

Matt Mitchell

David Pennenga

John Formica

Squad Tactics

- Things to consider with regard to correct movement in AI;
 - Levels of Threat
 - The more targets in the player's line of sight (LOS) while the NPC is in the player's LOS, the greater the possibility of obstructing the player's LOS. There are several ways to fix this.
 - If the player is looking in the direction of the NPC, the NPC should move to the angle of the player's LOS which has the least amount of targets.
 - If the player is looking away from the NPC, the player should move in the opposite direction that the player is looking.
 - If the above prove insufficient, prioritize which targets to obstruct with threat level assignments. A player is more likely to target enemies with greater levels of threat so the NPC should obstruct the player's view of enemies of lesser threat levels. Similarly, the player is more likely to target groups of enemies over individuals so the NPC should opt for obstructing the view of smaller groups or individuals. Threat level should also be considered when using the latter heuristic.
 - Straying from the team

- NPCs should stay close to the team by referencing a preset maximum radius allowed from the team.
 - Reasons for this are so that the NPCs do not alert enemy combatants, distract the player from the game (such as the player trying to find lost teammates), or fail to assist the player in combat.
 - Use of caution
 - Teammates should avoid risking the welfare of the team. So first of all, the NPC should not be the first to engage an enemy, explore a room, move over a hill, or move around a corner. If an NPC acts first in these circumstances, it may undermine the player's play style.
 - Team context
 - The NPCs should take different play styles of the player into consideration. There are different ways to approach this but one example is to allow the player to manually set play styles.
 - Such as in Star Wars: Republic Commando, the player can set stances such as "search and destroy," "form up," or "hold position." Each one of these stances corresponds to different NPC default behavior.
 - The NPC should also move at the same pace as the player. However, the NPC should not obstruct the player's LOS but at the same time consider tactically sound behavior.
- Things to consider with regards to correct behavior in AI
 - Use of cover

- The NPC should incorporate preference for cover into its pathfinding algorithm. This design will add to the realism of the game and give the player the impression of tactical prowess in the NPCs. This would also prevent accidentally alerting enemies of the squad's movement.
- However, the NPCs should take player cues if the NPC's behavior does not match the player's.
 - For example, the NPC should consider the speed of the player in the event that the player seems to be showing disregard for staying hidden.
- Selective Firing
 - The NPC should only fire when within range and in view of the target. Also while firing, the NPC should use controlled bursts to preserve accuracy and conserve ammunition.
 - The NPC should also consider reaction times which should be manipulated if the individual combat-related AI is too powerful. To balance an AI's high accuracy, the programmer can set low reaction times. However, the context of the game should affect how reaction times should be set. In a Rainbow Six style game, where the NPCs are supposed to mimic the combat prowess of Navy Seals, reaction time should be fast and accuracy should be high.
 - Care should still be taken to make sure the team's reaction time is not lower than the enemy team's reaction time because this will result in more probable losses for the player's team.

- Reloading
 - When a group of NPCs are attacking, there should be a natural cycling of attacking NPCs that occurs. When NPCs reload they should fall back or take cover and then return when finished.
 - Additionally, NPCs should stagger their firing so that the group does not collectively run out of ammunition at the same time. This design will most likely ensure that the player will always have supporting fire.
- Things to consider with regards to supporting the player in AI
 - Reporting to the player
 - The NPCs should protect the player's sides.
 - This means that the NPCs should report threats via audio or visual cues to the player. Messages are the staple of squad game play but it is also important not to overwhelm the player with messages.
 - For example, reporting threats while in combat is not usually necessary.
 - NPCs should also pick appropriate engagement times for fighting enemies.
 - For example, the NPC should not attack an enemy while the friendly squad is already in a fire fight because this could cause overwhelming opposition.
 - Selecting a target
 - The NPC should choose targets other than the player's. This can be accomplished by choosing targets farthest from the player's LOS.

- NPCs should also take threat level of enemies into account because the player might need assistance. This judgment call can be made when the player's target's health is greater than the player's health.
 - Considering higher concentrations of targets also could help prioritize targets.
- The player is most important
 - When selecting a weapon, the NPC's weapon should not be more powerful than the player's weapon unless dictated by the player.
 - In weapon and item selection, the player should be allowed the first choice. This can be implemented by tracking the player's proximity to items. So if the player approaches items and then walks away, the NPC may then pick up the items.
 - Reaction time of the NPC's should also allow for the player to attack enemies first unless the player is otherwise distracted such as engaging another target in combat. In the event of the player being distracted, the NPC may attack a target.
- Implementation of putting the player first
 - One can implement a line of sight avoidance hierarchy. In this tree, the child agent avoids the line of sight of the parent agent. This design allows for simple restricting in the event of casualties because one can simply implement a node replacement algorithm, such as always assigning precedence to certain child node.

