



SEARCH

LOGIN

GAME JOBS

UPDATES

BLOGS

EVENTS

CONTRACTORS

NEWSLETTER

STORE

SEARCH

GO

ALL

CONSOLE/PC

SMARTPHONE/TABLET

INDEPENDENT

VR/AR

SOCIAL/ONLINE



Member Login

Email:

Password:

Login

Forgot Password? [Sign Up](#)

Blogs

The Prismata AI: How I learned to stop worrying and love the bots

by Dave Churchill on 12/18/14 03:33:00 pm

Featured Post

4 comments



The following blog post, unless otherwise noted, was written by a member of Gamasutra's community. The thoughts and opinions expressed are those of the writer and not Gamasutra or its parent company.

PROGRAMMING

ART

AUDIO

DESIGN

PRODUCTION

BIZ/MARKETING

Latest Jobs

View All RSS

September 10, 2015

- ✦ E-Line Media
Software Engineer
- ✦ The New York Times
Senior Software Engineer, Games
- ✦ Six Foot
Senior Game Engineer
- ✦ Six Foot
Game Engineer
- ✦ Hello Games
Senior Engine Programmer
- ✦ Tesla Motors Inc.
Software Developer for Visualization

Latest Blogs

View All Post RSS

September 10, 2015

- ✦ The Indie Bubble Revisited (or, Are We All Totally-Doomed, or Just Regular-Doomed?) [1]
- ✦ The Story Behind the Story
- ✦ Stretch Goals - Best Practices for video games crowdfunding [Part 2]
- ✦ Data obsession and the

Hey Everyone! I'm [Dave Churchill](#), lead AI programmer on [Prismata](#). While working for Lunarch I'm also finishing up my PhD in video game AI at the University of Alberta. In 2013 my StarCraft AI bot UAlbertaBot won the [AIIDE StarCraft AI Competition](#), which we have been hosting at UofA since 2010. There are a lot of similarities between the AI for StarCraft and for Prismata, and we've been able to use some state-of-the-art RTS AI research techniques with new twists. For those not familiar with it, [Prismata](#) is a hybrid strategy game that combines ideas from RTS games, card games, and tabletop strategy games. A "how to play" video is [available here](#). Some people describe Prismata as a "real-time strategy game without the real-time". However, Prismata presents many new and unique challenges that make it different from a traditional RTS in terms of AI design. Let's dive into an overview of the AI in Prismata: from initial motivations all the way to implementation details.

"Video game AI is about making agents in the game do things that engage the player and make them have fun". - Max Dyckhoff (AI programmer for the Halo series)

This was our goal in Prismata - to design an intelligent, dynamic, and flexible AI system to allow players to have the most fun. The AI was designed with several goals mind. We want it to:

- **Give new players a computer opponent to learn the rules more easily.** Jumping right into player vs. player games can be scary when you aren't sure of all the rules or techniques of a new strategy game. The Prismata AI offers several difficulty settings from 'docile punching bag' all the way up to 'Skynet wannabe Master Bot,' ensuring that when you step into the PvP arena you'll be ready to brawl.
- **Give more experienced players a challenging opponent to practice against.** The AI can be a strong opponent for you to practice strategies or build orders against. It won't even give away your secrets and "leet strats" to another player.
- **Give the single player campaign more intelligent and dynamic gameplay.** Single-player missions/campaigns in most video games are scripted sequences of events that play out the same way every time, allowing you to just memorize strategies in order to defeat them. In Prismata, the missions are guided by our dynamic AI system, which ensures you don't fight against the same tactics every time you play. Our AI can adapt to any game situation we give it without having to write specific behaviours for every scenario we create. This means that we can pump out more content without worrying about hard-coding rules and strategies for each one, so we can release content faster.
- **Allow players queued for ranked play to play against the AI while waiting.** If you've ever watched top level League of Legends or StarCraft players streaming, you know that top-level ranked queues can sometimes take a while to find a player. In Prismata, you have the option to play against a computer opponent while you are waiting, and instantly jump to your human opponent when your queue pops.
- **Help identify game balance issues.** With a strong AI, we don't need to rely completely on player testing in order to test whether or not new units are fair and balanced. We can play millions of AI vs. AI games to test to see if some units are too strong or too weak before we release them to our users to play with.

