Contact

dave@boltcreative.com

www.linkedin.com/in/dave-castelnuovo-1828b3 (LinkedIn)

Top Skills

Unity3D
Virtual Reality
Game Development

Languages

English (Native or Bilingual)
Japanese (Elementary)

Dave Castelnuovo

Owner, Bolt Creative and Consultant
San Francisco, California, United States

Summary

At the start of my career I was thrown into a room with a box of Japanese manuals for the Sega Genesis and told to figure it out and make a cool game. That has been the underlying theme throughout my 27 years in game and interactive product development.

With a unique mix of creative, business and technical talent, I've built many cutting edge products for myself and my clients. I have a knack for seeing creative and technical solutions that can either simplify an otherwise complex project (bringing it on-time and on-schedule) or to push a platform beyond it's normal limits.

On the creative side, I love conceptualizing and evolving designs. Because of this, prototyping is something I am very good at.

On the technical side, I pick up languages and platforms very quickly. It took me 10 hours to learn Objective C, the iOS platform and publish my first app. The first release of my hit game Pocket God took 1 week to create.

I have a very pragmatic mind and can strip away the BS when it comes to getting something done without killing the cool stuff. I'm also really good at coming into existing projects and fixing tough problems.

Some personal highlights:

Created the game Pocket God which went on to sell 8 million units, became a #1 app in the app store for an entire month and spun off 2 apps, a 26 issue comic book, and a toy line. Pocket God has consistently been in Apples yearly list of best games of all time.

World Golf Tour is something I am especially proud of. I started at the prototyping phase and architected a flash golf game that is

probably the most realistic golf game available. The technical details are found below.

Experience

Kinoo
VP of Engineering
November 2019 - Present (7 years)
San Francisco, California, United States

Bolt Creative
Founder
January 2001 - Present (25 years 10 months)
San Francisco Bay Area

We started out as a flash interactive services company and gradually evolved into a product driven company specializing in iOS apps.

This culminated in our hit project Pocket God which sold over 8 million units and spun off toys, comics and 2 other apps.

All of the non-iOS projects below were contracted through Bolt. At the time I handled all sales, project management, recruiting and creative and technical direction as well as being the main technical resource on each sub project. For larger projects we recruit supplemental staff through our deep network of creative and technical talent.

Jaunt XR
Senior Programmer
February 2018 - June 2019 (1 year 5 months)
San Mateo, CA

Maintain and extend the JauntXR Player Engine. Simultaneous support of both native player and Unity plugin on Windows, MacOS, Android, and iOS. Supported Unity versions from 5.6 to 2018.

Responsible for the streaming, HW video decoding, and rendering pipeline.

Modified the ARToolkit library to accept input images from ARKit in order to allow ARToolkit to detect barcode markers while ARkit handles all 6dof tracking.

FS Studio at Google DayDream

Architect, Lead Programmer

December 2016 - February 2018 (1 year 3 months)

Mountain View, CA

Architect and implement Unity Eye Tracking SDK for use by other teams within Google.

Exploration of multi-modal/AR/VR interactions and productivity applications using Unity (concepting and demo creation).

Next gen VR/AR headset prototyping (hardware integration and demos)

Custom VR peripheral prototyping: form factor design, embedded systems coding (Nucleo l476rg), and demo integration.

Created technology to remote control multiple individual Windows apps in a mobile VR/AR environment.

Created a real-time ETC2/LZ4 encoder/decoder to get low latency/high resolution app streaming from Windows to Android over a low bandwidth connection.

Heavy low level customization of the chromium embedded framework so that it could be remoted from Windows to Android.

Lots of work with very complex and fragile hardware prototypes. Some requiring 10-15 different hardware components to function.

Eyefluence

Architect, Lead Programmer

April 2016 - December 2016 (9 months)

Milpitas, CA

Brainstorm, design and implement demos demonstrating new gaze based interaction methods using Unity.

Created a gaze based keyboard using deep learning.

Bring up eye tracking in multiple AR/VR platforms (calibration, troubleshooting, optimization, demo development).

Architect cross platform framework so all legacy single-platform demos could be easily deployed to multiple platforms (Vive, Oculus, Gear VR, Day Dream, custom internal hardware).

Embedded systems integration (DART over serial port and Bluetooth).

Re-architect transport layer that connects gaze tracker (geometric and neural net methods) to client.

