# timeSinceFire

*Steve Cummming*

*21 October 2016*

## Overview

Yet Another Age Map Maintainer. This one is peculiar to the LandWeb application.

AgeMap is incremented without bound on all flammable cells; Cells identified as having been burned in the curent year are set to 0.

Any statistics on age structure are to be calculated here, but none are yet implemented. . . because with the current fire model they would be pretty boring.

## Events

### Init

The `Init` event creates the RasterLayer `rstTimeSinceFire`. To do this, it rasterizes the template vegetation map `LCC05` using the `FireReturnInterval` field of the SpatialPoygonDataFrame `shpStudyRegion`. This procedure retains the `NA`s which mask the actual study region within the template bounding rectangle.

Then, the RasterLayer `rstFlammable` is used to mask out areas of open water, rock, etc which can't burn and thus for which TimeSinceFire is not applicable. These become `NA`s in `rstTimeSinceFire`.

The results is that all flammable cells within each polygon in the shapefile are set to the fire return interval specified for that polygon / ecoregion. Under the basic van Wagner model being implemented, this is the expected landscape mean age. The ecoregion age structure will equilibrate to the exponential distribution within a few multiples of the return interval.

No colour ramp or legend is created for this layer.

In the short term, this initial uniform age distribution will result in very high proportions of cells with tsf's greater than the return interval. If this becomes a problem, one could initialise to the regional median age. This can be done by multiplying the FireReturnInterval by $log(2)$ and then rounding; or some other lower quantile could be chosen: see the [wikipedia page] ](https://en.wikipedia.org/wiki/Exponential%60distribution) for the general quantile function.

Alternatively, a random exponential age structure could be generated for each ecoregion from the current `rstTimeSinceFire`, roughly as follows. See the wiki page for details and possible alternative methods.

```
U_ <- runif(ncell(rstTimeSinceFire))
T_ <- (-log(U_)) * rstTimeSinceFire[]
rstTimeSinceFire[] <- round(T_)
```

### Plotting

A bare call to `Plot(sim$rstTimeSinceFire)`. If you really want to see this, you'll have to live the wutomated colour scheme and legend, or hack `Init` to your satisfaction.

### Saving

Nothing is saved at present.

### Age

This is the main event. `rstFlammable` is incremented by one. Then burned cells, as specified in the inpit vector `burnLoci` are set to age 0.

## Data dependencies

### Input data

This module reads no input data directly.

It requires three inputs for the `Init` event, as noted, and a vector of indices of cells burned for the `age` event.

The input `LCC05` and `shpStudyRegion` could be read in from disk, and `rstFlammable` derived from them, but why would you want to? That's what module `initBaseMaps` and `fireData`Prep‘ do.

### Output data

The only output produced is the RasterLayer it maintains, `rstTimeSinceFire`.

## Links to other modules

This module depends on the existence of module `fireNull`, or simular disturbance module to provide a vector of indices of disturbed cell to reset to `tsf=0`.

Could be input to any module tracking an age dependent event or attribute.