Hello Gentlemen.

Once again, thanks for considering me for this position. I prepared this document to demonstrate relevant experience for the job.

1. **Unreal Datasmith to Blender import/export:**
Link to project: https://github.com/botero-dev/bl_datasmith/
User demo:https://www.youtube.com/watch?v=XFuNaFvBMiQ&list=PLm0v_NETI5zz8uq9Scd3OiMyZBWtnXY79&index=3

This plugin started as a way to import (.udatasmith) files from other DCC into blender. After some time the I added support to exporting datasmith files to Unreal.

- Reverse engineered .udatasmith (xml) and .udsmesh (binary) files to import into blender scene tree
- Loaded multiple object data types (meshes, materials, textures, lights, cameras) into blender scene with correct 3d transforms, uv mapping, etc.
- Implemented exporting 3d scenes to .udatasmith format, as much as the format allows. This includes meshes, materials, textures, object animations, cameras, lights, light probes, instanced meshes (foliage).
- Implemented translation of Blender shader nodes to Datasmith material definitions. This involved translating many Blender built-ins to a Unreal shader library to allow 1:1 translation from blender materials
- Implemented plugin build automation for many unreal versions (4.27-5.3) over multiple OS (Win, Mac, Linux)
- Implemented regression validation scripts, where plugin was run against a control asset corpus and generated delta reports.

2. 3d Globe rendering for Battle Road Digital: https://www.youtube.com/watch?v=EwulAW1Dcos

This is a globe scale simulation game. At first stages it was devised to be a kind of simcity game, but military customers were interested in the product so we pivoted towards a warfare simulation platform.

My role in this project was mostly technical direction and graphics stack implementation. We used Godot engine to create this project. I implemented most of the 3d globe rendering:
- Geography relief displacement and light shading
- Color and elevation Tile streaming from geo web services
- Loading street and building data from OpenStreetMap
- piped geoserver data into navmesh system
- implemented drawing of screen-space decorations like path lines, attack ranges, etc.

3. **Level editor for Vultures: Scavenegrs of Death:**
Game: https://www.youtube.com/watch?v=omiG35UIb28
Tools demo: https://www.youtube.com/watch?v=F157jBcNmng&list=PLm0v_NETI5zz8uq9Scd3OiMyZBWtnXY79&index=1

I was contracted for a few months at early production stages for this project. My goal was to provide the maximum amount of value in the shortest amount of time:
- Implemented a Gizmo UI to easily modify level contents
    - Gizmo supported move, rotate scale.
    - Snap to grid/angle. Snap object to surface, rotate object along surface.
- Implemented undo system for transforms, and object creation and deletion
- Implemented asset drawer with ultra fast search by name
- Improved camera navigation (zoom to pointer, orbit around pointer)
- Optimized level load times

4. Godot Engine Contributions:
PRs: https://github.com/godotengine/godot/pulls?q=is%3Apr+author%3Abotero-dev

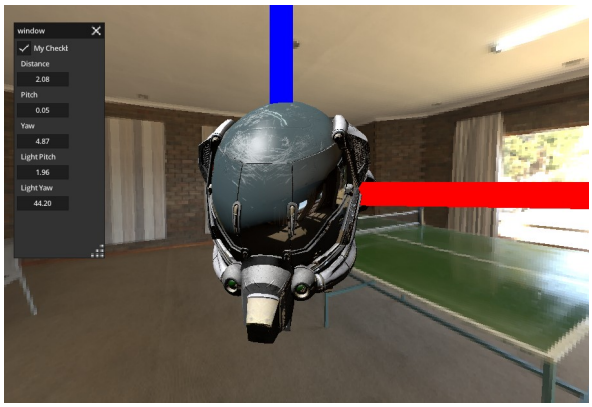
Highlights:
- Feature: Allow per-character Transform2D: https://github.com/godotengine/godot/pull/77819
- Feature: Add Vector3.min/max https://github.com/godotengine/godot/pull/72316
- Bugfix: Fix alignment when Path2D is evaluated https://github.com/godotengine/godot/pull/78378

More on next page.

5. Personal projects:

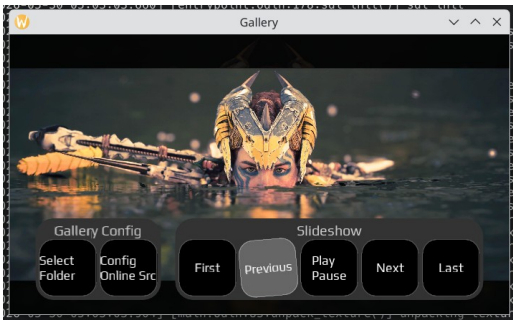
I have some personal projects that were mostly explorations around specific technologies.

3D graphics renderer: https://github.com/botero-dev/ab_sdl
- using SDL_gpu and Odin language
- From-scratch 3d renderer with PBR shader and IBL specular reflections
- Linear colorspace rendering.
- gltf and OpenEXR asset loading



2D graphics renderer: https://github.com/botero-dev/sdl-odin-boilerplate

This is an experimental project where I used a gallery app as testbed to implement a 2d engine. Buttons are not functional (except the last one that switches DPI config),
OLDER BUILD ONLINE: https://labs.botero.dev/gallery/
- using SDL and SDL_Render (HAL for gles2/webgpu)
- cross-platform deployment (Win/Linux, Android, Web)
- UI Rendering with:
    - antialiased lines and rounded corners (desktop only)
    - UI Layout with clay (https://www.nicbarker.com/clay)
    - transform modifiers (check buttons rotation when focused)
    - Keyboard navigation (press down in gallery to focus buttons and then sideways to move focus)
    - font rendering with SDL_ttf
    - HiDPI handling
- Async filesystem API (with emscripten fetch in web)
- image loading
- simple shape drawing (lines, circles)



DXF Renderer:
I wanted to implement a good DXF renderer and editor, as I feel there are no good options for Linux (LibreCAD and QCAD are lacking in some ways)

This is also very early prototype: LIVE DEMO: https://labs.botero.dev/dxf/
(looks better in chrome/edge/vivaldi)
- Load DXF Files
- Draw some entities (lines, polylines, circles, text)
- Use entity color index field for coloring
- simple zoom (scroll wheel) and panning (click)

`

6. Other Game Projects:
Game: Orbit Outlaws https://www.youtube.com/watch?v=YU-JoHlD3X4

This was my first big experience. Its a bit dated now but still relevant
- Implemented the UI, which shows weapons and customizations dynamically from data tables.
- Implemented multiplayer state and input handling in menus and gameplay.
- I implemented the shaders in this project to allow for dynamic character coloring.
- I also reimplemented and optimized the orbital mechanics simulation of projectiles.
- Implemented predictive aiming which also involved optimization of game state data structures.
- Did bugfixing of shaders for mobile platforms.
- Implemented compilation and deployment automations.

Game: Almighty Idiots: https://www.youtube.com/watch?v=1tl7SUqLQC0

- Implemented early stage prototypes, advised UI and gameplay changes
- Implemented rendering strategy.
    - Characters and environment shading with palette textures.
    - Layered render targets for 3d-over-2d effects
- Systems design and implementation, led and advised engineers about implementation approaches.