- In the case of threat awareness, an NPC can use a line of sight algorithm which upon scanning a target, for example, it updates all desired information on that target. More specifically, an NPC can do things such as constantly “looking around” so that the environment is being constantly updated. This method was visible in many team’s Robocode this semester.
 - While “looking around” the NPC should also update information about inanimate objects such as cover, restricted areas, or team locale.
- Algorithms such as “looking around” could feed into a team knowledge base which can govern NPC behavior. Such a system would allow NPCs to reference it for easy access to collective group knowledge.
- In finding available NPCs for assistance of a player or fellow NPC, one can use an algorithm such as:
 - Availability = $(1+N)(1+O)(1+P)+(Q*\text{[infinity]})$
 - Where...
 - N= # enemies in covering area (integer value)
 - O= # enemies within range (integer value)
 - P= # enemies threatening team (integer value)
 - Q= supporting another teammate (boolean value)
 - This formula assigns a finite, comparable value to all NPCs which are not supporting another teammate. Because in the event that an NPC is referenced while assisting another teammate the Boolean value for Q would evaluate to 1 and

1*[infinity] is an infinite number and thus infinitely larger than any availability value of another NPC which is not supporting another teammate.

- Example of correct NPC behavior in Rainbow Six: Raven Shield:

<http://www.youtube.com/watch?v=-9f3cXBB-Nc>

- In this video one can see how the NPCs maintain constant awareness of the environment by constantly looking around and watching the rear and front of the team. Additionally the reaction times are altered so that they mimic elite soldier behavior. Note the “stick to the shadows” path-finding behavior discussed earlier. Moreover these NPCs maintain controlled fire and stay close to the team. It is also visible that there is some sort of line of sight avoidance hierarchy in the way that each successor in the formation avoids the line of sight of its predecessor.
- Notes on the topics of team AI and emergent behaviors of different approaches
 - One approach to team AI is the decentralized approach which creates a similar dynamic as a group of ants carrying a piece of food back to the nest. The ants do not obey a higher power in coordinating their behavior, they simply communicate amongst themselves to accomplish a common goal. Similarly, in the decentralized approach there is no commanding force and actions are instead a result of communication in messages.
 - Another approach is the centralized approach which is the same way that most militaries function: a commanding entity takes information from the subordinates and coordinates the behavior of the group.

- The decentralized approach
 - There are plenty of advantages to the decentralized approach. Some of these advantages include, but are not limited to:
 - The decentralized approach is an extension of the individual AI so it is easier to implement because all decisions are solved by the individual AI and sending messages.
 - This approach is also robustly handles situations because the programming of the individual AI handles problem solving on its own without the need for a comprehensive playbook of how to handle each situation.
 - One can also easily include different types of AI into the same group because there is no top-down design that expects certain AI in the group
 - This includes scripted AI that simply send messages when appropriate
 - There are also disadvantages which include:
 - The decentralized approach has weak autonomy and loose coordination because there is no playbook which can effectively tackle any situation.
 - There is also no account for individual strengths or weaknesses because each agent assumes anonymity of the others in the group and does not try to compensate for differences in individual AI
- Squad Maneuvers

- In a tightly coordinated squad maneuver, the members rely on detailed, repeatedly rehearsed drills, and a continuous flow of information.
 - In other words, events are anticipated and coded specifically to be solved.
- In a loosely coordinated squad maneuver, the members do not rely on planned maneuvers and rely more on messages to accomplish goals
- The decentralized approach has much more emergent behavior from which to benefit.
 - Fire and maneuver behavior is the result of communicating “turns” at attacking within the squad so that there is a cycling of attacking NPCs.
 - Staying close to cover is created when the group communicates the whereabouts of nearby cover.
 - Preventing blocking of the LOS of others is done by communicating the direction of each NPC and players’ LOS.
 - Taking weapon capabilities into account again is the result of communication within the squad the abilities of each NPC and player.
 - The group will also appear to maintain group cohesion by communicating locations and intentions to the rest of the squad.
 - However the squad will also spread out when necessary by communicating distance from other NPCs.
 - The group will also appear to keep their collective line of sight on a target by alerting other teammates when a target will enter another teammates line of sight.

