunity for the AI to fire the smaller unit to the exclusion of the latter. In considering artillery, if the firing process normally inhibits more than 1 unit in a hex from responding, batteries are preferred to sections. One 4-gun battery is just as likely to fire as one 2-gun section, but with obviously better results.

Other considerations

As readers know, the AI makes no distinction when targeting a single unit or a stack of units. They are also aware that it will never fire at static units no matter how much one may wish. The trigger to execute the AI code is movement. Movement is a transient state. The normal state for ALL units is static. Thus, when the AI fires at a moving target 3 hexes away, yet not at the stack of 1,000 infantry sitting quietly in an adjacent hex, it is understandable. If one envisions all moving units to be doing so simultaneously, it also becomes easier to accept. *Terrain* and *elevation differences* did not appear to have any effect on firing potential, although I did not test these situations specifically.

Whether the game engine behaves exactly in the manner I have described previously is, of course, unknown. I do believe that, in general terms, it does. Simplicity in programming – and there is nothing wrong with this approach – probably carried greater weight than the development of some highly complex outcome. The purpose of this exercise was to attempt an understanding of the AI, not to engage in a critique of what is or could be. Later in this document I do offer an idea as to how the existing system could be improved with relatively modest coding changes with no expectation that they will actually occur. I extend these thoughts to include actual PDT changes for game play.

Testing Procedure

Since my main focus was on the functioning of artillery, I chose the Wheatfield to Roundtop scenario in Campaign Gettysburg to conduct the testing. It allows for the movement of a quantity of infantry across open ground against stacks of guns. I reasoned that this would provide the "best" test of the AI. Pickett's Charge was another possibility, but the massed Union guns presented a scenario which would not be a normal occurrence.

To facilitate testing I gave the CSA infantry movement factors of 48 and set the cost of terrain to "1". While this did not give them the capability to march all the way to Washington, it did allow for a great deal of random maneuvering in every test in order to give multiple opportunities to the AI. I could move them within range, out of range, back into range, sideways, forward, backward and finally close to 1 hex. This was the fastest method of collecting data since the AI only responds to movement and has no "knowledge" of the movement's intent.

I wanted to measure artillery and small arms fire separately. To do so I turned off small arms in the artillery tests by setting the range to 1 with effectiveness 0. I reversed the process for the infantry test. A final test combining both was made in an attempt to determine 1) if any priority existed for artillery vs. small arms as to firing unit, 2) did the firing of one type preclude the other, and 3) would pre-Melee fire be the same?

ARTILLERY

All Small Arms Turned OFF

Hexes to target [1]	**5+**	**4**	**3**	**2**	**1**	**TOTAL**
Test #1	8	16/3	27/7	55/18	19/8	117/36 (30.8%) [2]
Test #2	4	11/2	7/3	19/7	12/3	49/15 (30.6%)
Test #3	2	12/2	11/5	6/0	10/5	39/12 (30.7%)
Test #4	4	18/3	40/7	57/20	11/5	126/35 (27.8%)
Test #5	4	21/6	29/7	39/13	15/6	104/32 (30.7%)
Total 5+ Hexes	22 [3]					
Totals Tests 1-5		78/16	114/29	176/58	67/27	420/130 (30.9%)
		20.5%	25.4%	33.0%	40.3% [4, 5]	

INFANTRY

All Artillery Turned OFF

Max range set to 3

Hexes to target	**3**	**2**	**1**	**TOTAL**
Test #1	69/35	101/48	38/25	208/108 (51.9%)
	50.7%	47.5%	65.7%	

[1] Distance of target hex to *nearest* enemy hexes containing a unit(s) which could fire at the attackers. In some cases the nearest units did not fire; instead a more distant one responded.

[2] X/Y lists the hexes entered at the indicated distance (X) and the number of times receiving fire (Y). Most defensive fire came only from a single unit although occasionally a second unit might fire. No data was collected for this secondary fire since it was not the regular response and had no discernible pattern. Percentages reflect the average of all hexes/fires while an attacker moves from 4 hexes to 1 hex adjacent a defender. The percentage itself has no direct application, but it does illustrate a consistency from the AI which lends credence to the individual hex results.

