Construct a building

This test verifies that a building has been placed on the selected tile.

Spending to build a building

Every building costs a different amount of resources. This checks that the correct number of resources was subtracted from a civilization’s stockpile.

Movement

Worker units can build buildings. This checks that the worker can move properly.

Adding resources on turn

On moving to the next turn, the civilization’s stockpile should increase relative to the resources acquired and developed.

Happiness

If the player’s food is negative, the player will eventually lose. This test verifies that the happiness value decreases correctly when the food is negative.

Growing a city- both sizes

A city can have 3 population sizes. This checks that on the proper population, the city grows to the next size.

Correct building placement

This test ensures that a building can only be built on its resource type, i.e. a sawmill can only be built on forest.

Research unlocks

This game has a rudimentary tech tree. This verifies that a certain technology is enabled after the number of turns is elapsed.

Losing conditions- time expired, <0 happiness

The player can lose if the civilization has negative food for too long, or if the time limit is reached. This checks both conditions.

Win conditions- # resources, amt of land

Currently, the player wins if they have a certain number of each resource, or a certain amount of land controlled on the board. This test checks both conditions. Also, this may be expanded if more victory conditions are created.

Random checkers

Part of this program randomly generates a map with the different terrain types. This checks that we have at least some of every terrain type on the map.

Loading predefined map

We will provide the program with a .csv that has a variety of terrain types defined already. This lets us test that different terrains are correctly being recognized in the map.

Resource generation

This checks both predefined resource generation by .csv, as well as random generation for the random map.