Portfolio

Sr Software engineer

Levan Natsvlishvili

Email: LevanNatsvli@gmail.com

Linkedin: https://linkedin.com/in/levnac

Website: https://levan-natsvlishvili.com

Github: https://github.com/Levannatsvlishvili

Frontend: React, Next

Backend: NodeJS, Nest, PostgreSQL

AI tools: Claude, Cursor, ComfyUI, Flux, SDXL, Ollama

WebGL: Three.js, Pixi.js

CSS libraries: Tailwind, Css-modules, JSS, Sass, Styled-components

Animation: Gsap, Framer-motion

PROJECTS

AI

2D Game AI Asset pipeline

Tech stack: ComfyUI, FLUX, Wan 2.2, Lora, Openpose ControlNet

Built pipeline for 2D Game asset generator. Idea was to create a tool unlike other GenAI for images, where using it is cost effective and you don't have rely on Token usage and can be used on local non-server hardware. Pipeline includes generating characters, modification and animating sprites.

- Built ComfyUI pipeline on SDXL& Flux with custom LoRAs for a consistent art style
- Trained custom Loras for pixelated style and consistency.
- OpenPose ControlNet for controlling character poses, IP-Adapter for keeping characters consistent across images
- Wan 2.2 image-to-video for sprite animations: walk cycles and attacks, etc.
- Integrated multiple LLM into MCP (Claude, Gemini, Cursor)

Autonomous ATS Job Radar

Tech stack: Google Gemini 1.5 Flash, Python, NextJS

Built an autonomous job surveillance radar and AI application copilot. The goal was to eliminate manual job searching by directly crawling company ATS endpoints, evaluating opportunities with Gemini 1.5 Flash, and automating personalized application outreach.

- Built a monitoring tool for company job boards (Greenhouse, Ashby, Lever, Djinni) that searches and fetches filtered application.
- Integrated Google Gemini 1.5 Flash to analyze job descriptions and assign qualification scores (1–5 points) based on qualification provided.
- Integrated instant alert notifications.Integrated instant alert notifications.

WebGL

Sweeper

Tech stack: Pixi.js

Inspired from Candy Crush, built as one HTML file for mobile ad networks. size: **600.3 KB**

- Single-file Vite output, zero runtime network requests, MRAID viewability + pause/resume
- Procedural PIXI.Graphics candies, no image assets

- Under ~7% of a typical 2.5 MB ad-size budget

Wall Dodge

Tech stack: Pixi.js

A 13-wall lane-dodge playable, built as one HTML file for mobile ad networks.

- Single-file Vite output, zero runtime network requests, MRAID viewability + pause/resume
- Procedural PIXI.Graphics walls and ball — no image assets
- Shield shatter + slow-motion on first crash; second crash goes to CTA

Garden-RPG

Tech stack: Three.js, Pixi.js, Vite, lil-gui, Blender

Description: A browser-based 3D farming and survival RPG with a day/night cycle. During the day you build a garden: you place a well, plant tomatoes, cucumbers and vines on a grid, harvest them for gold, and spend it in the shop on upgrades and health potions. At night the lights go down, your character lights a lantern, and waves of monsters attack, getting stronger each night. It runs on desktop (keyboard) and mobile (virtual joystick and attack button).

Responsibilities/Achievements:

- Designed the full gameplay loop: build and plant by day, fight by night, and plants ripen for harvest after you survive a night
- Built a grid-based placement tool using raycasting, with cell highlighting, blocked and occupied cell markers, and collision so objects can't overlap the character or each other
- Built the day/night cycle by changing lighting at runtime: ambient and directional light fade, a lantern light turns on, and scene sprites get a night tint
- Built real-time combat: nearest-target melee attacks, cooldowns, hit-flash and knockback feedback, and health bars above monsters
- Wrote monster AI that chases the player and avoids obstacles, with night waves that grow up to a peak and then reset with tougher, harder-hitting monsters
- Animated the character and monsters (idle, walk, slash) with Three.js AnimationMixer, plus a camera that follows the player smoothly regardless of frame rate
- Built the whole 2D HUD as a Pixi.js layer on top of the Three.js scene: top bar (health, damage, gold), shop, tutorial tips with glowing buttons, game-over screen, and touch controls
- Wrote a step-by-step tutorial that gates progress on conditions (place the well, plant, survive, harvest, upgrade)
- Matched every shadow to one light direction: real shadow maps for 3D models, plus soft contact shadows drawn at runtime for 2D sprites
- Cut asset size from about 65 MB to 2–3 MB by optimizing models and textures in Blender and switching to WebP
- Organized the codebase into clear modules for scene, gameplay, UI, state and config
- Built a loading screen with a progress percentage, and capped the frame rate for stable performance

Cozy-room

Tech stack: Three.js, Vite, GSAP, lil-gui, stats.js, blender, MeshyAI

Description: An interactive 3D scene, with theme “cozy room”, with ability to interact with models (TV, lights, fireplace/candles), atmospheric weather (rain + thunder), and lightning effect driven by window lightning.

