Wargame: Airland Battle & Red Dragon

Game Scales

by Pyro

Update 9 September 2014: revised acceleration formula in step 7. Added time conversion. Added last paragraph.
Update 10 September 2014: revised derivation of constant in blue.
Update 13 December 2015: changed to Wargame: Red Dragon. Wargame: Airland Battle data in brackets [].
Update 7 May 2017: changed the name of SpeedScale to TimeScale

Introduction

This document describes the relationship between two important scales "Map Scale" and "Time Scale" and one derived variable "Fake Factor" in the *Wargame: Red Dragon* computer game. The same relationships occur in other related games in the *Wargame:* franchise, such as *Airland Battle*.

The three variables are remarkably clever and flexible. They allow game data to be reused even if the scales are changed. The three scales lie behind every conversion from game data to a distance, speed or acceleration.

There are three sections. The first describes the variables used in the formulas. The second describes the formulas themselves, with a few explanations. The third section has additional notes.

Variables

MapScale	the ratio of lengths in the game, e.g. the size of vehicles, to their real world lengths.
TimeScale	the ratio of how much faster units move compared to the their speeds on the unit information screen.
Multi	0: 115 [0: 105 in AB] MultiplicateurRTSVersDistanceFeedbackTactique in gdconstanteoriginal.ndfbin
FakeFactor	271: 1549 [280: 1618 in AB] SpeedFakeFactorVehicle 271: 1550 [280: 1619 in AB] SpeedFakeFactorInfantry 271: 1551 [280: 1620 in AB] SpeedFakeFactorAir
Distance	distance in the real world and on the map as measured by the cursor, in metres
Ddata	distance stored in the database, in game units
Dmultiple	some game distances must be a multiple of this number, in metres
Speed	speed in the real world and on the unit information screens, in km/h
Smeasured	speed in the game as measured by cursor and stopwatch, in km/h
Sdata	speed stored in the database, in game units
Accel	acceleration (or deceleration) in the real world, in km/h per second
Adata	acceleration (or deceleration) stored in the database, in game units
Time	time in the real world and as measured while playing the game, in seconds
Tdata	time stored in the database, in seconds

Formulae

An example is given for each formula, highlighted in yellow. The example is the default scaling used in *Wargame: Red Dragon* and *Airland Battle*. Often at each step the result for the game database variables will be a

strange number or non-integer. For programming sanity, it is desirable to round game database variables to a sensible number, so the reverse formula will also be shown. This means that TimeScale will not be a nice round numbers, but a close approximation to the desired outcome. A special constant is used, highlighted in blue. This value is described in the third section.

Step 1 – given MapScale, determine Multi:

Example: desired MapScale = 3.5

Multi = MapScale ÷ 10 = 0.35

Step 2 – given TimeScale, determine FakeFactor:

Example: desired TimeScale = 2.5

FakeFactor = { TimeScale ÷ 0.36 } ÷ MapScale = 1.98 *round this to 2*

FakeFactor does not have to be an integer; in this example it could have been rounded to 1.9 or 2.1, etc.

Revise TimeScale using rounded FakeFactor:

TimeScale = FakeFactor × MapScale × 0.36 = 2.52

Step 3 – determine TimeScale and FakeFactor for other unit types:

There are different TimeScale and FakeFactor for vehicles, infantry and air units. Repeat step 2 for each type of unit. You may use the same scales and factors for two or all three unit types, if you desire.

For vehicle and air: Same as the examples worked in step 2.

For infantry: FakeFactor = 3
TimeScale = 3.78

Step 4 – determine how distances are converted from the real world to database units:

Ddata = Distance × 260 ÷ MapScale = Distance × 74.285714 recurring

Step 5 – determine the distance multiple:

Some distances in the database are required to be in multiples of a particular number; this is the number.

Dmultiple = MapScale × 50 = 175 metres

Step 6 – determine how speeds are converted from the real world to database units:

Sdata = Speed × 26 × FakeFactor = Speed × 52 for vehicles and air units
= Speed × 78 for infantry units

There is effectively no minimum speed multiple. Sdata will accept even a value of 1.

Repeat this step for all three FakeFactors.

Step 7 – determine how accelerations are converted from the real world to database units:

Adata = Accel × 26 × FakeFactor × TimeScale = Accel × 131 for vehicles and air units
= Accel × 295 for infantry units

There is effectively no minimum acceleration multiple. Adata will accept even a value of 1.

Repeat this step for all three FakeFactors.

Step 8 – determine how time is converted from the real world to database units:

Tdata = Time ÷ TimeScale = Time ÷ 2.52 *All real world times should be converted before putting in the database.*

The Relationship between Map Scale, Time Scale and Fake Factor

Time scale, fake factor and map scale are closely related by formula. The formula may be rearranged so that any two of the variables can determine the third. The equations are:

TimeScale = FakeFactor × MapScale × 0.36

FakeFactor = { TimeScale ÷ 0.36 } ÷ MapScale

MapScale = { TimeScale ÷ 0.36 } ÷ FakeFactor

Some combinations that can be considered are:

- Time Scale = 1, Fake Factor = 2, Map Scale determined by formula to be 1.388. This has a Time Factor of exactly one, so there is no unrealistic discrepancy between speed of movement and rate of fire. This combination has the advantage of keeping the Fake Factor the same at two, avoiding the tedious necessity of changing all the speed and acceleration data. Additionally, the Map Scale is close to one and some say that even at 1.4 that measurements of vehicles on the map match real world data; it is unlikely that any small difference could be noticed.