Project Tango AR Demo
Programmer
April 2016 - April 2016 (1 month)
San Francisco

Create game concept for augmented reality taking computer vision depth field and creating a 3d space that Lemmings-like characters can move around and interact in.

Whamix, “Bones” VR Demo and AI engine
Architect, Lead Programmer
January 2016 - April 2016 (4 months)
San Francisco Bay Area

Develop VR Demo centered on a young girl that interacts with the player in her backyard. As she shows player around and shares her story, the demo highlights her rich emotional reactions to the player’s interest level in what she is doing.

Architect and develop E-motion AI engine that can simulate body language throughout her performance based on player focus and interest level.

Build Maya to Unity animation pipeline that makes large use of motion blending, IK, and root motion.

Create VR centric interaction mechanisms such as head gesture detection (head shake and nod).

Impending, Hatch Virtual Pet Game
Lead Programmer
May 2014 - April 2016 (2 years)
San Francisco Bay Area

Create virtual pet game with many new feature updates.

Complex AI system that monitors health and emotion of your pet and allows it to react in unique ways to the player's actions.

Unique features like the pedometer feature where you can take your pet for a walk and get rewards if you meet certain activity goals and Peek-a-Boo which uses camera face detection so your pet can tell when you are looking at it.

Capital One
Client Architect, Lead Programmer
June 2015 - November 2015 (6 months)
San Francisco Bay Area

Consulted on two skunk works projects for Capital One.

Develop ideas that leverage game design principals and presentation to encourage responsible financial behavior in customers.

Build lavish 3D environment that represents a customer's financial status with characters and environment pieces that evolve as customers save for goals and pay down debt.

Create rich card customization tool that allows a user to choose between materials and decals, and finally lets them sign their card in–app. Meant for mobile as well as kiosk platforms.

River Studios, 360 Degree VR Video Player
Programmer
April 2015 - May 2015 (2 months)
San Francisco Bay Area

Develop native Unity plugins for rendering stereoscopic video in VR on both Android and iOS.

The Hobby, Dear Leader Apple Watch
Project Lead, Complete Outsource
January 2015 - April 2015 (4 months)
San Francisco Bay Area

Develop light-hearted Apple Watch game about an aspiring dictator that uses his watch to communicate with his staff in order to get things done.

Rich satirical story about taking over the world and keeping your people in line, while meeting as many celebrities as possible and trying to keep up to date on the latest gadgets despite sanctions against your country.

Heavy customization of native iOS components on both the phone and the apple watch using Xcode’s storyboard and layout tools.

Pocket God: Ooga Jump

Programmer and Game Designer

February 2013 - November 2013 (10 months)

San Francisco Bay Area

Released in October 2013 this game takes one of the more popular mini games in Pocket God and upgrades it into a complete game with incredibly high production values.

The game was build with Objective C and our custom openGL engine. The game loads in less than 5 seconds and has no load times after the initial load. A lot of time went into a new texture atlas streaming engine that automatically loads and unloads our 122 2048x2048 32bit texture atlases when the game needs them.

The released version of the game uses PVR compression but the original game was built to use huge, flawless, uncompressed 32bit iPad Retina graphics without suffering from longer load times across all available iOS devices.

The game includes iCloud, StoreKit, and Game Center support.

A huge amount of graphical and animation fidelity was added to the game. all graphics are easily skinnable.

the game includes it's own currency (gems) where a player can earn gems in the game or buy them through IAP. Instead of using a server to store gems and consumables, we use the Keychain and iCloud. While not as secure as a custom server, the types of exploits for the system doesn't justify the cost of creating and maintaining our own backend. Also, because we use both the keychain and iCloud, it's extremely difficult for a user to lose their progress.

We incorporate images and names of real players on the leaderboard into interactive objects in the game. You can literally crush their score to gain extra gems.

We integrated a custom challenge system. Players can challenge each other to a high score duel. In its current state this is modeled on Game Center Challenges but it is built to be expanded with features like like gem wagering.

The UI in the game is getting closer to the more advanced UI systems that I previously built in flash. They are all very rich, have complex transition states and are just fun to play with.

Pocket God: The Comic
Plotter and Co-Creator
January 2010 - October 2013 (3 years 10 months)
San Francisco Bay Area

26 issue comic book collaboration with Ape Entertainment based around the Pocket God brand and characters.

