What do you select based on?

The imagined value of expanding that certain part of the game tree.

Which includes the amount of times that leaf node has been visited.

This is calculated by the UCB1 equation.

So the system just calculates the UCB1 values of each leaf nodes and then chooses the highest of the values to explore next.

How many times do you roll out?

You roll out once, before finding the child nodes of the leaf node.

Then you select again using UCB1.

Eventually after selecting, you come back to the child nodes and then roll them out and back propagate the values of the leaf node and other higher nodes.

This continues to give better values for the states.

The values of the state nodes, then, gets better the more iterations are allowed.

When does it stop?

It keeps going until a specified amount of time runs out (say 3 seconds) or when a specified number of iterations of the algorithm are completed.

Once it has stopped, the node with the best value is chosen.

The balancing act here is between the expansion (exploring new nodes) and the exploitation (seeing how good this node is by rolling out random simulations). (by my current understanding of the definitions of these 2).

So the rolling out per child or leaf node is just 1 but the leaf node

What happens to the simulation tree when the algorithm stops and the actual move is selected?

It is scrapped, and a new tree is made where the first initial node is the new real game state.

Since the new real game state would want as much depth of search as the previous game state before stopping (based on time etc..), the old simulation data is scrapped.

Some different approaches and slight changes to the algorithm, however, have tried to make use of it. I should research this next.

Do you roll out the all the child nodes of the leaf node or just 1 child node? Which is better and why and in what situations (if so)?

Things to research more/ I am unsure about

Provides Algorithm aswell (Not completely exactly what Ill be doing probably):

<https://www.youtube.com/watch?v=UXW2yZndl7U>

<https://www.youtube.com/watch?v=onBYsen2_eA>

<https://www.youtube.com/results?search_query=monte+carlo+tree+search+python>

<https://en.wikipedia.org/wiki/Monte_Carlo_tree_search>