[3] Artillery was set to MIN for defensive fire, yet it still fired at beyond 4 hexes. The differences between settings (Min, Med, Max) are unclear. It is clearly evident that 4 hexes is a breakpoint since the total fire recorded at that distance nearly equaled that of all 5+ hexes combined. This latter fire was very random across a range of hexes.

[4] Percentage chance that a unit will draw fire at the given distance as calculated from the data collected.

[5] Hex 1 totals do not include any pre-Melee fire results. Pre-Melee fire should be considered as an extension of firing at 1 hex distance. Firing units are determined by some random method.

NOTE: Of the things I learned in this process of testing AI opportunity fire, the findings outlined in superscript ([5]) were the most surprising. The problem with judging pre-Melee fire is that it is difficult to determine which unit(s) fired. Only once did I observe **all** units fire -- the program listed 3 results -- from the hex to be meleed. In all other cases it appeared to be a subset (1 or 2) of the stacked units. In some cases **NO** units fired from the targeted hex. This latter situation was encountered in 3 of 18 melees initiated against **artillery** only, 5 of 29 initiated against **infantry** only, and 7 of 28 initiated in a separate test against artillery and infantry **combined**. In some cases pre-Melee fire initiated from other than the hex to be meleed. In some pre-Melee situations units which were unfired *prior* remained so *after* completion. Most melees were conducted from a *single* attacking hex, although some were from multiples.

The following sentence is lifted from game documentation:

Select **Full Melee Defensive Fire** to have defensive fire conducted by the program against attacking units in melee at full strength instead of half-strength.

The text does not indicate *automatic* fire (even from unfired units), but the *implication* is of a consistent, not random, response. If one is the *defender* in a PBEM game, Melee resolution happens so fast in the replay that one can determine nothing other than the outcome. As far as the *attacker* is concerned, units are added to the Melee and it is resolved. The lack of defensive fire is welcome, if even noticed. By marking a hex for Melee in Turn-based play and adding attackers, one causes "movement" which appears to be handled in much the same manner as regular opportunity fire. Melee resolution stands as distinct and is the same in either Phase or Turn play.

Analyzing the data

After each test the game was exited without saving. The next test began as new. Therefore, all defenders were in good order with zero fatigue. How the AI works for "D" units or at various fatigue levels is unknown and cannot in any way be influenced by PDT or other changes.

Artillery:

Test #1 involved movement by single units, not stacks. Stacking only occurred at hex 1.

Tests #2, #3 involved movement by stack, but limited in scope. I was attempting to measure how smaller samples would deviate. These are what gamers normally encounter. AI functioning can seem to be *totally* random.

Tests #4, #5 involved movement by stack, but in larger quantity. The results track with Test #1 and validate that the AI makes no distinction (or certainly not one that is obvious) between stacks and single units.

Unless the recorded data is somehow an aberration -- I don't see how it could be since I simply logged AI fire results – it clearly shows an increasing pattern as distance to target decreases. The fractional breakdowns are clear (1/5, 1/4, 1/3, 2/5). Unknown is what percentage is used in calculating pre-Melee fire (50% ?), but it definitely is not 100%.

Would five more tests produce significantly different results? Doubtful, but lacking any other source of information, I assume that the data indicates how the AI is coded. If the data were only informative, it would not be of much use. The percentages can be used to alter PDT data, however, as will be detailed later in this document.

Infantry:

This consisted of one test moving as many CSA infantry against the USA line as possible. The chances that a defender will fire at 2 or 3 hexes is about the same; higher for 1 hex. Breakdown is (1/2, 1/2, 2/3). Infantry fire was set to Max.

Combined:

This test involved moving the CSA infantry as previously, but with all weapons turned ON. The frequency of fire was greater, but not significantly so. The algorithm did not appear to favor artillery over small arms, but fire was still restricted somewhat to only single units. The AI must consider separate probabilities, but I cannot make any determination as to how it does so. Pre-Melee fire still exhibited the *no fire* situation despite the combined arms. All that can be stated is that pre-Melee fire is subject to the same randomness as other opportunity fire.

Having collected the data, what now?