Allan and I were heavily involved in the production of the comic. We plotted and outlined every issue and worked very closely with the writer and artists.

Pocket God: The Runs
Lead Programmer, Project Manager, and Game Designer
March 2011 - June 2013 (2 years 4 months)
San Francisco Bay Area

Unreleased game based on a popular Pocket God mini game.

We developed some really interesting techniques for this game.

Our Flash based JSFL tool got a massive upgrade with the ability to lay out massive 100,000x100,000 px levels. The tools allows us to lay out multiple parallax planes in each level and to see where each background plane will align with the foreground layer when the player is viewing that part of the level in the game.

We can define level collision with vector shapes allowing us to have smooth curves throughout the level.

We can add interactive objects to the level and assign both logic and display classes (with customizable parameters).

Our graphics engine was upgraded with an ability to display hybrid vector/ bitmap primitives. Using vector outlines instead of alpha to show organic shapes is much easier on the GPU. The tool starts with vector primitives with bitmap fills that are designed in flash and triangulates the shapes so it can be rendered in openGL.

there are two uses for this, we can lay out huge shapes with various fills. stone, wood, sand, etc. We can simulate shadow and highlights by using other fills with the same pattern that are darker (which is faster but takes more ram), or apply color adjustments to the same fill (slight render hit but less ram).

the second use is for bitmaps that would normally just use an alpha channel. take a statue. normally the statue will be a rectangular texture with alpha at the edges to define the actual shape. Our tool allows you to trace a vector shape around the statue and render it as a small mesh that only displays the visible pixels. This cuts down on fill rate because the GPU is not rendering the invisible pixels and also makes it to you can disable alpha on that texture which further speeds up the GPU.

Pocket God

Programmer and Game Designer

January 2009 - December 2012 (4 years)

I created this Game with the help of Allan Dye who handled all the artwork. This was our introduction to iOS programming and we spent a total of 1 week on the initial release of the game. It soon went on to sell 8 million copies.

After the initial release, we updated the app with new interactive content every week for 14 weeks straight and eventually added 47 content filled updates over 4 years. Updates include things like new mini games, new islands with interactive features, new play modes like a card based multiplayer battle system. Pocket God is a sandbox app whose framework takes advantage of our quick prototyping capabilities. In many cases we were able to create full featured games in a matter of a week or two.

The entire game is coded in Objective C and uses my own custom OpenGL based 2d engine. I created a series of jsfl plugins for flash to allow us to

leverage Allan's flash animating expertise and export key frame based multilayer animation into our game.

The engine has a unique system that allows us to script complex interactive sequences quickly.

We also have a texture loading system that allows us to completely hide loading times when moving from one island to another despite having 164 texture atlases in the game (not including skins).

Since this game is an open sandbox, the logic in terms of interactivity between objects is extremely complex. The main pygmy object can be interacted with in hundreds of different ways mini games. We were able to successfully manage the amount of complexity and keep the game relatively bug free despite very short concept to production cycles.

Versions of the original Pocket God game were released on Android, Windows Phone, Kindle, and Facebook.

Pocket God: Journey to Uranus

Lead Programmer, Project Manager, and Game Designer

February 2010 - October 2011 (1 year 9 months)

San Francisco Bay Area

Launched in December 2010, this is the first spin off to the original Pocket God game.

this project is coded in Objective C, uses my own custom openGL 2d engine. It uses a refactored version of the game engine from Pocket God. We hired 2 extra programmers and an extra artist to help with this game.

The concept of PGU is to take the Pygmies to space and centers around larger scale mini games inspired by classic games of the past.

We mixed 3d with 2d in this game. All the planets in space are a mix of 3d objects with 2d overlays which creates a unique effect.

Mini Games

Earth - Volcano Blast mini game that is inspired by paper toss. you have to judge the strength of the wind and toss pygmies into the volcano. bonus points for hitting passing ships and sea monsters.

Uranus - Dragons on Uranus mini game is inspired by Joust and is said to be one of the best Joust clones on the app store.

Space - A-Hole in Time mini game is inspired by Tempest and includes a grumpy time traveling alien that shoots abducted pygmies down a tube at robots.

Mercury - Hover Jump mini game is inspired by a really old game called Bump n Jump. the goal is to get as far as you can without hitting the walls and crashing or falling through gaps in the road. aliens will try to bump you off course but you have to cause them to crash first by either ramming them into a wall, or jumping and landing on them.

