

Overview of your AI application. What is actually happening/what does it include.

My AI program is designed to have 3 entities onscreen. 2 enemies and a player.

The player has a shield that moves independently of the player, but inherits the player's position and moves with the player.

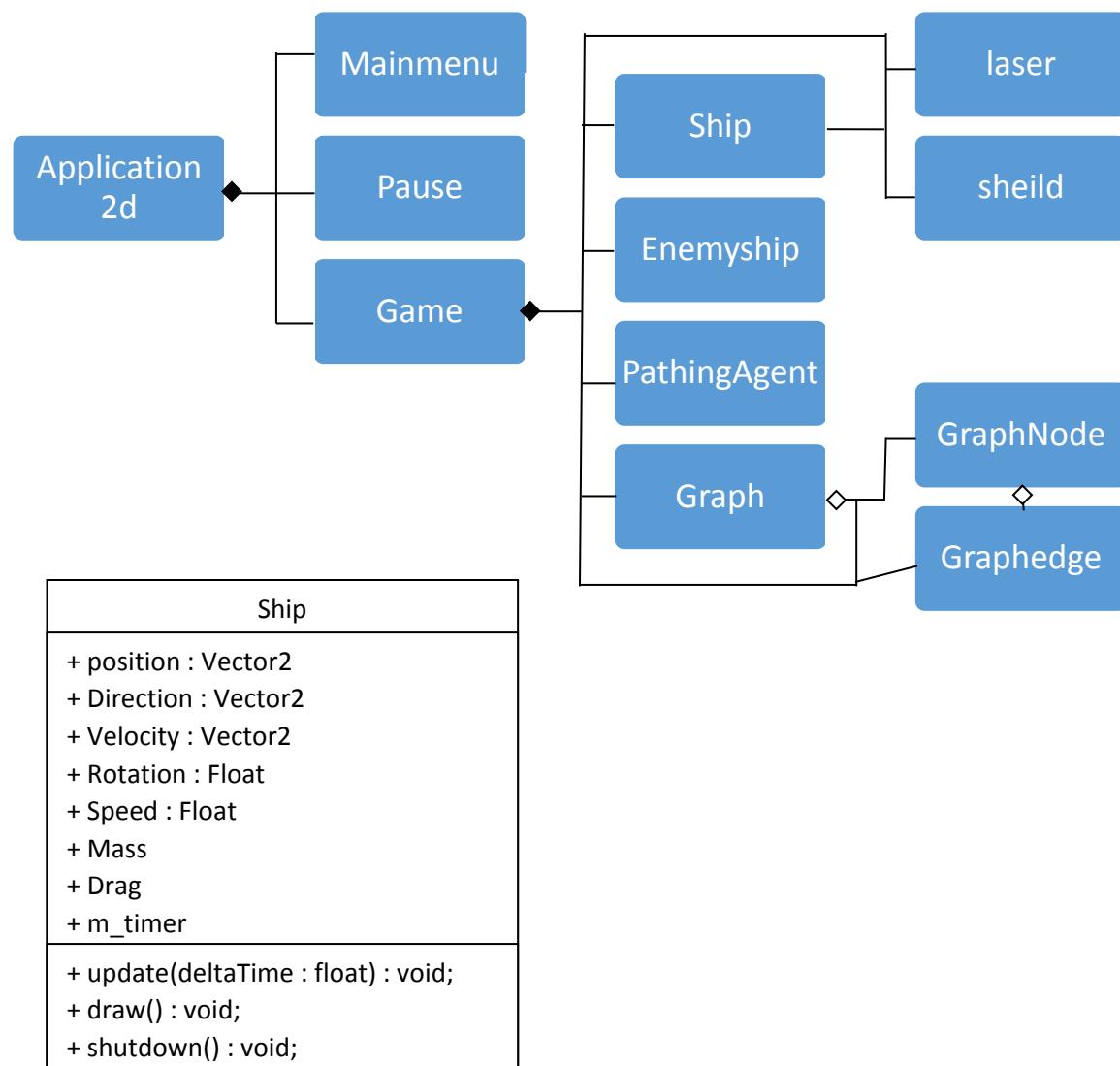
Player controls are standard W for up, A for left, S for down, D for right.

Further controls are Q to rotate anticlockwise, and E to rotate clockwise.

