I’m here today to talk about AI. Why is this an interesting to talk about? Because, I’m going to argue, AI is playing a more and more important role in game design and development. Increasingly it is AI that makes great games great!
Here are some of the games I’ve worked on. I’ve worked on some aspect of the AI of all of them. Most recently, I was the project lead on a game called Third Eye Crime (released 5/14 for iOS) that was explicitly designed as an AI-based game.
Recently I started a new company with some ex-Irrational (Bioshock) developers, named The Flame in the Flood.
We’re working on a new game called The Flame in the Flood, which was successfully Kickstarted earlier this year (in November ‘14). This game too has a some very interesting AI behind it. But we’ll get to that presently.
Here's the agenda for today.

- What is [Game] A.I.?
  - Plus: A Couple Fundamental Truths
- Where does A.I. Live?
- Further Thoughts
WHAT IS [GAME] A.I.?
A.I. = Artificial Intelligence

Generally
- Make computers smart (primary)
- Make computers think like humans (secondary)
Early A.I.

“Within a generation ... the problem of creating 'artificial intelligence' will substantially be solved.”
- Marvin Minsky 1967

“Machines will be capable, within twenty years, of doing any work a man can do.”
- Herbert Simon 1965

Early AI was characterized by a great interest in common sense and human psychology. Also by great optimism. Quotes like Herbert Simon’s are quite inspirational, until you realize that he said it half a century ago.
What those early AI researchers ran into was what I call the central fundamental truth of AI, which is that the stuff that looks easy turns out to be hard, and the stuff that appears to be hard turns out actually rather easy.

The best illustration of this is this scene from 2001: A Space Odyssey. Audiences saw this scene and thought to themselves “Wow! That computer is smart, it’s playing chess!”
But in fact even at the time, chess was a well-understood, largely solved problem. What the audience SHOULD have thought was “Wow! That computer is smart! It knows how to understand and speak a natural human language and engage in a conversation!” That’s something we still can’t do. And we’re not particularly close.
What happened to AI, very generally, in the years since that early optimism, is that the field broke up into many many different subfields, as this general, vague notion of “intelligence” got gradually broken down into it’s constituent parts, reflecting the new understanding that “intelligence” is not one monolithic capability, but thousands of tiny ones.

Today, many of the places where the greatest strides are being made are in application areas that make extensive use of what are, at base, statistics techniques. Game AI is not yet making much use of these techniques – but they will!
But What is *Game* A.I.?  

So what about Game AI?
If we talked about “Game AI” in the early days, we’d be talking about something like Chess. As an extremely discrete game of “complete information” (i.e. no aspect of the state of the game is hidden from either player) Chess leant itself well to the techniques of early AI.
But it didn’t take long to get to Pac Man. I see Pac Man as the beginning of Video Game AI. After all, each of the four ghosts have their own rules for how they chase the player down within the maze. It is the combination of their behaviors that makes the game, and makes it fun. Without the ghosts as hostile NPCs, there would be no gameplay at all.
I’m going to make 3 points about what I think makes game AI unique.  
Point 1: whereas traditional AI might be about make computers smart, game AI is about making them fun.  
Now smart is often not fun – imagine a chess game that is so good that it never lets you win (no fun!)  
Conversely, it is fun to be “not smart”, allowing by design NPCs to make mistakes that the player can learn to recognize and exploit.
Point 2

Game AI is tightly interwoven with Game Design

AI is the gameplay.

1. AI coders need great design intuition, OR
2. Designers need to be able to code, OR
3. We need great tools for designers to express behavior

Point 2: Game AI is inseparable from Game design. Most of the time, the AI IS the gameplay.
That does lead to questions about who should be the ones on a development team who authors AI, coders or designers?
It remains a point of interesting debate within the industry whether it should be coders with great design sense, designers with coding ability or some sophisticated system or tool that tries to mediate between the two. Reality will ultimately be combination of the three.
Point 3

Game AI benefits from lots and lots of content.

If it looks and moves right, players think it’s smart.
If it looks or moves wrong, players think it’s dumb.