Ideally, artillery in Turn-based play should be able to generate the same firepower as is achieved in Phase-based. Though it is not possible to control the *frequency* of opportunity fire, changes to PDT values can add to the *severity*; this in an attempt to produce casualties more in line with Phase-based play. However, the focus will be on fire *factors* rather than *casualties* since the latter can be influenced by other considerations. Factors are constant.

Utilizing the concept of *expected value (EV)* -- defined as *the sum of the values of a random variable with each value multiplied by its probability of occurrence* – the following equation is true:

EV = (% possibility of an event) x (value of the event)

In this particular case we will be adding potential results for hexes 4 through 2 as though an infantry unit were moving in this pattern. First a few "rules":

1. A single gun will be used as a proxy for all guns. What is true statistically for one is true for all. Factors will differ between gun types, but the concept is the same.

2. Situations in which the program will *not fire* will be treated as though the gun fires with a value of zero (0).

3. The gun will "fire" once per hex at the advancing infantry unit.

Using a Napoleon as an example, and referencing the results I noted for the Artillery tests, it would have a 20% chance of firing at 4 hexes and an 80% chance of not firing. Referencing the PDT, the AI will fire at ½ value (2.75). The *EV* calculation is as follows:

EV = .2 (2.75) + .8 (0)

EV = .55 + 0

EV = .55

If this Napoleon could fire 100% of the time at a 4 hex distance, this is the factor it would use.

I would like to briefly go on a tangent and engage in a flight of fancy before returning to the topic. Given that the game engine is unlikely to be further changed, I will, nonetheless, document how I would do it using as much of the existing code as possible. Some of what I discuss can be applied to the current Turn-based play to better it.

Turn-based play would become a hybrid combining both the possibility of opportunity and the certainty of defensive fire as employed in Phase play. This would become the only option of play.

1. A separate Melee Phase would be the only game option.

2. The game engine stores data about every unit in the game in tabular form. I would add a column to this table which would be updated with every defensive fire in terms of factors. At the beginning of each turn this column is initialized to 0 for every unit. When a unit fires defensively, the program would update the table with the value of the factor used. For example, if a Napoleon fired at 2 hexes with a factor of 6, then the table would be updated by *adding* this value to the existing entry. If it had not previously fired defensively, then the updated figure would be 6. If it had previously fired at an attacker – let's say 4 hexes with a factor of 2.75 --, then the table value would reflect 8.75 (2.75 + 6). This same process would be used for all units firing using opportunity fire. Artillery firing in *counterbattery* would have those factors applied as though they were firing at infantry. Small arms would accumulate factors in the same manner. The maximum EV will be a unit's PDT factor at hex ONE.

3. When all movement and offensive fire is complete, the phasing player clicks to advance to the Melee Phase as is currently done. The program would branch to the current Automated Defensive Fire (ADF) as utilized in Phase-based play. The fire of any defending unit would be (PDT value – any accumulated factors). Using the Napoleon example in #2, firing factor at one hex distance would be 5.25 (14-8.75); at 2 hexes 3.25 (12-8.75). The difference must be > 0 for a unit to fire. When all fire is completed, a message would post to this effect and any Melee can be initiated. It would be handled as in Phase-based play (no pre-Melee fire sequence).

While the system I described above is unlikely to ever see the light of day, some of the logic therein can be used to affect some change in the current Turn-based system.

If we assume that the hybrid system described above uses the *same* factors for firing at hexes 2 and beyond as the current Turn-based, then the *EV* in all cases will be the same for both systems. The focus is upon hex distance ONE for the following reasons:

1. These games are won or lost at hex ONE distance. In a sense so were actual Civil War battles.

2. Melee can only occur from hex ONE.

3. The *EV* for the 2 current methods of play cannot be equal in hex ONE, if the factor used for artillery is the same in both. Phase-based play has certainty of fire; Turn-based does not.

4. Hex ONE is the only that is *visible* to the AI 100% of the time, thus a constant.

Visible (and non-visible) hexes

Given 2 possibilities (visible or non-visible) and 3 hexes (2, 3, 4), this creates ($2^3 = 8$) potential situations. Two such would be all hexes *visible* or all hexes *non-visible*. The remaining are in various combinations of which 50% of all hexes are non-visible. This fact impacts the potential *EV*.