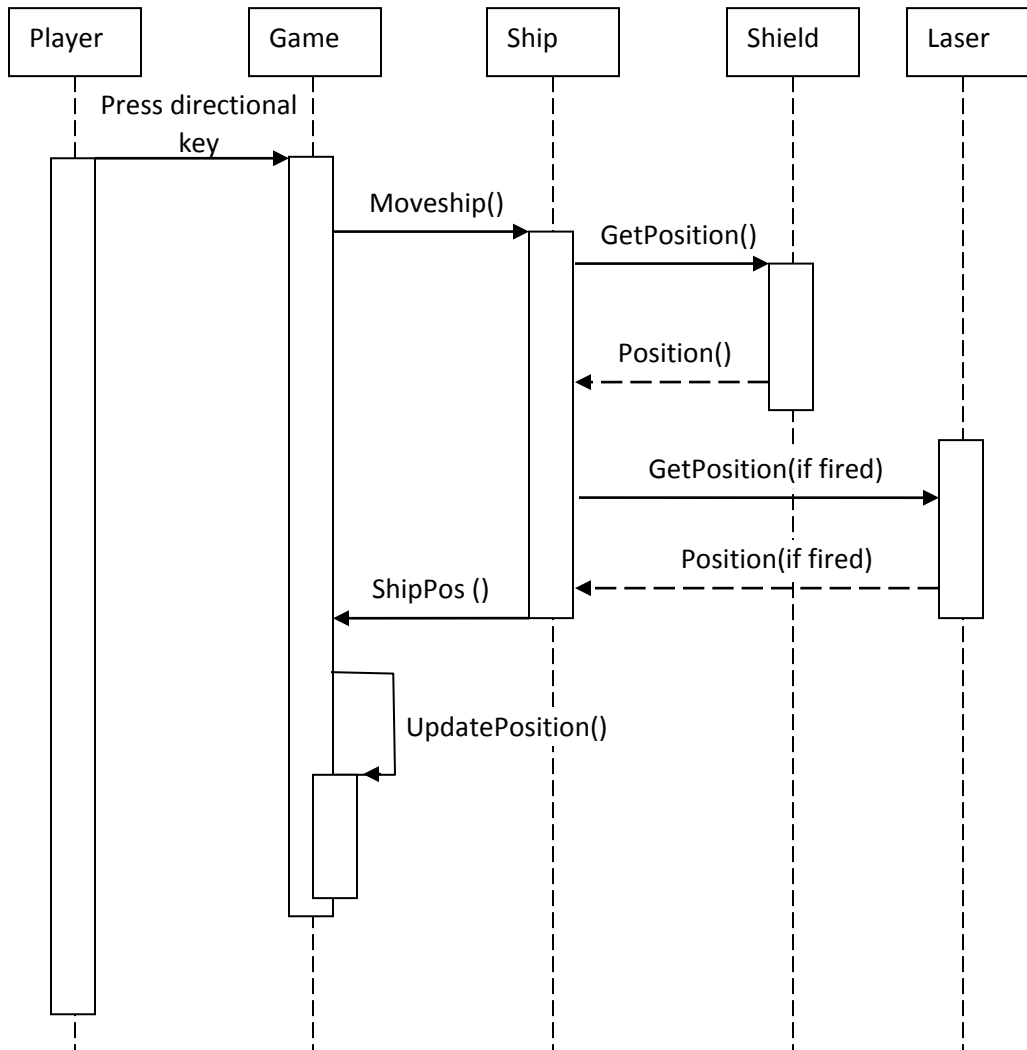
Shield controls are ↑ to move it away from the player ↓ to move towards the player, → to move to the right and ← to move to the left of the player (rotation will rotate both the player and the shield together, as one entity).