Space - Quantum Entanglement mini game is about discovery and memorization. Your pygmy has been diagnosed with a quantum sickness and you only have 10 seconds to each quantum particle to make him better.

Earth - Decapithon mini game is inspired by ghosts n' goblins. You are an axe wielding pygmy that needs to progress through areas filled with Zombies. You tilt to move right and left and there is a button to jump. You lob your axes into the air toward any point on the screen that you touch. Each time you hit a zombie, one of their body parts fall off.

World Golf Tour

Flash Architect / Contract

May 2006 - September 2009 (3 years 5 months)

San Francisco Bay Area

Prototyped, designed and created the World Golf Tour game engine from scratch. The game engine uses terrain data and photographs from famous golf course around the world and creates the most realistic golf game experience on any platform.

Each hole has over 3000 images so you can see a view of the course no matter where the ball lands. Each image has it's own 3d transform along with the courses terrain data that allows us to layer elements over the image and

make it feel like they are reacting realistically to each bump, crest, and valley in the course.

We also have a wireframe scrubber that allows the user to see what the terrain actually looks like form any image on the course.

Created the overall design and coded the client portion of multiplayer aspect of the game that allows up to 4 people to play a game together.

Worked on the content pipeline that allows us to process courses, avatars, and club sets quickly and easily.

The avatar system uses many layers of full motion video. Shirts, Pants, Skin Color, Clubs, are all customizable. It took a lot of work on the cpu and memory side to get flash to process these layers of video real time at 60 fps.

Dived deep into the flash memory system. With all this content, memory was a huge issue for us. I created some innovative ways of loading content into the client and "gaming" flash's garbage collection system to allow us to bring all this to a reasonable level. The very first version that had all elements put together took over 1.5GB of RAM. I got it down to 250MB.

Bonus.com
Flash Architect and Creative Director / Contract
September 2005 - February 2008 (2 years 6 months)
San Francisco Bay Area

Concepted, directed and built a new version of Bonus.com, a flash gaming portal for kids.

I recruited a team to handle all aspects of the front end work (including initial product design) while client supplied the server side programming.

The project was done completely in Flash. It supported a game library (we created many of the games), a comic comic viewer, Video Viewer and a full featured avatar system where kids can go online and play games together.

We build multiplayer capability into the portal using a custom socket server that would allow two players to see each other through their avatars with the ability to trigger animations to wave, taunt, or laugh at each other during any activity on the site.

A sample of some of the separate apps we created that fit into the portal.

A collaborative scrap booking app where kids can import images, add pre-made decals and draw in a shared creative space. We created a collaborative movie viewer where they could watch the same movie together with their avatars sitting in the front row. The avatars had the ability to laugh, throw popcorn at the screen, among many other actions.

A collaborative Comic Book Viewer where one user can turn the pages, or mark notes and highlight areas of the comic.

An avatar creator. the avatar creator was skeleton based and allowed the players to customize heads, hands, feet, clothing, etc. The content pipeline for new skins was really easy to use.

A game called Weapons of Math Destruction. where a player is prompted to answer math problems before aliens blast them.

A.S.K. Learning

Flash Actionscript Programmer / Contract

February 2004 - August 2007 (3 years 7 months)

San Francisco Bay Area

Worked on 18 separate projects for clients such as Cisco and HSBC. Most projects were e-learning based but many of them were interactive games that were used to educate customers and train internal staff.

For many projects I was brought to save them when other contractors were overloaded with other projects and didn't tell A.S.K. until right before the deadline. I was very successful at coming into these projects, organizing what was left and developing an efficient attack list so that we could still meet the deadline as well as make sure we had nice technical eye candy to make A.S.K look good in front of their clients.

Among the specific projects. I created 3 point and click style games. 4 multi-level arcade style games. A variety of testing apps with interesting interactive elements. A really cool role playing app that used multiple layers of full motion video.

Yahoo

Flash Actionscript Architect / Contract
February 2005 - June 2006 (1 year 5 months)
San Francisco Bay Area

I architected their rapid prototyping platform using flex, flash and actionscript.

Utilize Macromedia Flex to create a prototyping framework that allows Yahoo! to rapidly iterate user interface concepts and perform focus group testing.

Develop a workflow for developers in other teams to easily get up to speed on Flex technology.

