eTriops Technical Documentation
v.1.0r

Preamble:

The purpose of this document is to discuss the technical details of the eTriops software, to aid anyone who may be interested in testing, contributing to, or modifying the project. This document is NOT intended to be a general user guide, and likely won't be helpful to anybody who simply wants to play eTriops.

Logging:

The game will generate a log file in the user home directory named "etriops.log". This log contains time-stamped records of every major event a triops experiences. Additionally, if you are using the debug dump command (see below) those messages will be recorded in the log. When running the native Python script the logger will write both to the file AND to the console. Since the Windows EXE build does not provide a console, log messages will only be written to file. The log file is always opened in "append" mode, so occasionally deleting the log to remove old messages might be prudent.

Cheats and Debugging Commands:

The following hot keys are provided to allow special functionality for those testing or debugging the program. Please note that these characters must be capital in order to register (in other words, hold down Shift when using them).

Key	Description
P	Dumps debugging information to the console and log, including the current game state and the values of the core global variables. NOTE: The windows EXE does not provide a console, so you will have to refer to the log file (~/etriops.log) if you are running that build.
O	Force Egg Laying. NOTE: This will likely be commented out in main releases; you will need to un-comment the code in the interact() function to enable it.
I	Force Molting. NOTE: This will likely be commented out in the main releases; you will need to un-comment the code in the interact() function to enable it.

Game State:

The game state is encoded using a dictionary format to hold the game's core variables:

```
gamestate = {'name': 'unnamed', 'age': 0, 'tod': 0, 'hcap': 100, 'health': 100,
'hunger': 100, 'ammonia': 0, 'foodInTank': 0, 'eggs': 0}
```

name	Holds the name of the current triops
age	Age of the triops in days
tod	"Time Of Day" - the number of ticks since the current day started
hcap	"Health Cap" - the maximum health the triops can attain under current conditions
health	The current health of the triops

hunger	How “full” the triops is, food-wise. A lower value indicates the triops is hungry
ammonia	How much ammonia is currently in the tank
foodInTank	How much food is in the tank
eggs	Total number of eggs that the triops has laid

Game state is written to the file "etriops.sav" in the user's home directory. Game state is written to disk every 30 minutes, upon triops' death, and whenever the program exits.

Ticks and Increment/Decrement Rates:

The game ticks once per second. On each tick, the program will recompute the game state and update its values accordingly. After 86,400 ticks (the number of seconds in a 24-hour day) the triops' age will increase by 1 and the Time-Of-Day counter will reset to 0. Values are calibrated to increase/decrease at a rate that roughly emulates a real life triops:

Health	Increases at 0.08333 per tick, enough to restore 50HP over the course of about 10 minutes. This is to emulate a triops' natural healing/recovery ability.
Hunger	Decreases at 0.00092 per tick, enough to leave a triops at about 20 hunger after a 24 hour period.
Ammonia	Increases at 0.00008 per tick, enough for it to hit 100 in a little over 14 days. Emulates the slow buildup of ammonia in an un-cycled aquarium.
Health Cap	Half the value of the ammonia subtracted from 100. For example, if the current ammonia level is 10 then the health cap will be 100 - (10/2) = **95**. Meant to emulate how toxic water conditions impact a triops' overall health.

Special Cases:

- If health is already at the health cap then it won't increase.
- If hunger reaches 0 then health won't regenerate, and will in fact take a 0.00462 hit per tick. At this rate a starving triops (that is otherwise healthy) will die in approx. 6 hours.
- Any uneaten food pellets left in the aquarium will add an additional 0.00004 ammonia per pellet per tick. This is to emulate how overfeeding can lead to water toxicity.
- If a triops' age is under 3 days then it can't get hungry (and, subsequently, can't eat). Incidentally, this means that any food placed in to the tank will sit there and contribute to raising the ammonia levels. This is to emulate how real-world triops subsist on microbes when they are hatchlings.