Prismata AI Complexity

politics of facts [4]

- Seeking Home in a Changing Industry - IGDA Summit 2015 [3]

Features

View All RSS

September 10, 2015

- Postmortem: Pinball-RPG hybrid *Rollers of the Realm* [6]
- Best of 2014: Gamasutra's Top Games of the Year [10]
- Q&A: New AAA studio from 2K Games all about players' stories [7]

Special Reports

- Game Design Deep Dives
- A Peek Into the Future of VR: Oculus Connect 2014
- The YouTube Effect
- VR/Advanced Input/Output
- Digital Publishing
- Microconsole Game Development
- Crowdfunding and Alternative Funding

Press Releases

September 10, 2015

PR Newswire

Ludia Announces \$30 Million Investment and 100 New...

Create Android Apps in Under a Minute with AppsGey...

Youzu Attended 'Pocket Gamer Connects' to Share Ga...

View All

Games Press

- Your Horse, Your World, Your Adventure: "Horse..."
- Your Horse, Your World, Your Adventure: "Horse..."
- Starbreeze AB: PAYDAY 2 DLC "Gage Chivalry Pack"...
- Starbreeze AB: PAYDAY 2 DLC "Gage Chivalry Pack"...
- UBISOFT® ANNOUNCES AUSTIN WINTORY AS COMPOSER...

In game theoretical terms, Prismata has the following properties:

- **Two player** - While Prismata does have single player puzzle and campaign modes, this article will discuss the 1 vs. 1 competitive form of Prismata
- **Alternating Move** - Players take turns performing moves like in Chess, with one side being declared the winner when they have killed all the units of their opponent. If the game enters a state where no player can kill all the other player's units, then the game is a draw
- **Zero Sum** - The outcome of a game of Prismata is a win, loss or a draw
- **Perfect Information** - All players in Prismata have access to all of the game's information, with no player knowing anything that the other player doesn't. There are no decks or hands to keep secret from your opponent
- **Deterministic** - At the beginning of a game, a random set of 5 or 8 units (depending on the game type) is added to the pool of purchasable units. After this randomization of the initial state, the entire game is *deterministic* (ie: no randomness)

Part of making a fun AI is making an intelligent AI that can play the game well.

A common myth among beginners: "Prismata is simple, you can just solve it!"

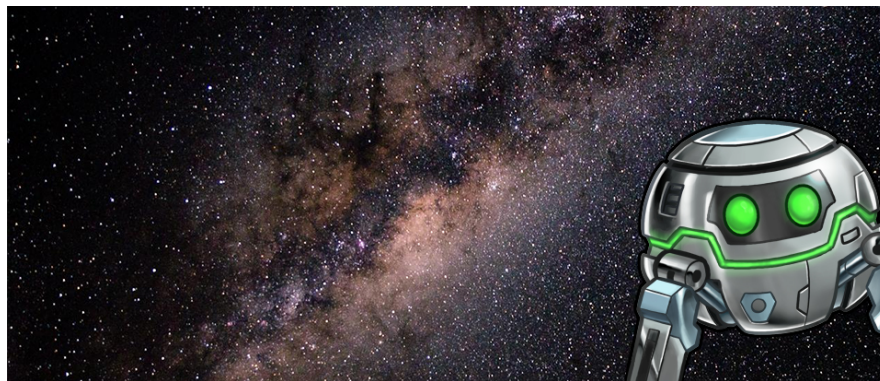
Actually, Prismata is a very complex game. In order to prove this, let's take a look at the two most popular ways to measure game complexity:

1. State Space

This is the term used to describe the number of possible positions (or states) a game can have. Take for example the game tic-tac-toe, which has 9 possible spaces and 3 possible values for each space: Blank, X, O. If each of the 9 spaces has 3 possible values, then we have 3 to the power of 9, or $3^9 = 19683$ possible tic-tac-toe boards. If we then subtract the number of impossible states (ie: all X's) and the states that are equal due to symmetry, that number shrinks to 765 states, which is easily solvable by computers. More interesting games have much larger state spaces. Checkers is the largest game that has ever been solved by computers. With approximately 5×10^{20} (five hundred billion billion) states, it took over 200 computers 18 years to solve (finally completing in 2007). More complex games like Chess (with 10^{47} states) have not been solved, even though AI researchers have been working on them since the 1950s.