Worked with many Yahoo! properties including yahoo maps and digital home.

iWin, Inc.
Flash programmer / Contract
September 2005 - April 2006 (8 months)
San Francisco Bay Area

I helped iWin ship some games that were behind schedule.

I streamlined the architecture for easier maintainability and retrofitted them for multiplayer capabilities.

Sony
Flash Programmer / Contract
January 2004 - May 2005 (1 year 5 months)
San Francisco Bay Area

Created prototype rich UI schemes for us in future Sony products.

Worked on a prototype for an early version of the playstation 3 UI.
Worked on prototypes for the second version of the Sony Connect music store.
Worked on various TV UI prototypes for the connected home of the future.

Electronic Arts
Flash Actionscript C++ Programmer / Contract
March 2004 - September 2004 (7 months)
Redwood City

Work the UI team for Lord of the Rings: The Third Age for Playstation 2 and XBox.

Utilize a custom flash player that EA created to bring flash content onto their console game.

C++ (in addition to Actionscript) was required to data drive UI elements on the screen.

Detailed knowledge of the inner workings of flash especially the swf file format was required to optimize the performance of flash based UI as an overlay in a 60fps game.

Sony Pictures Entertainment
Team Lead / Flash Programmer / Contract
May 2003 - June 2004 (1 year 2 months)
San Francisco Bay Area

Completely outsourced two projects for SPDE, Wheel of Fortune and Jeopardy.

Both games were created in flash and we delivered through their web site. I also created a special windows client that would allow them to sell the games as a stand-alone Windows application.

The client was coded with C++ using MFC and our own custom active X control to host the flash content. The flash content was created out of many files including XML that we didn't want the users messing with so I packed the whole directory structure into a giant encrypted file and through my plugin acted as a proxy under the flash app. when the flash app would request a file, I intervened and served content from the data file.

Worked on a few more smaller titles like Spiderman 2. I was brought in to rescue some games that were behind schedule.

ManyOne Networks
Flash Programmer Team Lead / Contract
September 2002 - November 2003 (1 year 3 months)
San Francisco Bay Area

I completely outsourced a few projects from ManyOne and recruited / managed a team to take care of their needs.

We created many flash based applications including a mail, calendar, contacts, instant message, plus a complete desktop that had an API for other developers to plug new modules into.

Our instant message app was based on the Jabber server, which we created a full featured flash client for. I also fixed a lot of bugs in the Jabber code in order to get integration with AIM, Yahoo Messenger, and MSN Messenger working.

the email client in particular was pretty insane, we started with an online mail service called critical point that could be sort of customizable though modifying it's jsp pages. I tore into the critical point code and made it return XML that flash could process. The client portion of the email client had multiple windows, the ability to open multiple mail messages with attachments, dockable windows. the ability to view, filter, sort all inboxes, and move messages form one inbox to the other. It was a pretty full featured online mail client for the time.

Also created a flash calendar app with all the normal bells and whistles.

World Sports Exchange
Flash Programmer and Creative Director / Contract
January 2001 - November 2003 (2 years 11 months)
San Francisco Bay Area

Flash, Java, ATG Dynamo

I completely outsourced 4 online gambling games. I handled much the flash programming but also recruited 2 programmers and 3 artists for the team. Another team that we were subcontracting through, handled the server side.

We created Multiplayer Blackjack, Multiplayer Caribbean Stud, 3 Slot machines and a multiplayer lobby and shell for the games.

Everything was 100% flash with a socket server to pass messages back and forth between players.

When a player went to the lobby, they were able to see tables with avatars for actual players sitting down in the seats. They would pick an empty seat to sit in and they would start playing that game.

Python Studios
Owner and Principal
June 1999 - November 2003 (4 years 6 months)
San Francisco Bay Area

Started a company that specializes in advanced flash game and application development.

The experience within this date range was contracted through Python Studios

LeapFrog
Flash Programmer / Contract
March 2003 - June 2003 (4 months)
San Francisco Bay Area

Flash prototyping

I was brought in to prototype game concepts using flash. the target platform was their flash based Leapster platform.

WagerWorks
Flash Architect / Contract
December 2000 - January 2003 (2 years 2 months)
San Francisco Bay Area

Flash, Java, JSP, Engine and Game and Pipeline Development

Flash architect for a startup funded by MGM / Mirage.