Hybrid System

An infantry unit advances from 4, 3, to 2 hex distance. If a target hex is *non-visible,* fire occurs at a value of zero (0). The *EV* for all fire in non-visible hexes is 0 and can be dropped from our equation. *EV* is halved because of this, however. Using the percentages identified in the Artillery tests and modifying for visible and non-visible hexes, the *EV* equation is as follows (Napoleon fires at ½ PDT value):

EV = .5[.2 (2.75)] [4 hexes] *+ .5[.25 (5.5)]* [3 hexes] *+ .5[.33 (6)]* [2 hexes]

EV = [.275] + [.6875] + [.99]

EV = 1.9525

This value indicates the *EV* for all factors expended for all AI opportunity fire in this interval for all Napoleons. It says nothing about any specific occurrence.

This is recorded in our "virtual" table. This leaves an *EV = 12.0475 (14 - 1.9525)* at hex 1. With an automatic process this Napoleon is certain to fire this "factor" prior to Melee.

Current Turn-based System

The question is "What factor (*EV)* equates to 12.0475 in the problematic world of the AI?" The *EV* for interval 4 to 2 hexes is the same in both systems since it uses the same probabilities and factors. We do need to account for this in the final factor determination, however.

The data suggests a 40% chance of firing in hex 1; one can easily say that there is a 60% chance of *no* fire. Using my *assumed* 50% chance of pre-Melee fire (it could be 40%, 60%, who knows?), there are four possibilities:

1 30% No fire hex one (0) No fire in pre-Melee phase (0)

2 30% No fire hex one (0) Full fire pre-Melee phase (X)

3 20% Fire at ½ value (.5X) No fire in pre-Melee phase (0)

4 20% Fire at ½ value (.5X) Full fire pre-Melee phase (X)

Accounting for each possibility and then equating the outcome to 12.0475:

1 .3(0) + .3(0) = 0

2 .3(0) + .3X = .3X

[3] .2(.5X) + .2(0) = .1X

[4] .2(.5X) + .2(X) = .3X

EV = .3X + .1X + .3X = .7X

.7X = 12.0475

EV = (12.0475 / .7) or 17.211

The factor for a Napoleon at one hex in Phase-based play is (1.9525 + 12.0475) = 14

The factor for a Napoleon at one hex in Turn-based play should be set to (1.9525 + 17.211) = 19.163 (rounded to 19.2).

Dividing 19.2 by 14 equals 1.37.

Note: An additional change is made to the Turn-based Model PDT to limit artillery fire to hexes with at least a value of .5. This prevents the AI from needless wastage of ammunition.

Doesn't the change make guns too strong offensively at hex ONE?

Compared strictly to the value of the Phase-based model, one might say "Yes". The answer is "No", however, since any additional casualties would partially compensate for losses that did *not* occur in defensive fire. Specific reference is made to artillery which does not fire at adjacent hexes although it could and those situations where units are in range, but static.

Why only alter the hex ONE value?

Although the defensive firing factors for artillery in Turn-based play are ½ those used in Phase-based, this is only an issue when measuring the effect against enemy units which *end their move* within the range 2-4 hexes. How often in Phase-based play does one move his infantry within the said range and stop them there, allowing enemy artillery to fire without any chance of being meleed? Not often, I would guess. In fact it could occur from 0 to N times and would be an unknowable number. Regardless, the AI will not fire at a static unit. It would not be a stretch to assert that more artillery factors are likely expended in Turn-based play at the range 2-4 hexes than similar fire in Phase-based. In a sense the *EV* calculations basically aggregate *lost* factors into hex ONE. Whether *casualties* are incurred in hexes 2-4 or at the hex ONE level does matters little.

What if I want to use a PDT with different gun values?

The calculations would be the same, but the various factors different. Start with the [.2X + .25Y + .33Z] formula to reach the EV for 4,3 and 2 hexes. Subtract from the hex one value and divide by .7 as above. Alternatively one can simply multiply the hex ONE value by *1.37*. It will be close enough for government work – Federal or Confederate.