In Prismata, we can calculate a rough estimate of the state space as follows. In a typical Base + 8 game players have access to 11 base units and 8 random units, for a total of 19 units per player, or 38 in total. If we give a conservative average supply limit of 10 per unit per player, then the number of possible combinations of units on the board at one time in Prismata is 11^{38} . We then have to consider that each unit can have different properties, can be used or unused, have different amounts of hit points or stamina, etc. If we give a very conservative estimate of an average of 30 units on the board at a time, each with 3 possible states, then we get 3^{30} combinations of properties of those units, or about 10^{14} . Now factor in the fact that Prismata has about 70 units (so far) of which 8 are selected randomly, and we have about 10^{10} possible starting states in Prismata.

Multiplying this all together gives a very low estimate of 10^{64} for the state space of Prismata, or about a billion billion times more complex than Chess. Good luck solving it!



There are about 10^{80} atoms in the universe, so 10^{64} is pretty big. Prismata has at least as many possible states as atoms in the Milky Way Galaxy!

2. Action Space

This is the term that describes the number of possible moves you can do at any given time in a game. You can consider the action space as a measure of complexity because it shows how hard it is to make a *single decision* in a game. In a typical board game like Chess, your turn consists of taking a single piece and moving it to another position on the board. An average chess turn has approximately 30 possibilities. Taking a turn in Prismata consists of 4 main strategic decisions: How to defend, what to buy, how to activate abilities, and how to breach your opponent. Even if we consider these problems individually, each of them

View All RSS

Calendar

View All Submit Event

- **Connecting the world on the sea – Webinar**
Nieporët, Poland
09.12.2015
- **Game Music Connect 2015**
London, United Kingdom
09.15.2015
- **Captivate Conference**
Austin, United States
09.18.2015
- **Gamecon 2015**
Atlantic City, United States
09.19.2015

About

- **Editor-In-Chief:**
Kris Graft
- **Blog Director:**
Christian Nutt
- **Senior Contributing Editor:**
Brandon Sheffield
- **News Editors:**
Alex Wawro
- **Advertising/ Recruitment:**
Jennifer Sulik
- **Recruitment/ Education:**
Pocco Jimenez

Contact Gamasutra

Report a Problem

Submit News

Comment Guidelines

Blogging Guidelines

How We Work

Advertise with
Gamasutra

Gama Network

If you enjoy reading this site, you might also want to check out these UBM Tech sites:

Game Career Guide

Indie Games

has an exponential number of possible sequences of actions. In computer science we call these type of problems NP-Hard. In other words, Good luck trying to solve them! Let's take buying units as an example.

Buying units in Prismata is what is known as a [Knapsack problem](#) in Computer Science. If we consider unit prices to be objects, and our total resources to be a knapsack, we are trying to fit as many of those objects into our knapsack that we can. For example, with just 8 gold and 2 energy in Prismata, there are 18 possible combinations of units that can be bought! (For the Prismata-inclined, they are: Nothing, E, EE, EEE, EEEE, D, DD, ED, EED, C, CC, EC, EEC, DC, B, EB, DB, A, EA.) By the middle of the game, the number of possible buying actions can easily reach tens of thousands. If we combine this with the number of ways to block, use abilities, and breach, we find in most mid-game situations, there are literally millions of different ways in which we can play a single turn!



With these resources, there are over 25,000 combinations of units that can be purchased... in the Prismata base set alone!

Prismata AI Techniques

Writing AI for complex games is very difficult, and often times game developers will take 'short cuts' with the AI in order to make the AI look stronger than it actually is. In Prismata, both players have access to all information about the game. There are no decks, hands, or RNG. This makes writing an AI for Prismata more difficult than most games, for several reasons:

- There is **no cheating**. There is no hidden information you can give the CPU to help it 'cheat' to make it look more intelligent. For example, in real-time strategy games the AI system will often cheat by knowing where your base is without scouting, or in a standard card game you could let the AI know what units are in an opponent's deck. Also, you can't just give the AI a ton more resources with which to make armies faster because you can see exactly how many resources they have.
- There is **no RNG to manipulate**. Some games often manipulate the RNG in order to give the AI an advantage, such as giving them higher dice rolls or letting them critically hit more often. Prismata has none of these shenanigans, so you always know that any move the AI makes is legit.
- You **can't hide mistakes**. With everything in view, there is no way to hide the fact that your AI made a mistake or cut a corner. Brian Schwab (former Blizzard AI programmer on Hearthstone) said about the Hearthstone AI at GDC 2013: "One of the things that's great about Hearthstone is that the opponent can't see your hand, and so it doesn't know that you're making a sub-par choice. As long as I don't target somebody stupid, it still looks intelligent"

There is no "looking intelligent" in Prismata. The AI is either is intelligent, or it isn't. We need to use state-of-the-art artificial intelligence techniques in order for the AI to play the game at a competent level.