Point 3: this is really a point about presentation. The AI has to present well, and express as clearly as possible to the player all the interesting things that are going on beneath the hood, otherwise the AI does not end up feeling like a living thing. Usually this requires a huge amount of content, in the form of lots of animations, lots of dialog, lots of metadata marking up the world, etc.
Fundamental Truth:
The hard stuff is easy, and the easy stuff is hard

Ultimately we too face this dilemma.
For us, the form this usually takes is that we end up, on almost any project, spending 90% of our time on the “easy” stuff, the stuff the player will never notice. Things like pathfinding. The essentials of pathfinding have been known for 40 years. So why do we still spend so much time on it? Because our worlds keep getting bigger, more dynamic, and overall more complex. So the bar keeps getting raised.
Where do we actually find AI in games?
The answer is “in all kinds of places.” What I’m going to do in this section of the talk is present a breakdown many of the places where AI can exist in games. And we’re going to go from the small scale to the large. So we start with the AI within a single character.
1 Character
Even at the character level there is an important distinction to draw, between adversaries and allies.
As in Pac Man, adversarial AI ARE the gameplay. Without their opposition to the player, there would be no game.
When talking about adversarial AI, I often talk about three primary requirements.

1) **Workability**: these AI represent in most cases a LOT of content and behavior. So they have to be workable for the AI coders and designers. If behavior or gameplay needs to be tweaked, it needs to be tweakable in a clean way.

2) **Coherence**: we’re generally trying to make these NPCs feel alive, and that requires them to have coherent goals, pay attention to the important things that are happening in the world. Observe the right priorities. Avoid “dithering” (quickly flipping back and forth between two options, which looks incredibly robotic and immediately breaks the illusion of life).

3) **Transparency**: these AI are ultimately GAME MECHANICS, and as game mechanics, we need the player to develop an intuition for the decisions they are making under the hood and why they are making them. And once that intuition is developed the player needs to be able to predict the AI’s decisions: “if I want the AI to do X, that means I should do Y.” That is the “player agency” at the heart of gameplay, that turns “a series of random attempts” into “a series of interesting choices” (Sid Meier’s famous definition of “game”).

---

**Adversaries**

- **Workable**
  - Designer and engineer can author and maintain lots of “A.I. content”.

- **Coherent**
  - A.I. seems alive
  - Focused attention, priorities

- **Transparent**
  - player can explain and interpret actions
  - player can predict
Here’s an extreme example of “transparency”. In my game Third Eye Crime, we actually give the player telepathy. The red glow in the video above, represents the AI’s beliefs about where the player is. If you avoid the red glow, you don’t get seen. Thus the AI is extremely transparent – you know, as a player, that it is always going towards the highest concentration of red.
Now, ally AI, or companion AI. This is a hot topic. Two of the most lauded recent companion AIs are from 2013 (Elizabeth from Bioshock: Infinite and Ellie from Last of Us).

Companions (with only a few notable exceptions) do not represent the core gameplay mechanics of the game, rather they represent the core storytelling mechanic and information- and emotion- and tone-conveyance mechanic. Because the player is often meant to form a relationship with them, they need to be charming, helpful, sympathetic, emotive, and not get in the way. All are incredibly difficult to achieve, and they are as often achieved through great art and appealing character design as through any particularly AI.

The danger of acting stupid is particularly great with companions – it’s one thing if your enemy does something stupid and you get to take advantage of it. It’s another if a character that is meant to be helping you consistently messes up and/or gets in the way, and / or behaves in a non-human manner.
http://youtu.be/2viudg2jsE8?t=10m46s

Here’s a scene from Bioshock: Infinite with Elizabeth transitioning smoothly between several tightly- and loosely-scripted sequences.
Level 2 is the squad level: controlling small groups of characters to act as a coordinated team. Typically the hard part here is making decisions and acting as a group while also maintaining the reactivity and coherence of an individual. Kind of like with people, balancing group versus individual needs can be tricky.
Some examples: sports games in which teams have to work together, or military sims, when squads have to maneuver together intelligently.
In the Halo games, squads-level intelligence was controlled through an encounter system, where large-scale battles with tons of guys were choreographed. We would start with some high-level sketches of a space, the major structural pieces and objectives, player approach, fallback points, etc.
We would turn that into a kind of state machine which would shuffle groups of guys around as the player pushed forward through the level.
Here’s an example of a complicated encounter, with tons of fallbacks, reinforcements and last-stands.
1. Character
2. Squad
3. Crowd
With crowds, the first challenge is ensuring visual variety – in the AC screenshot, there’s enough clothing, body-type and body-posture variety to sell the crowd, but not so much variety that it looks ugly or uncohesive.
The second challenge is motion – moving tons of characters around in way that is not a total mess.
At this point, with this number of characters, we are not making decisions at the “individual brain” level, we are solving for large flows, and then dropping the individuals into those flows (with perhaps some local / individual decision-making layered on top).
At the strategy level, we again have untold hundreds or thousands of characters, but now we are concerned with the meta-brain that is coordinating ALL their actions to achieve a goal.

