AI in video games is most noticeably used on the NPC’s or non-player characters that move around and make decisions based on events happening in the game. AI uses data structures these data structures are just different techniques for implementing AI or other types of programing into different languages. Focusing on AI there are different types of data structures some focusing on using graphs in a broad sense and others using a narrower type of graph called a tree. Graphs are used in AI is best explained when talking about graph theory this simply being that it is the easiest way of showing relationships between objects or in this case keeping track of each node available to the other nodes around it, nodes in programing are like different points on a graph or different floors on a building like going up or down a stair case if you start on flour 5 you need to get to floor 6 before you can get to floor 7 and you must go to floor 4 before you get to floor 3 and on floor 3 there may be room 1 and 2 but you need to be on floor 3 to access these rooms.

Graphs are most commonly used in AI programing not just because of the reason above but because they can easily show different types of relationships like the floors to rooms example above or it may be a set of books with chapters, in AI these would be different actions that the NPC may or may not take depending on the situation. but as stated there are multiple techniques for AI handling these techniques each have pros and cons to them these are things like finite state machines, decision trees, behavior tress or a pathfinding technique like A\* pronounced A star.

Finite state machines are probably the most commonly used technique available not in the game industry necessary but in an overall perspective to programing this is because state machines are easy to implement and are a way of easily packaging up code and using it when needed these packages are called states. These states are a good way of changing what’s happening on the screen such as showing text pausing make an enemy run towards the player when it sees them. Finite state machines use a graph rather than a tree because it needs to pick the states its using without needing to go through a different state a tree would need to pass through a different state before getting to the state needed which is why a graph is used instead in the building analogy this building needs an elevator instead of stairs so everything is accessible from the start. The pros of this is it’s easy to implement and easy to fix if something goes wrong with it the cons would be it cannot be used for anything to complex like pathfinding around objects without going through or crashing into them. Instead of a finite state machine a behavior tree can be used the pros of such being it would make things faster to a degree and the cons of which would be that it’s harder to implement then the state machine.

A\* pathfinding is used in video games as well path finding as in to find a path from one area to another like walking along a grid or avoiding things such as lava traps and walls this also includes selecting the shortest or best path for the NPC to get where it needs to go or when an event happens, in some cases like league of legends finding this path for the players after they select a place to move to or a creature to attack this uses is a tree instead of a graph for the exact opposite reason that a finite state machine is a graph rather than a tree if it was a graph the character could teleport or jump from one location to another without any hindrance and this would not be very realistic or useful for the purpose it fulfills it needs to move as if It is walking and that means going from one node, point or area to the one next to it until it reaches its destination.

The pros for A\* are its very useful for pathfinding and adding it to creatures or object so they can find the best solution to an obstacle or another object the cons are if something goes wrong it’s not as easy to fix as other techniques and its more complicated as well. A\* could use maybe another technique although no other technique can get the result that A\* produces and a lot of pathfinding techniques use A\* which would make changing it to something else redundant since its being used anyway.

To go over quickly graphs are used because they can show different types of relationships between objects such as buildings and floors, books and chapters or actions a NPC may take. Finite state machines are useful for using an action when its needed without having to go through a different action a behavior tree could also be used for this but its more complicated to use. A\* is for creating paths from one area to another for things such as movement when you need to go through one area before getting to the next it really shouldn’t be replaced with anything but a different pathfinding technique which may or may not use A\* anyway. These techniques are used in the game industry because they are practical and useful for the purpose they serve an ultimately it is the programmers decision on which one to use for the situation needed.