Game Tree Search Algorithms

We can't solve Prismata by brute force, so we need a way to manage the complexity. To drive the AI in Prismata, we use a method called *Game Tree Search*. Game tree search is essentially a method that checks "If I do this action, then my opponent does that, then I do this... which is the best move?" You may have heard of grandmaster chess players who are able to look 10 to 20 moves into the future in order to choose a good move. They are actually executing a form of game tree search! The specific algorithm we use to perform the game tree search is called Monte-Carlo Tree Search (MCTS). This algorithm and variants of it have been used by several world champion computer programs for playing board games such as Go. Basically, the algorithm does many simulations of the game (called "playouts") to predict what will happen in the future if a player makes a certain move. Each time a playout is performed, the algorithm gains more information about which moves are better and which are worse, eventually deciding on one it thinks is the best. There are several advantages of using MCTS in a game like Prismata:

- It can deal with any game situation you throw at it. We don't need to write an entirely new set of behaviours or algorithms every time we add new units or change existing units.
- It is an **any-time** algorithm, meaning you can run it for as long or as short as you want and it will always give you a move that it thinks is the best. The longer you let it run, the better the move it will produce, which is why we give the Master Bots a longer time setting!
- It doesn't rely on a pre-defined **evaluation function**. Some games like Chess have rules for determining who is 'in the lead', such as assigning points to pieces like a Queen being worth 9 and a Rook being worth 5. By using playout evaluations this sort of thing is no longer necessary. This means that we don't have to spend time coming up with formulas for how powerful units are in order to use the algorithm. By simulating what will happen in the future the algorithm automatically determines which moves are good and which are bad all by itself.
- It performs better than many other algorithms in games with a high number of possible moves (like Prismata) due to the intelligent way it selects which move to traverse in the game tree.



The Prismata AI in action! :)

Dealing With a Large Number of Actions

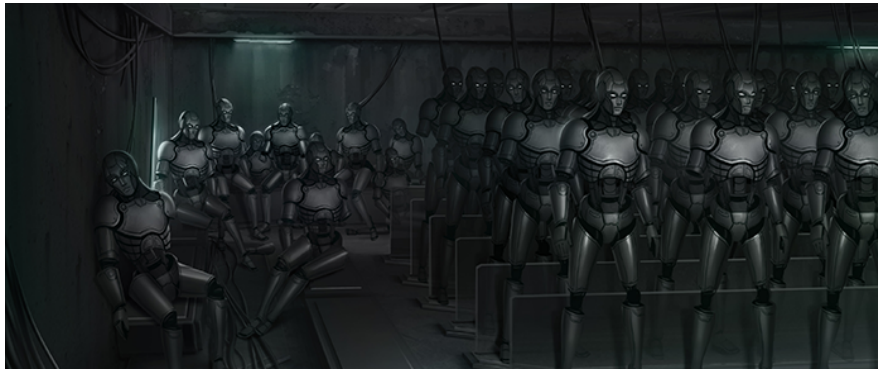
If you've been following along, you might now be thinking "but you just got finished saying that there could be millions of actions in Prismata, surely even MCTS can't handle that many actions." And you'd be right. This is why we've developed a system for dramatically reducing the number of moves that MCTS deals with from possibly millions down to just a few dozen good candidates. Part of my research in Real-Time Strategy Game AI was writing algorithms to control how units attack in StarCraft, or "unit micro" as players call it. This is another scenario where a player has a bunch of units, each with their own possible actions to perform, resulting in an exponential number of moves possible at any given time.

We found a technique that works very well in these domains, called a "portfolio" approach: instead of trying every possible move that we can perform, instead we use a "portfolio" of algorithms to generate a much smaller, yet more intelligent set of moves to search over. It turns out that Prismata has some nice properties that make this method especially powerful. Prismata has 3 distinct game phases: Defense, Action, and Breach, each with their own rules and set of goals. In the defense phase you are trying to most efficiently keep your units alive, in the action phase you are trying to perform actions to generate attack and kill your opponent, and in the breach phase you are trying to most effectively destroy your opponent's units. We can break these 3 phases down even further by considering the action phase as two separate sub-phases: using abilities, and buying units, leaving us with 4 phases. While these phases are technically all part of the same turn, even the best human players often consider them as independent problems that they try to solve separately, as the entire turn would be too much to mentally process at the same time.