Star Craft II has received a particular amount of attention thanks to the yearly Star Craft AI Competition, in which AIs created by different developer teams are pitted against each other to see which performs the best.
At this level, one of the most important pieces of the AI is the spatial analysis — many of the decisions that these AI have to make center on WHERE resources should be acquired, defenses built, attacks launched, etc. And all of those decisions depend on a good understanding of the space they occupy.
1. Character
2. Squad
3. Crowd
4. Strategy
5. **Societies**
Clear at the “society” level, we simulate not individuals or armies but entire cities or civilizations.
One thing you might note is that in a way, each scale level (character, squad, etc.) needs to solve TO SOME DEGREE each of the levels below it. Civ is a great example, since there is a warfare/strategy level to the game as well as a great deal of character-level AI imbued in the leaders of the various rival civilizations. Each of those leaders has a perceptible style and personality. Gandhi, for example, loves to drop nukes.
1. Character
2. Squad
3. Crowd
4. Strategy
5. Societies
6. Player Modeling
Now, rather than pulling out any further in terms of spatial or temporal scale (you can see how pulling out into, say, the galactic level, is really just the same as the “society” scale) we instead go a little “meta”.

Player modeling is all about watching the player play the game and trying to derive information from that.

In the two cases above for example, the game would observe the player and then try to create an AI that plays just like them. Essentially trying to make a player clone.
My favorite example of this is Left 4 Dead.
In Left 4 Dead, the AI Drama Manager would build an explicit model of “gameplay intensity” for each player (based on very simple game triggers) and then intentionally ramp up intensity until a threshold has been reached. After a short period of time at this level, the intensity would be backed off, giving the player a breather (but also adding a new tension, a tension of anticipation, to the mix).
1. Character
2. Squad
3. Crowd
4. Strategy
5. Societies
6. Player Modeling
7. Procedural Generation
The other super ultra hot topic right now is procedural generation. This coincides especially well with the rise of the indie gaming scene, as it is a (presumably) cost-effective way for a small team to produce an expansive world. (Yes, Minecraft is usually discussed in the context of USER-generated content, but remember that before the player starts to do anything, the beautiful and explorable world first has to be generated).
Our game, The Flame in the Flood, takes place on a PCG river. Here’s a video we made about that process.

http://youtu.be/SwSUPFovMTg
Now the real holy grail is procedurally generated stories. No one has cracked this nut yet. Some of the most compelling attempts (such as Façade and Versu, above) have involved combining proceduralism with human-authored story fragments.
Now maybe the most important kind of AI of all.
Never thought I’d see a slide with League of Legends and Farmville on it. You can’t get further away on the hardcore-to-casual spectrum. But what they do have in common is (a) they’re social and (b) they generate untold terabytes of gameplay data.
### Analytics & Big Data

<table>
<thead>
<tr>
<th>League of Legends (Oct 2013)</th>
<th>Farmville 2 (Dec 2012)</th>
</tr>
</thead>
<tbody>
<tr>
<td>▪ 32.5 Million <em>daily</em></td>
<td>▪ 40 Million Monthly</td>
</tr>
<tr>
<td>▪ 1.3 Billion hours played <em>per month</em></td>
<td>▪ 8 Million Daily</td>
</tr>
</tbody>
</table>

These numbers are staggering. These games are REALLY popular.
We don’t know exactly what they are doing with all the data they are generating and collecting, but we can make some guesses.
CONCLUSIONS
I like coming into work and having radically different types of challenges every day.
I think League of Legends and Farmville represent the future of games. We should all make sure we know our “linear regression” from our “principle components” if we want to contribute to games in the next decade!
Questions?

Damian Isla
@damian_isla