

Lucas Teixeira

I'm a seasoned Brazilian software engineer familiar with C#, ActionScript, JavaScript, Java and C++. Focused mainly on games, 3D, motion and client-side programs in general. Solid skills on design patterns, software architecture, sockets/network, mathematics and physics.

Experience

Software Engineer at Rogue Snail (2019)

Remote position.

Developed the loot/action game Relic Hunters Legend using Unity3D and C#.

- Gameplay, VFX and UI code;
- Steam API integration;
- Software architecture and planning;
- Used Photon to implement multiplayer;
- Worked on a existing Amazon AWS system (using C#) to build, deploy and upload the game to Steam platform.

Software Engineer at Possible Brasil (2018)

Remote position.

Developed the Siemens Quiz game for mobile platforms using Apache Cordova and HTML/CSS/JavaScript.

- Large code base with years of existence and several past contributors;
- System based on AngularJS;
- Worked on the client side creating new features and game modes;
- Server communication via HTTP requests;
- Deployed for Android and iOS.

Software Engineer at Sirlin Games (2017)

Remote position.

Mainly responsible for the UI and Steam integration of the fighting game Fantasy Strike using Unity3D and C# with the Universal Fighting Engine.

- Responsible for UI and screen navigation flow;
- Developed the training mode of the game;
- Steam API integration;
- Also worked on the matchmaking server (using Java) writing an implementation of the Elo rating system.

Software Engineer at Aquiris Game Studio (2014-2017)

Porto Alegre, Brazil.

During my time at Aquiris I've worked on two projects using Unity3D and C#. The first one was the first person shooter game Ballistic Overkill and the second was the mobile game Looney Tunes World of Mayhem (which I was the leader of the client team).

- Ported the game Ballistic from Facebook to Steam platform;
- Reimplemented the network system using Steam's peer-to-peer networking API;
- Created the dedicated server of the game;
- Maintained the game during the early access stage;
- Client team leader of the Looney Tunes game for mobiles platforms;
- Responsible for sprint planing, designing the core systems and supporting the members of the team;
- Developed two prototype of battle systems similar to JRPG games.

Motion Programmer at Conrad Caine (2010-2014)

Pelotas, Brazil.

Worked mainly with Flash (ActionScript2 and ActionScript3) to develop a wide range of web and motion applications including screen savers, panic boards and e-learning games.

- Developed advergames and e-learning games using ActionScript3;
- Implemented screen savers for Windows using C++ and OpenGL;
- Worked on a high profile project where we created a panic board application using ActionScript3 and JavaScript;
- Server communication using HTTP requests and TCP sockets;
- Developed prototypes using Microsoft Kinect for a movement-based input system.

Flash Developer at Eckert&Caine (2009-2010)

Pelotas, Brazil.

Part of a small team developing a large web system for strategy and management used by companies around the world.

- Developed multimedia components using Flash/ActionScript3 to be used by the JavaScript team;
- Created Flash scripts to be used as fallback in older browsers (like a Video Player, File Upload utility and so on).

Flash Developer at Conrad Caine (2005-2006)

Pelotas, Brazil.

Development of motion websites and enterprise softwares using ActionScript and ActionScript2.

- Worked on a large medical system developing the frontend (using ActionScript2) including UI and server communication;
- Developed a simple prototype of a racing game using J2ME and the M3G library for mobile platforms;
- Used projectors like Zync and mProjector to export Flash as desktop applications.

Multimedia Developer at Gestum (2004-2005)

Pelotas, Brazil.

Junior programmer using ActionScript2 to create e-learning games targeted for Flash Player 7.

- Worked on a isometric game engine written in ActionScript2;
- Implemented the A* algorithm;
- Developed game tools for character creation and level design;
- Packed the game using SCORM specification.

Technical Experience

Published Games

Here's a list of every published game that I've worked on:

- **Relic Hunters Legend:** loot/action game for Steam platform.
https://store.steampowered.com/app/633080/Relic_Hunters_Legend/
- **Siemens Quiz:** question and answer game for mobile platforms.
<https://play.google.com/store/apps/details?id=com.siemens.quizapp>
<https://apps.apple.com/br/app/siemens-quiz/id1243612766>
- **Fantasy Strike:** fighting game for Steam platform.
https://store.steampowered.com/app/390560/Fantasy_Strike/
- **Looney Tunes World of Mayhem:** free-to-play game for mobile platforms.
<https://play.google.com/store/apps/details?id=com.aqupepgames.projectpepe>
<https://apps.apple.com/nz/app/looney-tunes-world-of-mayhem/id1158042933>
- **Ballistic Overkill:** online first person shooter game for Steam platform.
https://store.steampowered.com/app/296300/Ballistic_Overkill

My Personal Project

All Too Human is a real time strategy game that I've been developing over the past years using Unity3D and C#.

The game is set in vast wastelands and it's composed essentially by two systems: units control and city management.

- **TigSource DevLog:** <https://forums.tigsource.com/index.php?topic=65558.0>
- **Download the game vertical slice (released in July/2019):** <https://lucas-ot.itch.io/all-too-human-king-of-the-settlement>

Open Source

as3-msgpack is an implementation of MessagePack specification for ActionScript3. I wrote this library during 2012-2013 because at the time there was no good implementation of that specification for AS3. Nowadays as3-msgpack is still listed in MessagePack frontpage and is eventually used by Haxe developers.

- **as3-msgpack at GitHub:** <https://github.com/loteixeira/as3-msgpack>

Skills & Technologies

Languages: C#, ActionScript3, JavaScript, Java, C++, Python and Lua.

Game/Rendering Engines: Unity3D, Three.js, Away3D, Irrlicht Engine, SDL, Allegro and Flash API.

Shading Languages: CG, GLSL and AGAL.

Libraries/Frameworks: Frictionless, Unity3D Editor API, Steam API, Windows API, OpenGL, OpenAL, Make, WebGL, Node.js, Angular.js, React and Amazon Web Services (EC2, Lambda, S3 and DynamoDB).

Appendix

- LinkedIn Profile: <https://www.linkedin.com/in/lucas-teixeira-aab475bb>
- GitHub Profile: <https://github.com/loteixeira>