- Map Scale = 2.6, Fake Factor = 2, Time Scale determined by formula to be 1.872. A judicious reduction in map scale slightly reduces available LOS, while still allowing long range weapons. Fake Factor is unchanged, so speed and acceleration data does not have to be replaced. With a Map Scale of 2.6 all the distance data will have to be changed, but the formula is much easier to remember as the distance data stored is 100 times the distance in metres, e.g. a distance of 450 m is stored as 45000. Time Scale has been reduced by almost half but is retained, which may help with play balance.

- Map Scale = 2.6, Fake Factor = 1.282, Time Scale determined by formula to be 1.200. A judicious reduction in map scale slightly reduces available LOS, while still allowing long range weapons. Map Scale and Fake Factor have both been changed, so all distance, speed and acceleration data will have to be replaced. With a Map Scale of 2.6 the formula is much easier to remember as the distance data stored is 100 times the distance in metres, e.g. a distance of 450 m is stored as 45000. With a Fake Factor of 1.282 the formula for speeds is also easier to remember, as the speed data is 100/3 times the speed in km/h, e.g. a speed of 60 km/h would be stored as 2000. Time Scale has been reduced to nearly one, and the small remaining difference would not be noticed.

Additional Notes – Map Scale

The map scale is the ratio of the size of the game map compared to the size of the various vehicles moving about it. The default Map Scale in *Wargame: Red Dragon* and *Airland Battle* is 3.5. If you reduce the Map Scale:

- the size of each map decreases.

- you will have to change all the distance data.

- the average LOS will be reduced because there is more blocking terrain per square km – this may be a good thing, more realistic in a European environment.

- you may find it more difficult to properly implement long range weapons such as artillery, as well as fast units such as aircraft.

Conversely, if you increase the Map Scale:

- the size of each map increases.
- you will have to change all the distance data.
- the average LOS will be extended because there is less blocking terrain per square km – this may not be a good thing, less realistic in a European environment.
- as there is a limit to the number of infantry units per city block, you may find that these few units are covering an unrealistically large area for the number of troops.

There is some speculation online as to what scale constitutes a 1:1 size, i.e. where a 6 m vehicle is actually measured as being 6 m long. Some say it is at a Map Scale of 1.0, but others say it is around 1.3 to 1.5. There are advantages to having a Map Scale of about 1.4 (see section above for example combinations).

Additional Notes – Time Scale

The time scale is the ratio of the speed that units move on the game map, as measured by cursor distance and stopwatch, compared to the speed information displayed in the unit information window. The default Time Scale in *Wargame: Red Dragon* and *Airland Battle* is 2.45 for vehicles and aircraft, and 3.68 for infantry.

One effect of having a time scale is that while it changes movement speeds, it has no effect on firing rates. For example, say there is a 500 m gap between cover, and a tank races across that gap at 60 km/h. In the real world that gap would be crossed in 30 seconds. An enemy unit with a rate of fire of 6 rounds per minute could get off 3 shots at the moving tank. However, if the speed factor was 2, then the moving tank would cross the cover in just 15 seconds. This would allow the enemy unit to fire one shot, possibly two. Therefore, high time scales may have to consider modifying rates of fire as well if a more realistic simulation were desired. This is done using the conversion formula in step 8.

Additional Notes – Fake Factor

Time scale, fake factor and map scale are closely related by formula. It is possible to change the time scale and the map scale without affecting the fake factor. This is important because if the fake factor changes, then you will also have to change all the speed and acceleration data.

Fake Factor and the corresponding Time Scale can be different for different unit types. This allows, for example, infantry to get an arbitrary boost to their speed (boost set at 50% using default settings), as otherwise it would mean excessive tedium as they move so slowly.

The Exact Determination of the Constant in Blue

The constant 0.36 in blue is simply one-tenth the conversion between km/h and m/s.

The value of 0.36 has been determined empirically by many tests using a stopwatch to measure foot, ground and air speeds under various combinations of Map Scale, Time Scale and Fake Factor. The error variation in this fraction is a maximum of ±3% and is due mainly to fat fingers being unable to time correctly.

Previous work thought that this value was 0.35 (which is within the error variation of ±3%). Further work with a stopwatch led to more data that supported 0.36 as the real value, not 0.35. Once it was realised that this could be related to the conversion between km/h and m/s, everything fell into place. Previous work connecting the value of the constant to the value in 218: (terrain): 1305 OneMeterInGameUnits turned out to be a red herring.

Determination of Formulae

Some formulae have been determined from known relationships. From these, the remaining equations have been determined mathematically, and then their conclusions tested.

How the Game Looks and Feels with these Changes

Changing Multi and Fake Factor has *no* effect on the look of the game – all units move exactly as they did before. The only change is to how information is *presented*, for example in the unit information screen or the range cursor. To change the way that units move, you have to change the actual game data for those units.