VS-Saturn

A Proposal by Noah Mezher

The most valuable resource one has is time. Time is money, and there is no way to acquire more time than the time one is destined to have. How one chooses to spend their time depends on the person in question, but if productivity is the goal, then one must choose to spend their time wisely. To manage how much time I spend doing productive tasks, I often employ the Pomodoro technique of time management. In short, this technique follows a few steps:

1. Set a timer for 25 minutes and work on a specific task for that amount of time.
2. After 25 minutes has elapsed, put a check-mark on a piece of paper.
   a. If you have recorded fewer than four check-marks, take a 3-5 minute break.
   b. If you reach your fourth check-mark, take a 20-30 minute break.
3. After break time has elapsed, set another timer for 25 minutes and repeat.[1]

This simple technique increases focus on the task at hand and allows down-time for your brain to rest and re-focus. This solves the problem of staring at work for hours at a time and reduces the risk of mistakes due to mental fatigue. While the creator of this technique encourages a pencil and paper methodology with a physical timer, I often find myself in front of my PC while working so I use an online timer and notepad program to track my pomodoros. As a programmer, I often also find myself using an IDE (namely, Visual Studio Code) to develop, so my idea aims to streamline the number of steps it takes to track my time.
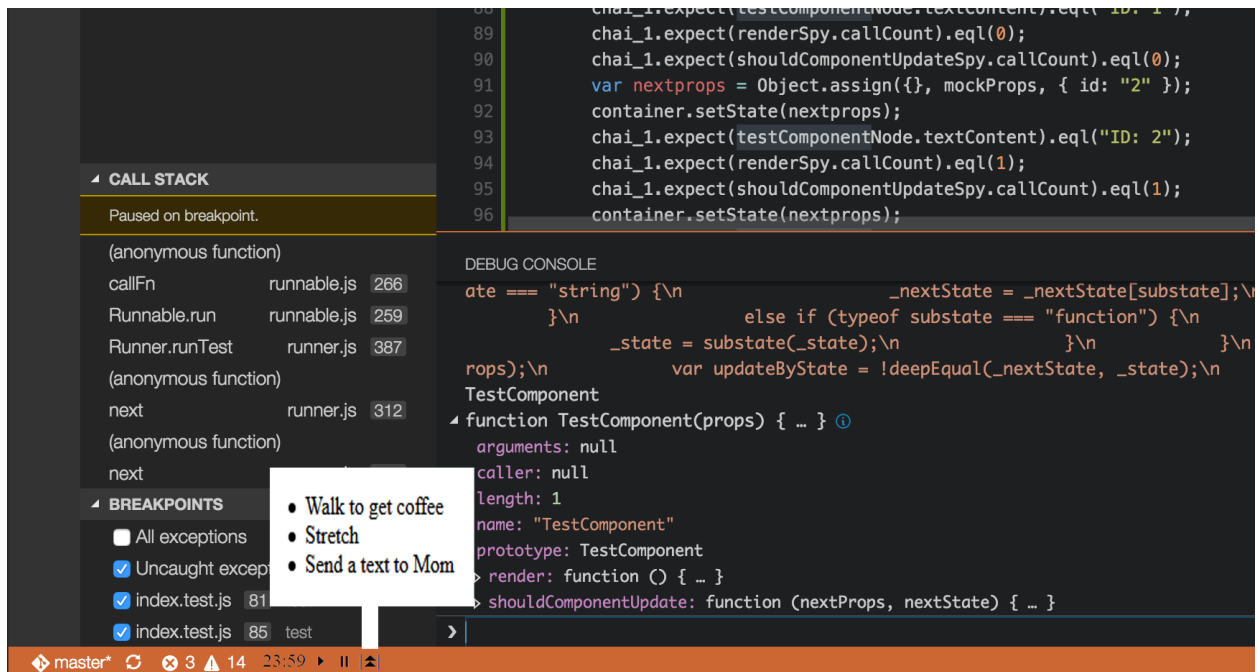
If the reader has ever used Visual Studio Code (abbr. to VSCode) then they would know that it supports the creation of add-ons. Add-ons can be used to change the appearance of the IDE, add functionality not built-in such as PDF viewing or Intellisense, and is a great way to customize VSCode to fit your need. My proposal is to build a pomodoro timer for VSCode that will provide task tracking functionality not offered by any other add-on currently on the marketplace. Other pomodoro timers have been developed, but the following features that I intend to implement make VS-Saturn the optimal choice:

- Snooze button for delaying the end of a work or rest cycle.
- Task checklist for the 25-minute timer and rest timer so you can track progress and set goals.
- Non-intrusive notification for when a timer is up (changes VSCode's theme from light to dark and vice versa so that you are not inclined to look at your IDE).

All VSCode applications are developed with JavaScript or TypeScript and VS-Saturn is no exception. Since VS-Saturn is an add-on, it will be contained within the VSCode application and will not require compilation of an executable. All development and testing can be done within VSCode, and anything outside of that platform is out of the scope of this application. This add-on will be built using the VSCode extension API.[2]

---

[1] https://francescocirillo.com/pages/pomodoro-technique
[2] https://code.visualstudio.com/api/get-started/your-first-extension

```
89              chai_1.expect(renderSpy.callCount).eql(0);
90              chai_1.expect(shouldComponentUpdateSpy.callCount).eql(0);
91              var nextprops = Object.assign({}, mockProps, { id: "2" });
92              container.setState(nextprops);
93              chai_1.expect(testComponentNode.textContent).eql("ID: 2");
94              chai_1.expect(renderSpy.callCount).eql(1);
95              chai_1.expect(shouldComponentUpdateSpy.callCount).eql(1);
96              container.setState(nextprops);
```

▲ CALL STACK

Paused on breakpoint.

(anonymous function)
callFn   runnable.js   266
Runnable.run   runnable.js   259
Runner.runTest   runner.js   387
(anonymous function)
next   runner.js   312
(anonymous function)
next

▲ BREAKPOINTS

■ All exceptions
☑ Uncaught except
☑ index.test.js   81
☑ index.test.js   85   test

• Walk to get coffee
• Stretch
• Send a text to Mom

DEBUG CONSOLE

```
ate === "string") {\n                          _nextState = _nextState[substate];\
        }\n                    else if (typeof substate === "function") {\n
                _state = substate(_state);\n                    }\n              }\n
rops);\n            var updateByState = !deepEqual(_nextState, _state);\n
TestComponent
▲ function TestComponent(props) { … } ⓘ
  arguments: null
  caller: null
  length: 1
  name: "TestComponent"
  prototype: TestComponent
  ▸ render: function () { … }
  ▸ shouldComponentUpdate: function (nextProps, nextState) { … }
```

master* ↻ ⊗ 3 ⚠ 14 23:59 ▶ ‖ ▲

Mock-up of VS-Saturn, ability to see time remaining, pause, play, and check task list. The task list can be collapsed to be non-intrusive.