The model Turn PDTs assume all hex ONE fire takes place against infantry. If one is engaged in a scenario where certain guns would more likely fire at gunboats, I can't say whether there would be any great effect.

What about the infantry?

Small arms fire is ubiquitous in that it is used both offensively and defensively at the same time and a moving attacker will fire at ½ value, as will a defender. There is no way to alter defensive fire without having a

similar effect on offensive output. Since the odds of firing at 2 or 3 hexes appears to be the same, why not take the AI out of the equation altogether?

For small arms with a range of 3, the factors for 2 and 3 hexes are the same, that being the average of the two. For rifles the PDT would be changed from ***R 1 3 2 2 3 1*** to ***R 1 3 3 1.5.***

You've assumed a flat and clear battlefield. I'm fighting at Chickamauga.

Yes, this presents a problem. My purpose was to develop the probabilities, if they existed, for 2 hexes to 4 hexes. I am assuming that the game engine does not change these probabilities based upon the target hex. If it is *visible* it should exhibit the same probability of incurring fire as clear terrain. Terrain will affect *casualty* levels, but should not reduce probabilities or factors.

Let's assume a battlefield which is all *woods*. No unit is visible until it reaches hex ONE. The *EV* for hexes 4, 3, and 2 is zero(0) because the AI cannot detect movement and will never fire. How will this affect artillery?

In the hybrid "smart" system, the Napoleon would never fire prior to hex ONE, so it would have 14 factors available. Its *EV* is 14, not the 12.0475 which we used for setting gun values previously. The factor for Turn-based play is set in stone (19.2) and can't be changed. It too will never fire, but its value includes 1.9525 potential factors which it "loses" because the attacking units are not visible.

We can establish any effect by going back to our previous equations for hex ONE. Once again:

.3(0) + .3(0) = 0

.3(0) + .3(19.2) = .3 (19.2) = 5.76

.2(.5 (19.2)) + .2(0) = .1(19.2) = 1.92

.2(.5 (19.2) + .2(19.2) = .3 (19.2) = 5.76

EV = 5.76 + 1.92 + 5.76 = 13.44

Now 13.44 is not 14, but it is close enough. But this is a worst case scenario as though an entire battlefield is shrouded in fog in an eternal forest. Locally, maybe, but overall most battlefields will have a mixture of visible and non-visible hexes and will be closer to what I tested. Positioning your guns to maximize their field of fire becomes important.

What other changes can be made to improve Turn-based play?

Movement in Stacks: Even if one does not accept anything else contained in this document, the greatest effect on Turn-based play can be achieved with the following player-imposed restriction:

> Combat units must move individually in all situations. Any number of leaders can be stacked with said unit. A unit may start its move stacked, can move through another stack, and can end its move stacked, but may not move stacked with another combat unit.

I consider this a *standard of play* issue, not implementation of a house rule. The game engine enforces individual movement to obtain the benefits of roads, for example. It requires neither individual nor stacked movement. One could play an entire game using only individual movement, but players quickly learned that stacking was the best way to minimize any AI response. So, in a sense, it is engaging in "gamey" tactics to the greatest extent. One cannot "beat the system" in Phased play by stacking.

Design Considerations:

Gun batteries vs. sections: Opportunity fire appears to be determined randomly by *unit*, not composition or size. If true, a gun *battery* will produce greater results than its component *sections* all things being equal. This requires OOB changes in many cases, which cannot be affected until said OOBs are *unlocked*. This also simulates the difference between the more homogenous Union artillery and its mixed gun Confederate counterpart and its inherent difficulty of ammunition supply.

Supply Wagons: If one employs individual movement, the increased AI fire will drain supplies at a faster rate. One could modify supply wagons to carry perhaps 1/3 more ammunition. Again, this would require OOB changes. In lieu of this one can add supply wagons to any scenario.

Ammunition Change: I feel that the 1/24 ammo change ratio in the PDT is too low. Since LOW AMMO affects offensive fire, not defensive, this has a restraining effect on an attacker matching the historical precedents of actions which were affected by local ammunition shortages. In line with increasing ammunition supply of wagons, I would change the PDT to reflect a 1/18 change ratio.