

Dave The Computer Guy

Andrija Milovac, Roko Burilo

23.9.2021

Contents

1	Product vision	1
1.1	Simple explanation	1
1.2	Story intro	1
1.3	Graphics	2
1.4	Gameplay	3
2	Product market	4
3	Product implementation	4

1 Product vision

1.1 Simple explanation

Dave The Computer Guy is a webGL game whos goal is to teach you how to build a computer from scratch out of logic gates step by step.

1.2 Story intro

At the center of the game is fresh computer science graduate Dave. Dave got hired at an small digital electronic components manufacturer company called Completely-Digital.

He will work in an office with his senior hardware developer named John Dodi and their boss Dexter Crawford.

Daves day to day will consist of syncing with his boss about the work that needs to be done and doing the required work which is mostly creating new electronic components and manufacturing them.

Whenever Dave gets stuck on a task or has no idea where to even start he can always ask John Dodi for help as he is a senior developer with decades of experience and wisdom.

Dave is a very motivated young person and besides working hard at his new job he decided to practice and put to use everything he learns at his job by building a computer at home from scratch.

The reason he wants to do this is that he was originally a software (computer science graduate) guy and only recently got into the world of hardware and he wishes to connect his software knowledge with his newly found hardware knowledge so that he can have a deep understanding of computers.

1.3 Graphics

The game will use WebGL to display its graphics.

The graphics are 3D and voxel models will be used.

An isometric camera will be used to view the scenes.

There will be 3 main scenes in which the player will play:

1. The office - Dave's workplace



2. Home - Dave's home where he spends time learning and creating.



3. Electronics shop - Dave buys circuit parts here from a salesman named Adem Shady



Each of the mentioned 3D scenes will contain clickable elements that will open additional UI elements where the gameplay will actually happen like for example the circuit editor window.

1.4 Gameplay

The goal of the game is too build your own computer at home that can execute instructions of a predefined format (we made the instructions predefined so that we can validate the validity of the made computer by executing predefined programs).

The game will follow a linear path which is of the following form.

1. Get an assignment from your boss
2. Solve the assignemt (this include producing various components and buying the necessary parts to make the components)
3. Get money for solving the assignment and unlock new components

4. Use those components to build other more complex components and systems at home
5. Once you progressed and unlocked every type of component you will get the final mission of building a computer
6. Once you build a computer that satisfies the given conditions (checked via tests made by the developers) you have officially completed the main story of the game
7. After completing the game you unlock access to the black market where you can buy components made by other players and make anything you can imagine

2 Product market

The audience for this product are electronics and computer enthusiasts as well as college students.

The product could be used by universities to teach computer architecture in a fun way engaging and rewarding way.

Profit will be earned by leveraging the game economy and reward ads. (Up to discussion, detailed description needed).

3 Product implementation

The core simulator will be written in Rust (WASM) or in TS (ran in a separate worker).

The frontend of this application will be made using the following technologies.

- Svelte (Sveltekit) as the frontend framework
- Threlte (Three.js) as the WebGL framework (library)
- TailwindCSS for styling
- Google adsense ads

The backend of this application will be made using the following technologies.

- Vert.x as the backend web framework (toolkit more precisely)
- Postgres as the database

The application will be hosted on a VPS (most likely oracle on an oracle free VPS)