- Each squad member should maintain a “mental picture” of its environment.
 - For every other member of the team, the agent should remember the current position, activity, claimed destination, and LOS.
 - For every opponent, the agent should remember last known position and state, estimated current position, members engaging that opponent, members able to observe that opponent, and the LOS of that opponent.
 - For other hazards and threats the agent should remember the known and estimate position of the object and the damage radius if applicable.
- Messages are imperative to squad AI.
 - This is because they allow for:
 - The modeling of communication latency by queuing messages, the ability to present messages in game play, the ability to filter messages by assigning priority to the information in the messages, the transmission of messages to dead members who are assumed alive to add realism to the game, the ability to implement scripted AI in a squad, and the ability to accommodate human members into the squad.
- Example of implementing the decentralized approach: Ambush!
 - Goals of carrying out an ambush include things like waiting for an enemy to enter a kill zone then pull back to a predefined rally point after engagement as long as the team is not discovered.
 - If the team is discovered before being able to carry out the ambush, the team should fall back.

- While the decentralized approach is able to effectively return fire and prefer being near a rally point, it is unable to reach unanimous behavior on attacking because each agent will carry out its own behavior. Moreover the squad cannot orchestrate movement well because there is no central executive force to create a consensus of behavior.
- Planned maneuvers and the centralized approach to AI
 - There are two notable types of the central AI which are visible even in our own world.
 - The Authoritarian command style is one where a commanding force gives orders to subordinates and the orders cannot be ignored.
 - The Coaching command style is less demanding than the Authoritarian command style in that commands are more like suggestions to the subordinates. If the subordinate deems another task to be of higher priority than the orders, it may queue the orders to be carried out at a later time.
 - However one problem with the coaching command style is that agents may want to always perform their own behavior which could undermine the centralized approach all together.
 - The squad must consider its relation with the world around it in many respects.

- For example, the force ratio, which is the ratio of allies to enemies on the battle field, plays an important reference in squad tactics and could govern behavior such as advance flank, hold position, or fall back.
 - Another aspect to track is the line of fire ratio, which is the ratio of the number of friendlies in the line of fire of enemies to the number of enemies in the friendly team's line of fire. This is important because it could be used in considering tactical advantages.
 - In general, a team should maintain a first order logic knowledge base of Boolean states, much like what was spoken of in the decentralized approach so that judgments are made with low spatial and time complexity.
- Maneuvering plays an important role in effectiveness of the squad and therefore should be carried out in an equally efficient way
 - Even after a command is carried out to maneuver the squad, progress should be monitored. When individuals complete the task, they should inform the squad. That way they are able to receive another command earlier than the rest of squad so that no member of the squad is idle for any time.
 - Advantages to this approach include the ability to alter the squad's behavior mid-maneuver. So in the example of a member being trapped in a spot which forces a reevaluation of the path finding algorithm, the

central executive force or squad aggregate knowledge base could advise the NPC to act differently.

- Similarly, the squad cannot simply pull back in a haphazard way.
 - The squad should use resources such as maps or an alternative to the A* function to evaluate the most rational behavior.
 - In the case of an A* like function, the cost functions should consider things like the nodes that can be fired at from the preceding nodes, nodes that provide insufficient space to bypass, and nodes that are a bad position from which to fight. All of the above types of nodes should be assigned values which make them less desirable to the A* function.
 - However because of the time complexity of a function like this, it must be applied the squad's movement as a whole instead to individuals within the squad.
 - Moreover the algorithm should mark each position just before a bend in the path that blocks the LOS of many preceding positions on the path. This allows the algorithm to easily reference it if the need to backtrack arises.
 - There should also be a preference for locations with sufficient room for members to bypass it, this would mean the need to backtrack could arise and a new location searched.

- The chosen position should additionally be a good position from which to fire so the algorithm should favor a rocky hillside over an open field, for example.
- Command hierarchies play an invaluable role in the centralized approach.
 - The idea of a command hierarchy is heavily influenced by the idea of modularity yielding simplicity. Each level is only responsible for the input from its immediate subordinates. In the chain of command from commanders to captains to sergeant to soldiers with soldiers being the lowest on the hierarchy, commanders manage only input from captains, for example.
 - An example of how the centralized approach solves problems: outflanking maneuvers.
 - The strategic decision process for this problem is to first check if maneuver is possible or appropriate (e.g. if doors are accessible), and then compare distance between the opponent and the door with distance between the soldier and the door so that position assignments are efficient and quick to execute.
 - The maneuver should also be organized such that it is carried out in a realistic and rational way.
 - It should find how many soldiers are needed to cover the exits, order the soldiers to cover the designated positions, decide which room to search, order soldiers to move to

reachable doors of the first room, and when all else has been completed send a search command to the squad.