We combine this idea of separately solving phases with the computational power of game tree search. Prismata's AI has a number of algorithms for attempting to choose good actions for each individual phase. For example, in the defense phase we could have one algorithm that tries to minimize the amount of resources you will lose if you block a certain way, while another algorithm would try to maximize the amount of attack you have remaining to punish your opponent with. Similarly, one algorithm for the card-buying phase could try to maximize your economic output for the next turn, while another could try to buy as much defense as it could to thwart an incoming enemy barrage.

We then search over all combinations of these algorithmically-generated moves in the high-level MCTS algorithm to decide which combination of phase moves results in the best overall move for the turn. This reduces the number of possible moves searched by MCTS from (possibly) millions down to something manageable. By not considering all possible moves this does mean we might miss the perfectly optimal move for a turn, but that optimal move was nearly impossible to find in the first place! Right now, the

number of moves that our MCTS algorithm searches per turn is somewhere in the range of 10-40, depending on the complexity of the scenario.



Customizable AI Parameters

An important goal for the Prismata AI was also to have multiple difficulty settings for the AI so that players of any skill level can enjoy it. The portfolio approach actually makes different difficulty settings quite easy to manage. We start with the hardest AI, then we can turn off certain algorithms that make the AI play well in specific situations. For example, the main difference between the Easy and Medium AI is that the Easy AI does a really poor job of buying defensive units, leaving it vulnerable to attacks. All of the AI settings and parameters are completely data-driven, meaning that we only need to change a value in a text file to change how the AI behaves. In fact, starting from the Master Bot, all the other difficulty settings were created in about 15 minutes without editing a single line of code!

AI Implementation / Technologies

"But isn't simulating the entire Prismata game state thousands of times really slow?" It really is, especially in ActionScript. For this reason, the entire Prismata engine was completely re-written in C++, minus the GUI, networking and animation logic. This stripped-down version of the engine is highly optimized for speed, and is compiled from C++ back into ActionScript using a tool called CrossBridge. CrossBridge takes the optimized C++ assembly output and converts it into ActionScript bytecode, yielding code that is several times faster than if it was coded in ActionScript in the first place (around 10x slower than native C++ code). This ActionScript code is then injected back into the Prismata engine, and runs in a separate thread.

Whenever Prismata wants a decision to be made by the AI, it sends off the current game state information to the AI thread and waits for the result. This means that we don't have to interleave the AI logic with the animation / game logic, resulting in far more available CPU thinking time for the AI. A nice byproduct of this process is that there actually exists a native C++ version of the Prismata engine, which is capable of performing millions of moves per second. We can use this C++ engine offline for a number of tasks, such as automated QA testing, AI performance tuning, and game balance testing. This engine also has its own simple GUI written in OpenGL so that we can test AI functionality without requiring it to be compiled back into the Prismata client, resulting in much faster AI development cycles. Here's [a short video showing some of the AI testing in action](#).

Secret AI Experiment!

As an experiment to see how the most recent version would perform in the wild, we decided to secretly test the AI against human players on the Prismata ranked ladder. For 4 hours one morning, the bot automatically queued and played ranked games under the name "MyNameIsJeff" and climbed from its initial Rank III up to 50% into rank VI with a record of 20-13. By adding random time delays to the bot's clicking, we also seemed to have passed the Prismata Turing Test, with nobody saying that they felt that they were playing against a bot. Here is a screenshot of the bot's home screen after we turned it off:



Conclusions

By now, you should have a pretty good understanding of the AI in Prismata: Why we did it, how complex it is, and how we have implemented it. The AI in Prismata will never be 'finished'. It will always continue to be improved with new features, difficulties, and playing strength improvements. I really hope that you have fun playing against the bots, and feel free to give us any feedback about the AI at any time :)



Related Jobs



E-Line Media —
Tempe, Arizona,
United States
[09.10.15]
Software Engineer



The New York Times —
New York, New
York, United States
[09.10.15]
Senior Software
Engineer, Games



Six Foot — Houston,
Texas, United States
[09.10.15]
Senior Game Engineer



Six Foot — Houston,
Texas, United States
[09.10.15]
Game Engineer

[\[View All Jobs\]](#)

Comments

Michael Joseph

19 Dec 2014 at 5:51 am PST



Very interesting game. Thanks for the glimpse inside the AI.

