

You had me at 'AAAAHHH' – On the importance of reactions in game AI

Sergio Ocio Barriaes

1 Introduction

Reactions are an important aspect of an AI's performance, and getting them right is fundamental to any modern AAA game AI system.

This chapter is NOT about decision-making, it is about good reaction design regardless of the decision-making approach. How should our NPCs react to different stimuli? We will talk about our objectives when working on AI reactions, and how these can contribute to building good game AI. We will also talk about some of the basic reactions that every open-world/action AAA game should have.

Disclaimer: there is a wide array of video game experiences, and each presents different challenges. Some of the tips presented in this chapter may or may not be applicable to other genres.

2 Apparent Artificial Intelligence

Artificial Intelligence – it is everywhere. From manufacturing to robotics, self-driving cars to medical applications, or simply to enhancing your photos, AI has become one of the hot topics of our era. It has even been compared to the introduction of electricity a century ago (Ng 17). But what about game AI? Is it real AI? To answer this question, we should talk about our goal. What is the problem that we are trying to solve?

In a video game, our number one goal is to create a fun and rich experience. Our objective is not to build a perfect simulation, nor it is to always find the optimal solution to our problems - we are crafting an enjoyable experience. *An Apparent Artificial Intelligence*. An illusion.

3 Reactions and Decision Making

When an agent perceives a stimulus, this information goes to some decision-making system, choosing whether to ignore the event or handle it. If the event is handled, the AI is “reacting” to that change. In a sense, every change of behavior or strategy made by the AI can be considered a type of reaction, but this chapter is more focused on those changes that are a result of an incoming event that needs to be handled with some urgency (e.g. hearing a gunshot) vs other types of situations such as “it is 5 pm and it is time for me to stop working and need to start heading home”.

There are multiple techniques that have been used in video games to control decision-making in general, reaction handling in particular. Some common ones are:

- State-based solutions, such as Finite-State Machines (FSMs) or their Hierarchical versions (HFSMs) can handle reactions as state transitions, so there is no real difference between a reaction and another type of behavior change. However, this does not scale when the number of reactions is very high, as the number of transitions can be hard to control.
- Behavior Trees (BTs) are, in some ways, evolutions of HFSMs and can handle reactions in multiple different ways, from simply authoring the tree in a way that handling of external stimuli is higher priority and can cause other parts of the tree to bail out as necessary, to more elegant solutions that can inject new behaviors dynamically at certain points of the tree (Isla 05).
- Automated planners, such as GOAP or HTNs, could require a partial or full re-plan. These scale better (Orkin 06), but reactions will be less controllable in this situation, as the algorithm can be choosing different solutions with the same input event, depending on the current state of the world and what the goal of the NPC is.
- Some games also build solutions separate from their main decision-making system to control reactions. An example of such approach could be a rule-based system (Couvidou 17), (Blouin-Payer 2017).

Reactions are used to communicate state changes to the player, and although some of this information can also be reported through UI notifications, there is an expectation in modern AAA games of getting this information directly from the agents' behaviors and animations, so getting them right is key (Mononen 08).

4 Our Objectives

Our first objective when playing a reaction is *to always provide feedback to high-impact player actions*. In other words, if the player performs an action and is expecting a reaction from the AI, we need to match their expectations.

Our second goal is to build reactions that are *contextually appropriate*. Our agents must show behaviors that are believable and appropriate for the situation. We should not have NPCs dive out of the way when the player whistles at them (overreact) and we should not have an AI ignore a car that is about to run it over (underreact).

Our last goal is for our reactions always to be *readable* and *consistent*. Players need to learn how to play with our gameplay systems, and that includes the AI. This is an especially important point in stealth games, where players often have multiple ways to toy with the perception of the NPCs, and their predictable reactions are a big part of the gameplay expectations – we can have multiple versions of the same reaction, but the type of reaction (i.e. category and intensity) cannot change.

5 Suspension of Disbelief

It is quite common to hear the expression “suspension of disbelief” when we talk about

game AI. This idea was introduced back in 1817 by Samuel Taylor Coleridge (Coleridge 1817), when he talked about “*the willingness of readers to accept the implausibility of a story, so long as there was sufficient truth to engage their attention*”.

In a video game, we can define “suspension of disbelief” as the disposition of the player to accept our rules and get immersed in the world we are creating. As developers, it also represents our contract with our players – we should avoid breaking these rules and shattering the illusion. Reactions play an important role in this quest.