Note that it is possible for ammonia to go over 100! Given how the health cap is calculated it's theoretically possible for ammonia to reach as high as 200 before the tank becomes mathematically uninhabitable. In practice, however, a tank with extremely high ammonia levels will put a triops at risk, as its health will be severely capped and it won't be able to cope with other health hazards (such as molting) as easily.

Random Events:

- On any given tick a triops has a 1 in 259,200 chance of molting. This rate is sufficient enough to molt on average once every 3 days.
- On any given tick a triops has a 1 in 259,200 chance of laying eggs. Like with molting, this should result in an average rate of once every 3 or so days. Note that triops only have a chance of laying eggs once they mature at age 14.

Molting:

When molting, a triops will take a hit to their health, with the amount of damage increasing with age. This is to emulate how molting for a real-life triops gets progressively harder as they get older. The algorithm for computing the amount of molt damage is:

```
baseDamage = round(random float (1,10), 5 decimal places)
totalDamage = baseDamage + age
```

However, for the purposes of computing molt damage the age variable is capped at 93. This is to ensure that a triops always has at least a very small chance of survival and there are no artificial limits on its lifespan. With age at 93, if the random int is anything under 5 an otherwise-healthy triops will just barely survive, whereas if the random int is above a 5 they likely won't.

Eggs:

When a triops goes to lay it has a random chance of producing anywhere from 10 to 30 eggs, which will be added to the triops' total egg count. The lifetime count will be reported after the triops' death and serve as a sort of final score. If the triops' health is between 20 and 40 then the number of eggs it would have laid gets cut in half. If the triops' health is less than 20 then it can't lay eggs at all. This is to emulate how poor health can impact vitality in a real-world animal.

Status Descriptors:

The exact numerical state of the game does not get communicated to the players on screen. Instead, textual descriptions are used according to the following table:

Age	
< 3 Days	Hatchling
3-10 Days	Juvenile
10-14 Days	Young Adult
14-30 Days	Adult
30-60 Days	Middle Aged
60-80 Days	Senior
80-90 Days	Elder
90 > Days	LEGENDARY

Health	
< 20 HP	CRITICAL
20-50 HP	Poor
50-80 HP	Moderate
80-90 HP	Good
90-95 HP	Excellent
95 > HP	Peak Condition

Hunger	
< 0	STARVING
0-25	Famished
25-50	Hungry
50-75	Peckish
75-95	Full
95 >	Stuffed

Ammonia (Labeled on-screen as “Water Quality”)	
< 7	Excellent
7-50	Fair
50-100	Dirty
100-158	Toxic
158 >	SEVERE TOXICITY

Animations:

While the program is running it will display animations in its main view panel. All the animations are a series of image files which the program displays in a successive pattern. Normally the program will display its “idle” animation sequence of a triops swimming around the screen. The exact graphics used for “idle” animation is determined by the triops’ age. Certain events (such as tank cleaning) will trigger a unique animation sequence.

Directory Structure/Naming convention:

The image files for the animation sequences are located in the “assets” directory, with each animation sequence stored in its own sub-directory. For example, the files for the tank-cleaning animation are in ./assets/clean/.

For the most part, files are named after their frame number, starting at 0. The only exception to this are the idle animation frames, which have an age descriptor prefix before the frame number. For example, frame #7 for an adult triops’ idle animation would be adult-7.gif. The full path would be ./assets/idle/adult-7.gif

Image Format and Style:

All animation image files are static GIF images with a resolution of 300x300. All the images have a transparent background, with the green-ish underlying color being generated internally by the program itself.

Stylistically, the graphics are meant to give the impression that the user is viewing an old low-resolution monochrome LCD display, like what one would find on handheld electronics in the mid-1990's. To that end, the images are drawn in large pixels which imply a 30x30 resolution (as noted above, the true resolution is 300x300). The primary color used is black, with gray used very sparingly, to imply a monochrome or gray-scale display.