

Note from the Elevated School: this is a review on a former student's draft. They gave us explicit permission to use their work, and we have edited key details for privacy. Note that we will NEVER share your writing or edits publicly without permission.

R has previously uploaded this essay to our portal 5 time(s). For a link to their previous draft (Draft #1), which was rated a 7-7.5, see here: [\[Omitted\]](#)

Student Comments: Mix of both - nice polish + any content changes you may suggest

Oh my gosh, this essay *rocks!* Congratulations on being a winner, R.

I love that you totally found your way to the heart of this CAPS, including everything that once felt like it was jostling for attention without making the essay feel jumbled or the pacing too fast.

I'm with both Jeff and Kevin in thinking this one is up there in the 7.5-8/9 stratosphere!

I actually think that what's holding it back isn't necessarily a lack of more personality moments and humor. It's more about finding room to let some of your sentences unfurl with a bit more length. Strategic fragments and bits of staccato writing are strong, but they do best when they're deployed alongside longer, more graceful ones, so I tried to make room for those!

To make that room, the big thing I'd recommend is removing the epigraph from Harari up top, keeping it limited to the paraphrased version in your takeaway. The guy has thousands of pages in which to share his thoughts; let's not give him too many of your 650 words! I'd also cut your last two sentences, which don't feel like as smooth of a mic-drop moment as your previous sentence (see the clean copy for the full effect).

I love the way you've expanded the section with L, too, giving him the time and attention to share his fears and take steps towards overcoming them.

This is a well-paced, well-written essay that takes time to chew on its ideas without ever losing sight of the central character: you. It's genuinely interesting to read, beyond its function as a CAPS, but it really is a great one!

Amazing work, R. You're nailing this!

-Natalee

Share an essay on any topic of your choice. It can be one you've already written, one that responds to a different prompt, or one of your own design. (650 words)

"Especially as technology gets better at understanding humans, you might increasingly find yourself serving it, instead of it serving you." - Yuval Noah Harari, 21 Lessons for the 21st Century

Technology promised to democratize creation. Then it stole my friend's music. S's garage-rock demo leaked online in ninth grade. But what terrified me wasn't the betrayal. It was watching my friends share it without thinking. The platform had made sharing so automatic, so validated by trending algorithms, that the pause - the moment where you'd ask "should I?" - just vanished. When I called it out, they got it immediately. "Oh shoot, you're right. I didn't even think about it."

That's what scared me. Not that they couldn't think. That the system was designed so they didn't have to.

That fear followed me into our game-dev meetings. When J suggested open-sourcing our game, I refused immediately. If platforms trained people not to think, shouldn't we protect what we built?

He held up a hand. "R, didn't you say open-source builds community?"

I opened my mouth. Closed it. I realized that by keeping the code secret, I was just asking people to be passive players - giving them a toy to play with rather than tools to build something of their own. I'd become what I feared: making decisions for people instead of trusting them to think.

That evening, I created a Discord channel: #the-debate. My first message: How do we protect our vision while empowering our community? The next day, our whiteboard became a battlefield. I argued open doors invited chaos; M countered that locked ones shut out real devotees.

I uncapped a dry-erase marker and squeaked out a compromise: open-source the core mechanics, but curate player submissions. Players could build their own levels and characters; we'd be curators with standards. Not perfect, but honest about the power we held—and more importantly, trusting our community to think alongside us.

This August, I brought this lesson to the Boys & Girls Club, where I taught game design to L - eight years old, same red shirt every week.

When he asked if he could use Mario in his project, I didn't shut him down. I could have said no. Easy answer. But that's just another form of thinking for him.

"What do you love about Mario?" I asked.

"He jumps super high! And he's brave!"

Twenty minutes later, he came back beaming, holding up a pixel drawing of an anthropomorphic lizard with a jetpack. "His name is Spike!" He was bouncing in his chair, proud of something he'd created by thinking it through.

But it wasn't enough. The next week, L wanted to add a villain - and pulled up a dragon from a free asset pack half the kids were using.

So we spent thirty minutes talking - not about dragons, but about what scared him. The stove that burned his hand last summer. How he wishes he could be brave around things that frightened him.