Speaking of bots, Hearthstone has been known to have lots of bot players online. Since Hearthstone has some pay2win aspects, bots allow folks to gain access to items they might otherwise purchase meaning Blizzard is hurt financially through the use of bots. But that doesn't seem to be the case with Prismata. What are your thoughts on people using AI bots to play Prismata? There are some legitimate uses where a player who's learning wants to use a bot to give them suggestions on what actions to take and the logic behind it.

Also I think it will be interesting to see if the lack of randomness in the game makes the game

less appealing to casual players who like being dealt that game changer card that saves them from the ashes. That lack of randomness in competitive play does in that sense make it more like a Starcraft without the quick reflex reqts. I haven't followed up with Starcraft 2, but I recall reading some time ago that casual players ran off because the competition was too fierce. I personally don't have a problem with that because a complex game that rewards mastery shouldn't have to be nerfed into checkers.

Pure skill based games I think have always been intimidating to a lot of folks. No one to blame when you lose and often times the games can feel less friendly and even cold and clinical. In the war gaming scene there's really strong opinions against games that don't model "the human factor" and "luck" and "sh*t happening" and so nearly all war games involve RNG, dice rolls or card drawing from a shuffled deck.

[Login to Reply or Like](#)

Sung Kwon

19 Dec 2014 at 10:33 am PST



I think people ran off from casual Starcraft 2 because it was too mechanically demanding (worker production might as well as be on auto-build mode for everyone under grand master and non zerg) and has more to do with 1v1 focused multiplayer scene where each losses are more damaging to one's ego.

Competition wise more popular games like LoL or DotA2 have its own equally fierce pro scene, but it's just... easier to dabble into both physically and mentally.

So with that being said, this game doesn't have mechanical problem from what I've seen of their "how to play prismata" video (very cool video BTW!~), but I wonder how it deals with social aspect?

[Login to Reply or Like](#)

Sung Kwon

19 Dec 2014 at 10:40 am PST



Everyone should check out their "How To Play Prismata" video linked on author's profile page. What a cool looking game!

My newbie impression is that it's a card game where you have direct control of your resource management with combined battle phase. For anyone having played hearthstone, think of a game where Druid's Wild Growth is just built into the core game through cheap 0/1 minions. Then for attack, you aren't really dealing with individual cards (until breach phase) but rather totality of the board, so cards work like stat stick (again, until breach phase). Now that may sound like I'm just shitting on the combat phase but I actually think it's a cool way of removing a layer of mechanics so that the game can help players deal with its much more complicated resource management area. That and the fact that defending side actually get to make some very important decision (like who takes those hits) might have huge implication on player psychology (just watching your board getting destroyed slowly is really damaging IMO, and this game could avoid that for most of the early game).

[Login to Reply or Like](#)

Eric Ball

19 Dec 2014 at 11:50 am PST



One factoid I remember from a podcast with Sid Meier talking about the AI in Civilization Revolution went something like:

In a 4 player PvP game, given similar skill levels, each player is happy if they win more than 25% of the time. However, that same player playing against 5 computer players is unhappy if they win less than 50% of the time.

CivRev is also an interesting study in how PvP playstyles and tactics evolve over time. The CivRev AI represents a "day 0" for playstyle & tactics. However, since then strategies & tactics have advanced significantly; generally finding ways to gain advantages more quickly. Thus play against the CivRev AI has become almost irrelevant as preparation for PvP.

[Login to Reply or Like](#)

[Login to Comment](#)

UBM TECH BRANDS					COMMUNITIES SERVED	WORKING WITH US
Black Hat	Fusion	HDI	Network Computing		Enterprise IT	Advertising Contacts
Cloud Connect	GDC	InformationWeek	No Jitter		Enterprise Communications	Event Calendar
Dark Reading	GTEC	Interop	Tower & Small Cell Summit		Game Development	Tech Marketing Solutions
Enterprise Connect	Gamasutra			Terms of Service	Information Security	© 2015 UBM Tech, All rights reserved
					IT Services & Support	
						Contact Us
						Licensing