The two enemy ships are covered later in the AI Entity description

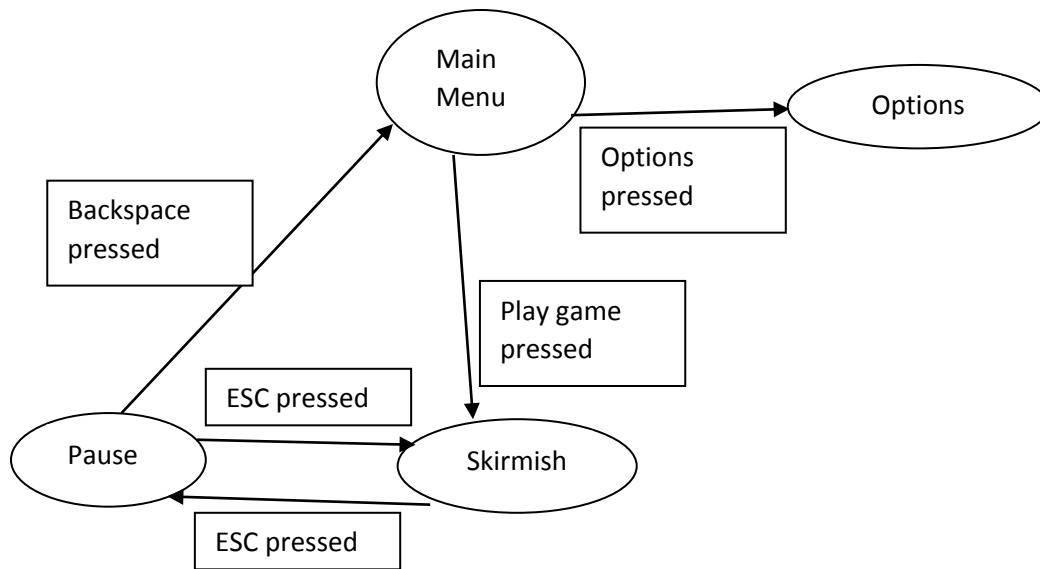
Class diagram of each of the C++ classes and their links (look at UML lectures on Canvas)



Sequence diagram describing a single interaction (eg. When the user hits the left key)



State diagram for your application (menu system) or Red Alert.