The objectives we presented in the previous section were aimed at achieving this preservation of the player’s immersion. For example, a lack of reaction violates the objective of “always providing feedback to player’s actions”; the player can interpret this as broken or unresponsive AI, and start seeing behind the curtain in our game of smoke and mirrors. Similarly, inappropriate reactions can look silly and out of context, or just buggy.

We also talked about readability and consistency as important qualities for our reactions. This is because our brains are extremely good at pattern recognition (Mattson 14). We want our players to have a high-level understanding of our systems and build certain expectations regarding the responses our agents will manifest to the players’ actions. But this also means our brains will easily spot repetition – and this is not a desirable trait for video game reactions.

6 Mitigating Repetition

In development, we are oftentimes limited by budget or time constraints. This mainly means that we can only afford to solve a finite number of problems and try to minimize unplanned, less polished situations as best we can. Applied to reactions, this means we can probably only build a handful of specialized reactions for certain important or frequent stimuli, while many others will be more generic by necessity.

Now, this is not strictly a bad thing. Going back to the discussion about repetition in video game AI, it is important to note that the more specific or exotic a reaction is, the easier it will be for players to notice it when it happens a second time, a third time, and so on. Soon they will see it every time, and the suspension of disbelief will be broken.

6.1 Voice Repetition

Think about the following case. Let us say we are playing a stealth game, and the player is spotted near a water well. An agent shouts “he is BY THE WELL” and combat breaks out. The player hides but is spotted again at the same location almost right away. “He is BY THE WELL”, the AI repeats. “BY THE WELL”. Although the fact that our AI is calling out a specific location will most likely surprise the player and be noted as good AI (Orkin 15), the perception of the player will quickly turn into annoyance when things get repeated over and over again.

A solution for this problem could come with the addition of cooldowns, or somehow remembering that certain “cool moments” have already recently been played, so it is better to hold off and use a generic response instead. More general call-outs, such as “OVER THERE” or “I SEE HIM” are less specific, and therefore less memorable. Building this type

of system and working with your writers can lead to great results (Ruskin 12).

6.2 Repetition in Groups

Another problem we can face is related to group reactions. If multiple AI agents are on screen (the closer they are, the more noticeable this problem will be) and all play the same reaction simultaneously, we are giving players another reason to break their immersion and just spot something that simply “does not look good”. Going back to our stealth game example, we have a group of two NPCs that are unaware of the player. The player then decides to engage our group and starts shooting around the agents, which triggers a “shots fired” reaction. At that point, both our agents play a reaction animation and transition into a combat state. There are two problems with this picture: they reacted the same way, and they reacted at the same exact time.

For asset repetition, we could tackle the problem simply by brute force (i.e. add variations to our animations, which is always a good idea). But, again, in a world ruled by budget constraints, this may not always be possible. Different approaches can be taken in this case, such as limiting the number of agents that can present a behavior or reaction at the same time, mix different types of appropriate reactions (e.g. have some of the NPCs run away instead of playing an animation) or some other procedural animation techniques (Shroff 2015).

For timing, slight time delays go a long way toward mitigating the ‘simultaneous reaction’ problem. In general, if all of the NPCs in our group react simultaneously, on the exact same frame (even if each of them is using a different, unique animation), something will feel off to the player. Humans have different reaction times – generally between 0.2 and 0.4 seconds (Rabin 15) –, so simultaneous responses will look robotic and unrealistic. To mitigate this problem, we can add some randomization to our reaction times – this will make our group reactions feel more natural.

7 Hit Reactions

Hit reactions are those played as a result of direct contact (or collision) with some other entity or object (e.g. being punched or hit by a bullet). Virtually every game uses this type of reactions in some form. They are intended to provide immediate feedback so the player knows that an attack has connected. The magnitude of the reaction is normally dictated by a combination of factors such as the force and direction of the impact, and the location of the hit (e.g. headshot vs. shooting a limb). Bad hit reactions can easily break the suspension of disbelief. For example, if we play a very painful hit reaction and immediately after our character goes back to being happy and without showing any sign of damage, this could be perceived as inconsistent or implausible. Maintaining some sort of memory, such as a damage state, could be a good way to solve some of these issues. Good hit reactions can be crucial for gameplay (Campbell 18).

8 Perception Reactions

A perception system is a set of simulated senses that allow AI agents to detect changes in their environment. Most games use “sight” as the main way for the AI to gather information; this sense is usually implemented as a combination of volume checks and raycasts that help the agents determine if an entity is in their field of view and needs to be recognized. “Hearing” is also commonly used; it can be implemented as some type of distance check from the source of the noise to the NPC. This topic is complex and outside of the scope of this chapter, but many good resources on the subject, such as (Walsh 15) and (Welsh 15).