Responsibilities/Achievements:

- Architected a maintainable, scalable structure with clear boundaries between rendering, assets, interactions, and animations
- Created candle and fireplace animations with video, blending it to make animation realistic and save on performance
- Used AI tools or free assets to import 3D assets, reduced size and polished it in blender for performance
- Created Lightning effect in room with ambient light that was depended on window animation.
- Fireplace light flickering and shadows
- Created interactive animations(TV, Fireplace, Candle, Sounds, Bulb)
- Build loading screen with loading percentage
- Delivered stable FPS while limiting frame rates for non-critical animations

---

Halloween

Tech stack: Three.js, Vite, lil-gui, blender

Description: 3D Halloween scene built with Three.js: a haunted house with textured environment, animated ghosts, graves, trees, pumpkin, and dynamic day sky. Users explore the scene with orbit controls

Responsibilities:

- Animate 3 ghosts walking in scene and realistic see through look
- lighting and shadows

Lucky Slots

Tech stack: PixiJS, Vite

Description: Lucky Slots is a browser-based casino-style slot game built with PixiJS. It features a 5×3 reel grid, balance/bet/win UI, and win celebration effects.

Responsibilities:

- Design and implement the slot machine game loop, reel spin logic, and win evaluation across multiple paylines
- Build a custom particle system for win celebrations (gravity, decay, rotation) using PixiJS Graphics
- Create UI components (balance, bet, win panels, spin/bet controls) with PixiJS containers and graphics
- Structure the codebase with modular classes (SlotMachine, Reel, UI, ParticleSystem, config) for maintainability

---

Vampire survivors

Tech stack: PixiJS, Vite

Description: Browser-based roguelike survival game inspired by Vampire Survivors. all visuals are procedurally generated at runtime using the PixiJS Graphics API. Features automatic weapons, XP/level-up system, multiple enemy types, particle effects, and a retro pixel-art aesthetic.

React/Next

Nbrown

Tech stack: Next, Typescript, Redux, Next-Auth0 / OpenID, Jest, Playwright, Micro-frontends, Storybook, A11y

Description: Nbrown project, enterprise microfrontend serving millions of users primarily in UK, with at least 10 brands.

Responsibilities:

- Design, develop, and integrate a Micro-frontend into global enterprise MFE ecosystem
- Implement secure authenticated pages, redirections for unauthorized users.
- Develop pages with high accessibility standards using WCAG /ARIA,
- Keep up unit test coverage across project with SONARQube
- Implement E2E tests
- Review Code

Achievements

- Implemented fully accessible app, meeting high standards.
- Reached minimum of 100 on Lighthouse metrics on Performance, Accessibility and Best practices (SEO was not of importance on the MFE)
- Implemented a fraud-prevention solution, reducing risk exposure and enhancing platform security.

Vip-Pay

Skills: NextJS, TailwindCSS

Vip-pay is a payment gateway for Georgia, Primarily focused on providing payment solutions for businesses

- Built fully responsive web application
- Optimized website's performance to reach minimum of 96+ score on Google's Lighthouse
- Implemented SEO best practices, including Server Side Rendering (SSR) and semantic markup, to enhance search engine visibility and ranking
- Built dedicated admin panel for managing content, statistics, and other data

---

Saba - E-book Store

Skills: Next.JS, Typescript, TailwindCSS

Saba is one of leading Electronic book stores in Georgia, with more than a decade old history.I modernized its former old website using the latest technologies..

● Transposed Figma designs into responsive web layouts utilizing Tailwind CSS, while ensuring smooth animations with Framer-motion

● Implemented SEO best practices, including Server Side Rendering (SSR) and semantic markup, to enhance search engine visibility and ranking

● Implemented a dynamic feature enabling users to revert to previous version of the webpage from the latest iteration

● Integrated Google, Facebook, and Email authentication mechanisms

● Developed comprehensive unit tests using Jest and react-testing-library to ensure application reliability and functionality

● Created protected routes specifically for authenticated users, safeguarding sensitive data and features ● Designed interactive statistic charts to visualize user reading time and potential award metrics

---

Archi - Real Estate

Skills: ReactJS, Bootstrap, Typescript,Canva

Archi is one of the leading real estate companies in Georgia. Built project that allows users to visually browse through buildings floors and pick a flat and other features

● Employed Solid principles to create easily maintainable, reusable and clean code

● Built app tool Polygon-tool, with which users could add visual representation of a building and show it to client ● Built dynamic template pages so users could create new pages on client side

● Built in app Filemanager tool to keep and manage massive amount of files,images and videos ● Optimized app to achieve a min score of 80 on Lighthouse metrics

---

National Bank of Georgia - Fintech

Skills: ReactJS, Material-UI, Redux

Project of National Bank of Georgia was massive project, and the largest project I've ever worked on. Team consisted of 7 developers and as one of Frontend developers I worked on Admin side of project:

● Collaborated with other Frontend developers to utilize performance and speed optimization, using lazy loading, caching and etc.

● Conducted regular checks for performance speed with tools like Lighthouse & Bundle-analyzer ● Took part in maintenance and updates of project

---

Dataninja - Social Marketing

Skills: ReactJS, TailwindCSS, Material-UI,

DataNinja is marketing automation platform, the next generation online advertising tool, it aims simplify marketing and bring all marketing campaigns together to create, track and optimize everything.

● Converted web design to responsive user interface

● Participated with team to create best UX

● Built component library

● Optimized app for 30% speed performance increase

● Mentored interns

Citrus - Ecommerce

Skills: ReactJS, Javascript, Sass

Citrus is one of leading ecommerce shop on Georgian market, with hundreds of active users everyday**:**

● Collaborated with the design team to create wireframes, mockups, and prototypes

● Converted web design to responsive user interface

● Built component library

● Employed Solid principles to write clean, reusable & maintainable code