WagerWorks was the first credible online gambling company that actually had a chance of becoming legal in the US. Although they didn't quite get that far and had to shift their business to Europe and Asia.

I was brought in to rescue a product that was suffering from a lot of stability problems, had a agonizing pipeline for creating new games, and was really behind schedule.

I helped clean up and refactor the original non-cash version and get it out the door and then architected a new engine that would make it a lot easier for them to launch new games down the road.

The client side was 100% flash with a java backend. Security was paramount and a lot of attention was paid to making sure players could not cheat or somehow deduce outcomes before their bet was made.

I also worked with the on the promotion, sponsorship and advertising side. Helping to create an engine that would be easy to plug sponsorships and

promotions into and coming up with a way to serve double click ads into a 100% flash environment which was not available at the time.

Musicade
Flash Architect and Creative Director
November 1999 - September 2000 (11 months)
San Francisco Bay Area

Flash, ATG Dynamo, Java Servlets

Multiplayer Flash music trivia game. Created initial prototype and grew it into a final product. I recruited a 15 person team that included artists, programmers, sound engineers, and writers (for the questions).

Spunky Productions
Flash Programmer / Contract
June 1999 - November 1999 (6 months)
San Francisco Bay Area

Flash, HTML, Javascript, SQL, ASP

One of the first flash cartoon web portals. I created the cartoon player, worked on some games, and created a pipeline for easy production of cartoons.

Middle Marz
Owner and Principle
January 1999 - June 1999 (6 months)
San Francisco Bay Area

C++, x86 Assembly, MFC, DirectX

Project Lead, Game Designer – PC/WIN98, Direct X, “Urban Warfire”
- Game design, technical design, development and project management of Urban Warfire.
- Recruited and managed 6-person technical and graphic artist team for publisher Stealth Productions.

Arbor Intelligent Systems
Lead Programmer
August 1997 - December 1998 (1 year 5 months)
Ann Arbor, MI

Windows application development, MFC, C++, Custom Video Compression Algorithm Creation

Senior Software Designer and Programmer, Acuson Review Station
- Designer and programmer of UI controls for Arbor's Ultrasound Review Station.
- Low-level design and programming using Microsoft Foundation Classes / C++.
- Received a Patent for my work in User Interface design for a medical application.

Zono Inc.
Lead Programmer
March 1995 - June 1997 (2 years 4 months)
Costa Mesa, CA

C++, Assembly for various processors, MFC, DirectX

Senior Programmer, PC/WIN98, Direct X, "Metal Fatigue"
- Developed extensive interpolation technology for 3D animation, C++ and Assembly.
- Publisher: Psygnosis.

Senior Programmer, PC/WIN98, Direct X, "Ringman", proof of concept
- Developed BSP Polygon Culling Algorithm, C++ and Assembly.
- Publisher: SegaSoft.

Senior Programmer, Level Designer, SEGA Saturn, "Mr. Bones"
- Designed and Developed custom "lossy" FMV Compression Technology.
- Publisher: SEGA.

Pacific Softscape
Lead Programmer
August 1993 - March 1995 (1 year 8 months)
Mission Viejo, CA

Assembly for various processors, C++, MFC, ASIC development

Lead Programmer, SEGA Channel Content
- Developed SEGA Channel Development Tools.
- Publisher: SegaSoft.

Lead Programmer, SEGA Channel BIOS, SEGA Genesis
- Developed and programmed the BIOS for the SEGA Channel hardware adaptor.
- Received 2 Patents for my work in broadband delivery of games.
- Publisher: SegaSoft.

Ringler Studios3D engine designer, ATARI Jaguar # Designed and developed the 3D gaming engine for Br

Lead Programmer

July 1992 - August 1993 (1 year 2 months)

Irvine CA

Assembly for various processors, MFC, C++

3D engine designer, ATARI Jaguar

- Designed and developed the 3D gaming engine for Brett Hull Hockey and Charles Barkley Basketball.
- Publisher: Atari.

Lead Programmer, SEGA Genesis, “Clay Fighter”

- C++ and Assembly development;
- Publisher: Interplay.

Junior Programmer, SEGA Genesis, “ESPN Football”

- C++ and Assembly development;
- Publisher: Sony.

Education

University of Southern California

Bachelor of Science in Aerospace Engineering, Aerospace, Aeronautical and Astronautical/Space Engineering · (1988 - 1992)