A perception reaction is the consequence of an event captured by the AI’s perception systems (e.g. seeing the player or hearing a loud noise). Having a strong set of these reactions is key if we want to build good, responsive, believable AI, and there are some basic types that every good game AI should have.

8.1 Combat State Transitions

The lifetime of an agent in a video game can usually be represented (at a very high-level) as a state machine with a very limited number of states. For example, the typical action game could be depicted, at a very high level, as an FSM with four states – PRECOMBAT, COMBAT, SEARCH and DEATH (note: these states can be split further into smaller parts, but this is not in the scope of this chapter). The structure of this state machine is shown in Figure 1.

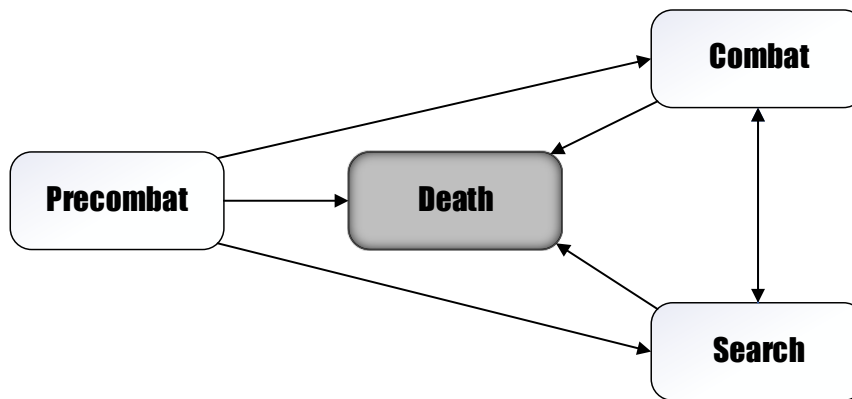


Figure 1 Basic high-level structure of an AI agent’s brain in an action video game.

Transitions between any of these states should be communicated to the player, as they are a big event in the lifetime of our NPCs. Players need to get clear feedback about these changes.

8.2 Player Sighting

In our high-level state machine, any transition into the Combat state is an interesting one. These transitions happen when the AI becomes fully aware of the player (enemy) and it is ready to engage, and we definitely should build a different reaction for each of the transitions that take us into Combat (e.g. Precombat → Combat; or Search → Combat).

There is, however, an even more special transition into combat, which happens when the player just pops in front of an AI at a very short range. This should cause an almost immediate change of awareness, where the agents become fully aware of the player almost instantly. It is important to differentiate this particular case from just regular detection – it will make perception feel much more realistic and snappy.

The agents' confidence in their knowledge of the player's whereabouts may also play into the type of reaction produced. If the AI is certain the player is at some location, but the player is spotted at a completely different position, the reaction should be different. NPCs that show surprise towards a player action are reinforcing the player's behavior (i.e. the game is acknowledging the player has bested it), enhancing the gameplay experience.

8.3 *Player Absence*

In action/stealth games, it is very common to avoid having AI agents that magically know the whereabouts of the player unless he/she is currently visible. Instead, the AI will create a "last-known position" model through the different perception stimuli it has received over time. When we have this type of system, there are two big reactions that we should consider building:

- *The player is located at a different location:* If the player is detected at a different location that is far enough from our original guess, the AI should acknowledge it has been outsmarted, by looking surprised. Some games forget this reaction, and the result on screen is an instantaneous change of orientation in the AIs to face the new position, but this looks very unnatural. If instead, the new location is very near to the old one, a special reaction is not really necessary. In fact, adding a big reaction could negatively affect the look of our AI.
Player detection updates can also inform what our locomotion should do: if the AI is moving toward the position (e.g. if it is trying to investigate it) and the position is updated and still in the same general area as the old LKP, all we should probably do is update the destination of our move command; on the other hand, if the new position is anywhere else (e.g. behind the moving agent), the AI should play a stop animation before restarting the movement.
- *Confusion:* If our agents decide to clear the position where they think the player is, but they discover that their belief was wrong (i.e. the player has moved away), it is common to simulate the notion of "confusion", or the degree to which the AI beliefs of the world have just proven false. In this case, we should once more acknowledge the player is "smarter than us", and clearly feedback that "we are going to start looking for him/her". This is actually the transition from "combat" into "search" in our high-level FSM.

8.4 *Tactical Failure*

To give the illusion of smart AI, we should try to find a way to detect situations in which players are taking advantage of an AI weakness, and react to them. Letting the player know "this is not going to work anymore", and modifying our agents' tactics accordingly will be a big boost to the believability and quality of our AI.

