|  |
| --- |
| const minDelay = 60; |
|  | const maxDelay = 60; | |
|  |  | |
|  |  | |
|  |  | |
|  | const keyOverrides = { | |
|  | [String.fromCharCode(160)]: ' ' // convert hardspace to normal space | |
|  | }; | |
|  |  | |
|  | function getTargetCharacters() { | |
|  | const els = Array.from(document.querySelectorAll('.token span.token\_unit')); | |
|  | const chrs = els | |
|  | .map(el => { | |
|  | // get letter to type from each letter DOM element | |
|  | if (el.firstChild?.classList?.contains('\_enter')) { | |
|  | // special case: ENTER | |
|  | return '\n'; | |
|  | } | |
|  | let text = el.textContent[0]; | |
|  | return text; | |
|  | }) | |
|  | .map(c => keyOverrides.hasOwnProperty(c) ? keyOverrides[c] : c); // convert special characters | |
|  | return chrs; | |
|  | } | |
|  |  | |
|  | function recordKey(chr) { | |
|  | // send it straight to the internal API | |
|  | window.core.record\_keydown\_time(chr); | |
|  | } | |
|  |  | |
|  | function sleep(ms) { | |
|  | return new Promise(r => setTimeout(r, ms)); | |
|  | } | |
|  |  | |
|  | async function autoPlay(finish) { | |
|  | const chrs = getTargetCharacters(); | |
|  | for (let i = 0; i < chrs.length - (!finish); ++i) { | |
|  | const c = chrs[i]; | |
|  | recordKey(c); | |
|  | //console.log(c, c.charCodeAt()); | |
|  | await sleep(Math.random() \* (maxDelay - minDelay) + minDelay); | |
|  | } | |
|  | } | |
|  |  | |
|  | // ############################################################################################################ | |
|  | // old utilities | |
|  | // ############################################################################################################ | |
|  |  | |
|  |  | |
|  | // /\*\* | |
|  | // \* @see https://stackoverflow.com/questions/8942678/keyboardevent-in-chrome-keycode-is-0/12522752#12522752 | |
|  | // \*/ | |
|  | // function simulateKey(chr, el) { | |
|  | // \_simulateKey(chr, 'keydown', el); | |
|  | // \_simulateKey(chr, 'keypress', el); | |
|  | // } | |
|  | // function \_simulateKey(chr, type, el) { | |
|  | // var eventObj = document.createEventObject ? | |
|  | // document.createEventObject() : document.createEvent("Events"); | |
|  |  | |
|  | // if (eventObj.initEvent) { | |
|  | // eventObj.initEvent(type || "keydown", true, true); | |
|  | // } | |
|  |  | |
|  | // let keyCode = chr.charCodeAt(0); | |
|  |  | |
|  | // eventObj.key = chr[0]; | |
|  | // eventObj.keyCode = keyCode; | |
|  | // eventObj.which = keyCode; | |
|  | // eventObj.isTrusted = true; | |
|  |  | |
|  | // el = el || document.body; | |
|  |  | |
|  | // // console.log(keyCode, eventObj); | |
|  |  | |
|  | // el.dispatchEvent ? el.dispatchEvent(eventObj) : el.fireEvent("onkeydown", eventObj); | |
|  | // } | |
|  |  | |
|  | // document.addEventListener("keydown", function (e) { | |
|  | // console.log('down', e); | |
|  | // }); | |
|  | // document.addEventListener("keypress", function (e) { | |
|  | // console.log('press', e); | |
|  | // }); | |
|  | //$($('.menu-btn')[0].parentNode).prepend('<button onclick=\'simulateKeyPress("c")\'>sim</button>'); | |
|  | // simulateKey('a', $('input')[0]); | |
|  |  | |
|  |  | |
|  |  | |
|  | // ############################################################################################################ | |
|  | // go! | |
|  | // ############################################################################################################ | |
|  |  | |
|  | autoPlay(true); | |
|  | | |
|  | | |
|  | | |