- Formations are an important part of group AI and adding to realistic qualities of a game
 - There are many types of formations like Lines, left and right flanks, column, box, wedge, and vee, just to name a few
 - Facing in these formations of course have a strong influence in the effectiveness of the formation in achieving its designated goal whether it be achieving a defensive or offensive maneuver, maintain wide lines of sight, decreasing friendly fire, or projecting more firepower.
 - A staggered line is one example of how facing and unit placement can aid in goals like projecting more firepower.
 - Outward facing in the case of the box formation is more of a defensive maneuver whereas inward facing is more of an offensive maneuver in the case of flanking an opponent from all sides.
 - Ordering of these formations also plays an important role in the formation's effectiveness.
 - Mix unit type ordering is the incorporation of different types of units into one formation. However the placement of these units makes great strategic difference.
 - For example, the weaker or longer-ranged units should be in the back of the line or the center of the box for protection

and the faster units should be placed at the flanks so that flanking maneuvers are carried out as quickly as possible.

- When sorting units, they should be sorted based on their minimum distance to the closest position and then placed in their formation location by iterating through the sorted list.
- Unit mobility is another quality of mix unit type formations that should be incorporated because telling one unit to turn and go somewhere may not be the best decision if that unit finds turning difficult.
 - Forming up should also avoid crossing the paths of units which can be achieved by implementing a movement vector and creating parallel paths to that movement vector so that the units never cross paths.
 - As units arrive at the rally point of the formation, units could be sorted in the order at which they arrive.
 - The size of the largest unit should also dictate the scaled distance between each unit in a formation.
 - Also, each formation should maintain a maximum occupancy so that when that maximum is reached, a new formation is made.
 - If certain places in the formation are reserved for specific units then restrictions should be coded and if no match is found in a group for that position, it should be left empty.

- When moving out, the AI should follow a few different variations in methodology.
 - The squad can move into formation before moving towards its destination but this might delay movement or cause backtracking of certain units which are already closer to the destination.
 - The squad can also move into formation just before the destination but this might give the player the impression that nothing is happening. The benefit of this style is that it allows for better coordination of attacks.
 - The squad could also move into formation upon reaching its destination. However, while this might result in faster movement and the gradual formation of units as they get closer to the destination, it leaves units not as effective while en route and not as organized upon arrival because of differences in arrival time.
 - Additionally, Group path-finding and movement should calculate the path for the leader and command others to apply the formation as a function of offsets to the path the leader is taking.
 - The flocking style where the group follows slightly behind the leader is also effective for formations such as the wedge.

- The only problem with some of these formations is that differences in locomotion of different units could break the formation's integrity and faster units might appear to move in slow motion or run in place as their animations struggle to match that of the slower units while moving at the same pace.
- In the event that the squad must react to an ambush, it should spread out to avoid the area of effect of a weapon or, in the event of an enemy's failure to ambush, converge to chase down an enemy
- After the group passes an obstacle or attack, it should regroup into the same formation that it was originally.
- One way of determining rational group behavior is with algorithms called influence vectors.
 - Influence vectors are algorithms capable of tactical assessments in a dynamic environment
 - These are also a modification of influence mapping where cells of influence are calculated outward from the units position because proximity often results in greater influence.
 - In this way, cells are calculated to have a value associated with agent and environmental qualities

such as containing allies versus enemies or weapons versus traps.

- Genetic programming is a form of adaptive programming which learns new behavior in real time
 - There are several advantages to this programming style such as creating more of a challenge for the player to defeat an NPC created with genetic programming. Furthermore learning algorithms allow for the discovery of new strategies which prove to be most effective.
 - Genetic program does have disadvantages and the most notable one is that developers cannot test what behaviors the NPCs will exhibit.
 - Genetic programming is structured so that it is a hierarchy within a hierarchy. Each node of one hierarchy corresponds to the root node of another hierarchy. The genetic programming of squad based AI allows the algorithm to select an NPC AI found within each sub-hierarchy until it proves to be less satisfactory and the algorithm backs out to select a different AI.
 - When appropriate, this style of programming has the ability to save nodes and construct new levels of the hierarchy around those saved nodes.

- For example, in the case of the fitness calculation method, the algorithm takes into consideration the game's duration and remaining health of the team and enemy agents.
 - Then the elitism method creates “m” copies of the best “n” chromosomes from each generation to insert into the next generation
 - There is also the roulette wheel selection which uses random selection factors like “mutation” and “cross-over” to select the nodes best suitable for carry over into the next generation.
- Finally, the video at the following link shows the capabilities of squad based AI in a decentralized approach and why this is such a powerful and effective method.
 - http://www.youtube.com/watch?feature=player_embedded&v=4ErEBkj_3PY