For example, let us say the player has a tool to lure AI agents to his or her position and keeps using this as a strategy to deal with all the agents in an area with low or no risk. We could “counteract” this on the AI side by changing the regular reaction and behavior of the third agent that is attracted by this tool – the NPC should acknowledge “something is not right” and change its strategy (e.g. calling for reinforcements or escalating the situation).

9 Conclusion

In this chapter, we have shown different ideas that can help readers create more engaging and human-like characters through reactions –building good ones can change the player’s perception of the quality of our games!

As a summary, the key takeaways are:

- Use smart barks that can convey a feeling of intelligence, but do not overuse them. Writers are your allies.
- Use animation variations as much as possible – this will avoid repetition problems.
- Randomize reaction times – this will make your agents’ responses look and feel more human.
- Match your players’ expectations when it comes to reactions – if a player is expecting a response from the game, make sure you have built one.

10 References

- [Blouin-Payer 17] Blouin-Payer, R. 2017. *Helping It All Emerge: Managing Crowd AI in ‘Watch Dogs 2’*. Game Developers Conference 2017.
<https://www.gdcvault.com/play/1024152/Helping-It-All-Emerge-Managing> (accessed April 2018)
- [Campbell 18] Campbell, J., Loudy, K. *Embracing Push Forward Combat in ‘DOOM’*. Game Developers Conference.
<https://www.gdcvault.com/play/1024940/Embracing-Push-Forward-Combat-in> (accessed April 2018)
- [Coleridge 1817] Coleridge, S. T. 1817. *Biographia Literaria*. Chapter XIV.
- [Couvidou 17] Couvidou, L., Sadoulet, X. 2017. *Taking Back What’s Ours: The AI of ‘Dishonored 2’*. Game Developers Conference 2017.
<https://www.gdcvault.com/play/1024159/Taking-Back-What-s-Ours> (accessed April 2018).
- [Isla 05] Isla, D. 2005. Handling Complexity in the Halo 2 AI. *In Proceedings of the Game Developers Conference*.
- [Mattson 14] Mattson, M. P. 2014. *Superior Pattern Processing is the Essence of the Evolved Human Brain*. *Frontiers in Neuroscience* 8, 265.

<http://doi.org/10.3389/fnins.2014.00265> (accessed January 2018)

[Mononen 08] Mononen, M. 2008. *The Structure of Action Game AI*. aiGameDev.com. <http://aigamedev.com/open/article/structure-action-game-ai/> (accessed April 2018)

[Ng 17] Ng, A. 2017. Artificial Intelligence is the new electricity. <https://www.youtube.com/watch?v=21EiKfQYZXc> (accessed January 2018).

[Orkin 06] Orkin, J. 2006. Three states and a plan: the AI of FEAR. In *Proceedings of the Game Developers Conference (vol. 2006, p. 4)*.

[Orkin 15] Orkin, J.. 2015. Combat Dialogue in FEAR: The Illusion of Communication. In *Game AI Pro 2: Collected Wisdom of Game AI Professionals*, ed. S. Rabin, Boca Raton, FL: A K Peters/CRC Press.

[Rabin 15] Rabin, S. 2015. Agent Reaction Time. In *Game AI Pro 2: Collected Wisdom of Game AI Professionals*, ed. S. Rabin, Boca Raton, FL: A K Peters/CRC Press.

[Ruskin 12] Ruskin, E. 2012. *AI-driven Dynamic Dialog through Fuzzy Pattern Matching. Empower Your Writers!* Game Developers Conference 2012. <https://www.gdcvault.com/play/1015528/AI-driven-Dynamic-Dialog-through> (accessed April 2018)

[Shroff 15] Shroff, J. 2015. Realizing NPCs: Animation and Behavior Control for Believable Characters. In *Game AI Pro 2: Collected Wisdom of Game AI Professionals*, ed. S. Rabin, Boca Raton, FL: A K Peters/CRC Press.

[Walsh 15] Walsh, M. 2015. Modeling Perception and Awareness in Tom Clancy's Splinter Cell Blacklist. In *Game AI Pro 2: Collected Wisdom of Game AI Professionals*, ed. S. Rabin, Boca Raton, FL: A K Peters/CRC Press.

[Welsh 15] Welsh, R. 2015. Crytek's Target Tracks Perception System. In *Game AI Pro 2: Collected Wisdom of Game AI Professionals*, ed. S. Rabin, Boca Raton, FL: A K Peters/CRC Press.