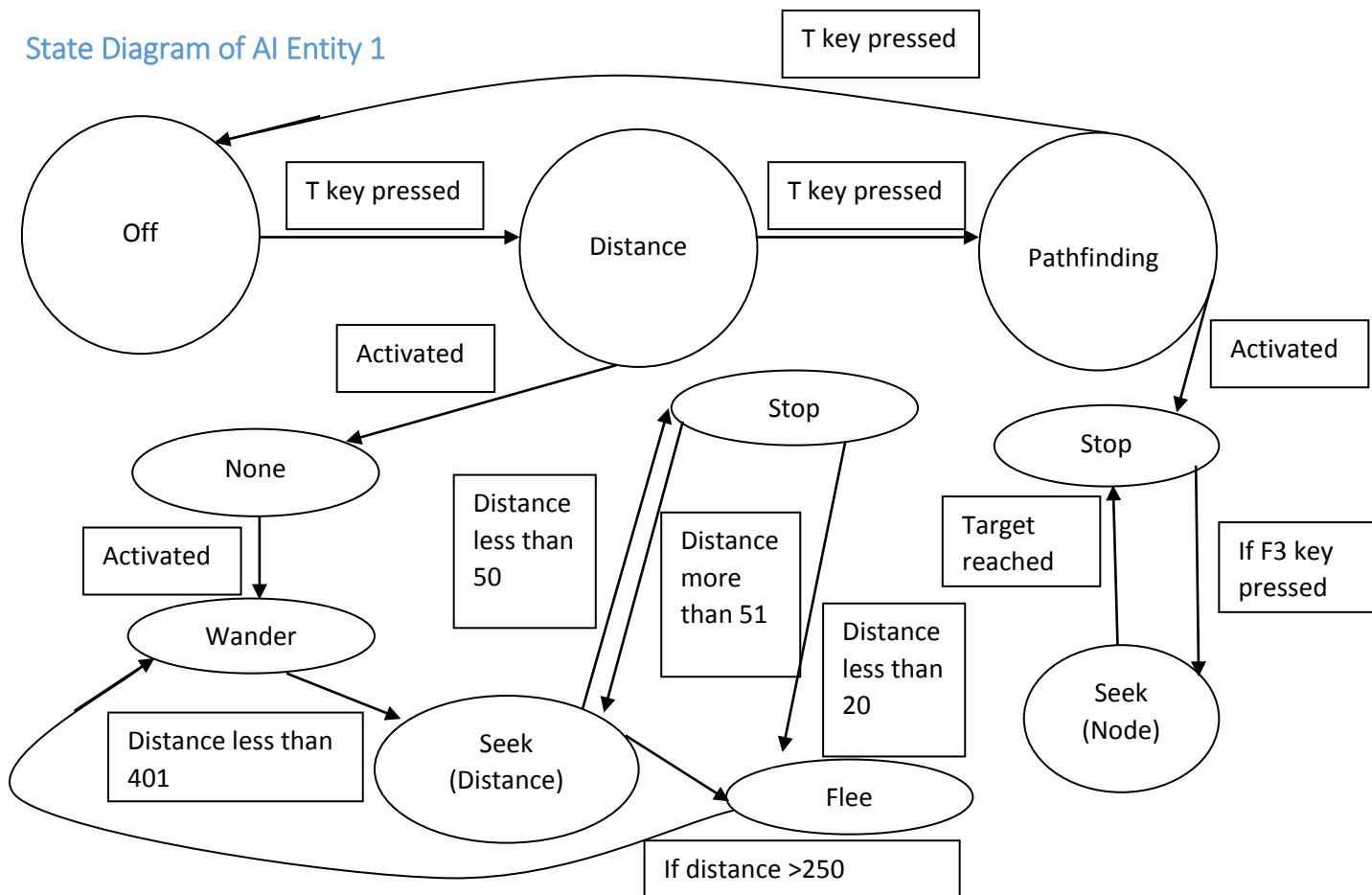


Description of AI Entity 1

The first AI entity (classified as "PathingAgent" in code, and displayed as the Yellow ship in game) has two modes; the first mode is AI Path finding, which uses Dijkstra algorithm to find a path to the player, and move towards it following that path.

The second mode is directional movement, to which the entity will switch game states dependant on where the player is in proximity to the entity.