The next week: a walking furnace named Burnie, with eyes he drew himself. The asset pack couldn't give him that. Thirty minutes and a question could.

Harari warned we'd serve technology instead of it serving us. He's right about the infrastructure - I can't dismantle billion-dollar platforms designed to make thinking optional.

But I'm realizing my role isn't to build walls against these systems. It's to open the right gates - to ask "What's your version of Spike?" and create spaces where thought is required, not bypassed.

Every weekend, I'm teaching L something tools can't replicate: that the thirty minutes before he draws matters more than the drawing. AI can generate a dragon in seconds. It can't sit with him while he figures out why fire scares him. In a world of instant output, human work isn't production - it's the process that makes production mean something.

In college, I'll bring this spirit to every hackathon, robotics build night, and group project - fostering environments where inspiration requires thought, where autonomy is practiced, not outsourced. If original thought dies when we stop asking questions, then engineering moments that demand them become the work itself.

That's not resignation. That's where thinking begins.

Clean copy (642 words)

Technology promised to democratize creation. Then it stole my friend's music.

When S's garage-rock demo leaked online in ninth grade, what terrified me wasn't the betrayal. It was watching my friends spread it without thinking. The platform had made sharing so automatic that the pause—the "*should I?*"—had just vanished. When I called it out, they understood immediately: "I didn't even think about it."

That's what scared me. It wasn't that they couldn't think. The system had just been designed so they wouldn't have to.

That fear followed me into our game-dev meetings. When J suggested open-sourcing our game, I refused immediately. If platforms trained people not to think, shouldn't we protect what we built?

He held up a hand. "R, didn't you say open-source builds community?"

I opened my mouth. Closed it. I realized that by keeping the code secret, I was just asking people to be passive players—giving them a toy to play with rather than tools to build something of their own. I was emulating the system I feared: making decisions for people instead of trusting them to think.

That evening, I created a Discord channel: #the-debate. My first message: How do we protect our vision while empowering our community? The next day, our whiteboard became a battlefield. I argued open doors invited chaos; M countered that locked ones shut out real devotees.

I uncapped a dry-erase marker and squeaked out a compromise: open-source the core mechanics, but curate player submissions. Players could build their own levels and characters; we'd be curators with standards. Not perfect, but honest about the power we held. Most importantly, we'd trust our community to think alongside us.

This August, I brought this lesson to the Boys & Girls Club, where I taught game design to L—eight years old, same red shirt every week.

When he asked if he could use Mario in his project, I didn't shut him down. I could have said no. But that would have been the easy answer, just another way to think for him.

"What do you love about Mario?" I asked.

"He jumps super high! And he's brave!"

Twenty minutes later, he came back beaming, holding up a pixel drawing of an anthropomorphic lizard with a jetpack. "His name is Spike!" He was bouncing in his chair, proud of something he'd created by thinking it through.

But it wasn't enough. The next week, L wanted to add a villain, so he chose a dragon from a free asset pack half the kids were using.

So we spent thirty minutes talking—not about dragons, but about what scared him, like the stove that burned his hand last summer. About how he wished he could be brave around things that frightened him.

The next week: a walking furnace named Burnie, with eyes he drew himself. The asset pack couldn't give him that. Thirty minutes and a question could.

Yuval Noah Harari has warned that we'll find ourselves serving technology instead of it serving us. He's right about the infrastructure; I can't dismantle billion-dollar platforms designed to make thinking optional.

But I'm realizing my role isn't to build walls against these systems. It's to open the right gates—to ask "What's your version of Spike?" and create spaces where thought is required, not bypassed.

Every weekend, I'm teaching L something tools can't replicate: that the thirty minutes before he draws matter more than the drawing itself. AI can generate a dragon in seconds. It can't sit with him while he figures out why fire scares him. In a world of instant output, human processes give our creations meaning.

In college, I'll bring this spirit to every hackathon, robotics build night, and group project, fostering environments where inspiration requires thought and autonomy is practiced, not outsourced. If original thought dies when we stop asking questions, then our work is to engineer moments that demand them.