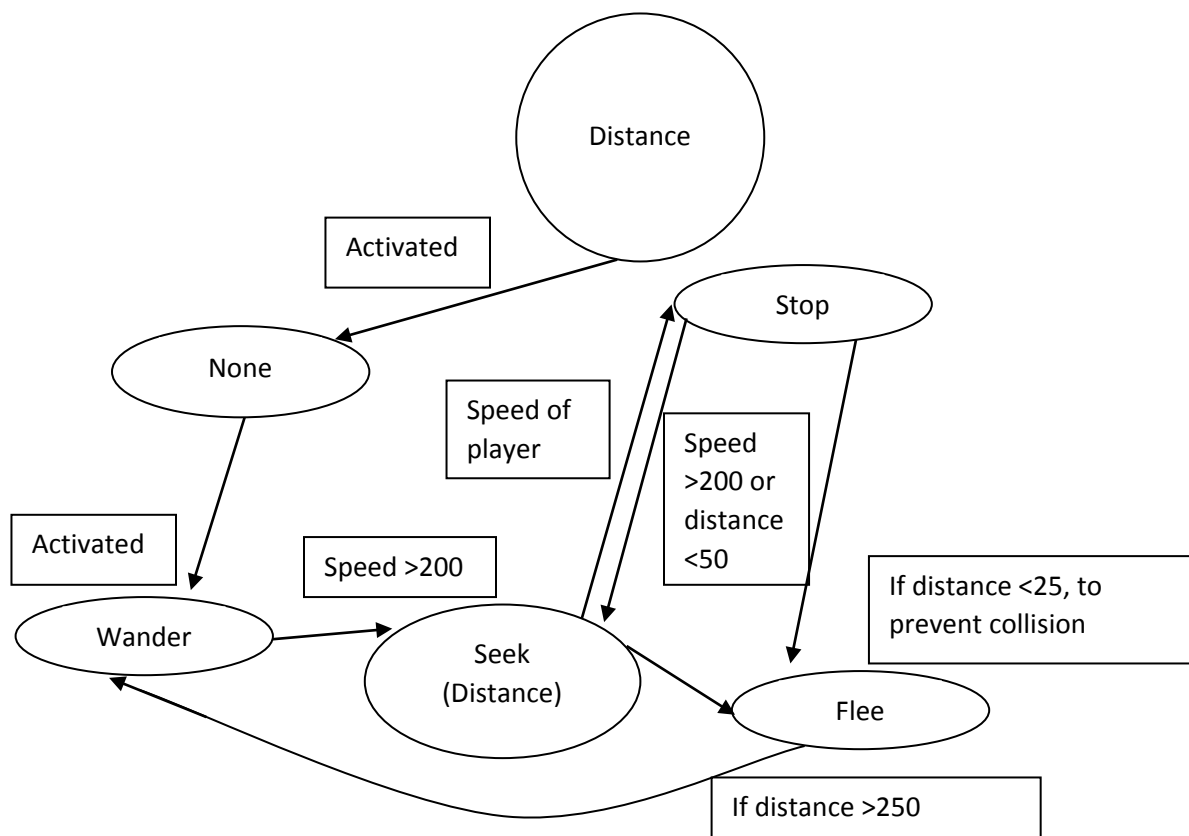
State Diagram of AI Entity 1



Description of AI Entity 2

The second entity only uses directional math and movement. However, this entity differs from the last one as it is triggered to follow the player on speed, when the calculated speed of the player is too fast, the entity will seek towards them. When the player slows down, and is a far enough distance from the entity, entity 2 will stop seeking towards the player.

State Diagram of AI Entity 2



Description of pathfinding (how is set up – talk about graph, how you generate the grid and which pathfinding algorithm you use and ‘why’).

Within the pathfinding code, I used dykstra’s algorithm. I used dykstra’s algorithm as I didn’t have enough time to use a*

I programed dykstra’s algorithm to use a Graph of nodes and node edges.

AI Evaluation

Known Bugs and issues:

- I had to use dykstra's algorithm as I didn't have time to implement A*
- Gamebugs
 - o PathingAgent
 - Pathfinding
 - Pathfinding is triggered on the press of f1 key, which turns the pathfinding on and off to lower to requirement to process it constantly. This causes the problem that the player can move from the calculated target but the pathingAgent will seek to the node that the player was on upon calculation.
 - If the PathingAgent (yellow ship) switches modes whilst a calculated path is still in effect, PathingAgent doesn't behave correctly. Instead of re-finding a new path, it will ignore everything else and navigate to the last node it was on, then continue the path to the target (this problem is in conjunction with the last one)
 - Wander...
 - Wander wasn't implemented (state was, actually moving wasn't)
 - For an unknown reason, the visual identifier (the ship becoming a white circle) flickers under specific conditions
 - This is caused by the states switching rapidly, but why it is switching without these conditions being met is unknown.
 - Stats
 - Does not have health stat
 - o Cannot die
 - Does not have attack stat (see laser)
 - o Cannot harm player
 - Laser
 - Does not have the ability to fire a laser
 - o Can not harm the player due to this (see stats -> attack)
 - Shield
 - Does not have a shield present
 - o Enemy ship
 - Stats
 - Does not have health stat
 - o Cannot die
 - Does not have attack stat (see laser)
 - o Cannot harm player
 - Laser
 - Does not have the ability to fire a laser
 - o Can not harm the player due to this (see stats -> attack)
 - Shield
 - Does not have a shield present

- Player related issues
 - laser
 - Laser does not spawn in the direction player is facing
 - Laser does not move in the direction the player was facing upon firing conditions
 - Multiple lasers cannot be fired.
 - Shield
 - Shield does not actual stop anything from hitting the player
 - Shield cannot be attacked thus cannot be destroyed.
 - Shield cannot be turned off and on

Improvements that could be made:

- Implementing A*:
 - A* would help the PathingAgent to get to the goal a lot more effectively, not going in random directions, but actually finding the fastest route to the target.
This could help should there be obstacles later in the development process should I choose to continue with it.
- Solutions to the bugs and problems indicated earlier (total: 24).

Possible Optimization in both code and design idea's:

The PathFinding activated every second, thus allowing the program to update the players location, and allowed the program to constantly seek towards the player. However, in doing so, was calculating a new path every frame. A solution I implemented to counteract a the problem was that the new path function would be called when a specific key was pressed, in this instance, being the "F1" key to observe the Graph and the graph nodes being used "F3" was used to start at x 0, y 0 and visualize the distance through green to white gradients. The problem with this method of path finding is that the player needs to run it every time for the algorithm to be initiated, and if the player is in movement, the path is only to the players position on the frame after F1 was pressed. A solution to that would be to implement a timer, that, upon each call or frame, the counter is incremented, when it hits a specific number, (or time elapsed) and the pathfinding is activated, and the mode is in pathfinding mode, and the PathingAgent entity is not at the player already, that it will run the calculate path function again. A suggestion for this is to run it every 5 seconds.