

The player object casts out several rays (cycle through each ray, doing one every x frames).

At each rays max distance OR when the ray hits some object, create an "Assistor Beacon".

Assistor Beacons are all unique:

Front

FrontL

FrontR

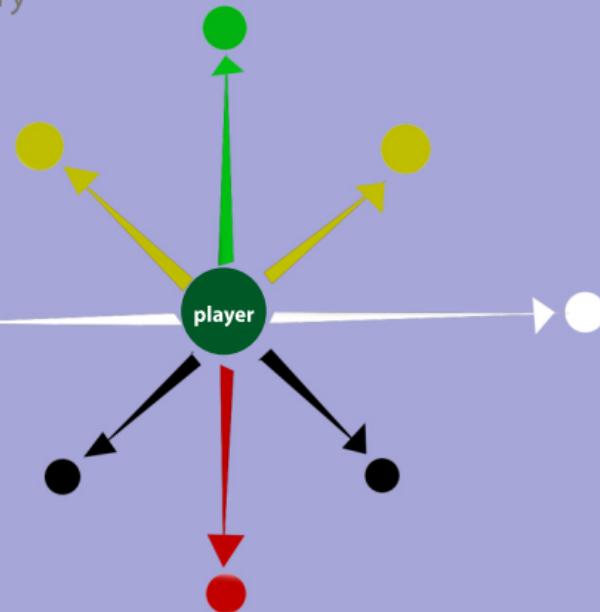
Right

Left

Back

BackL

BackR

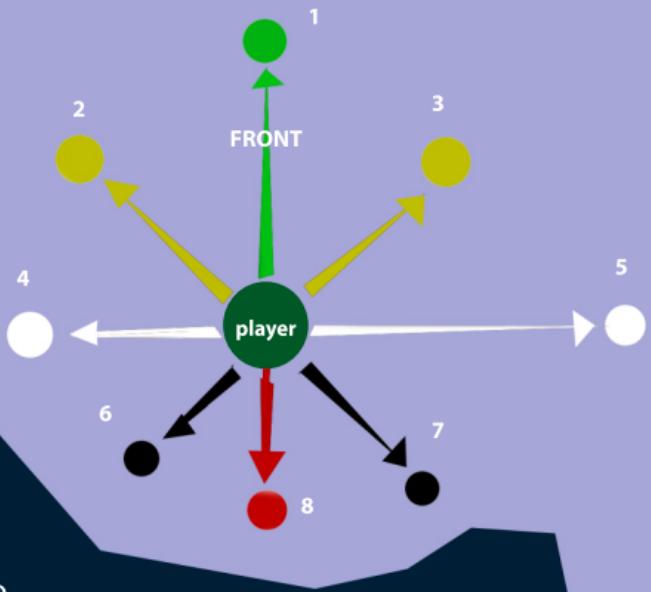


Essentially we create a 8 pronged feeler probe that is attached to the player

- we can use it to transmit player information to the AI so that they can be more intelligent in their approaches.

All we need is to create the state machine for probe compression conditions.

Some structure



PROBE ARRAY REPORT:

- 1: Unobstructed
- 2: Unobstructed
- 3: Unobstructed
- 4: Compressed to 40%
- 5: Unobstructed
- 6: Compressed to 30%
- 7: Compressed to 50%
- 8: Compressed to 50%

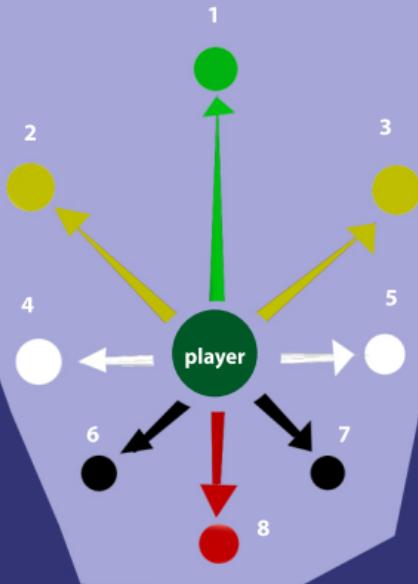
Processing Array information examples:
If 1 is unobstructed the player is looking
into an open area, do not approach from
#1.

Obstructed/Compressed probe rays reveal
that there is an obstacle that could allow
the AI to use on approach to remain
undetected.

#5 being unobstructed while #4 is not
compressed shows an open flank - attack
from this side! (Generate a line with X dist
from player through the probe, since
AI enemy to that location for Z seconds,
before approaching player.)

A very good example of how this data is useful for AI decisions.

Some obstacle



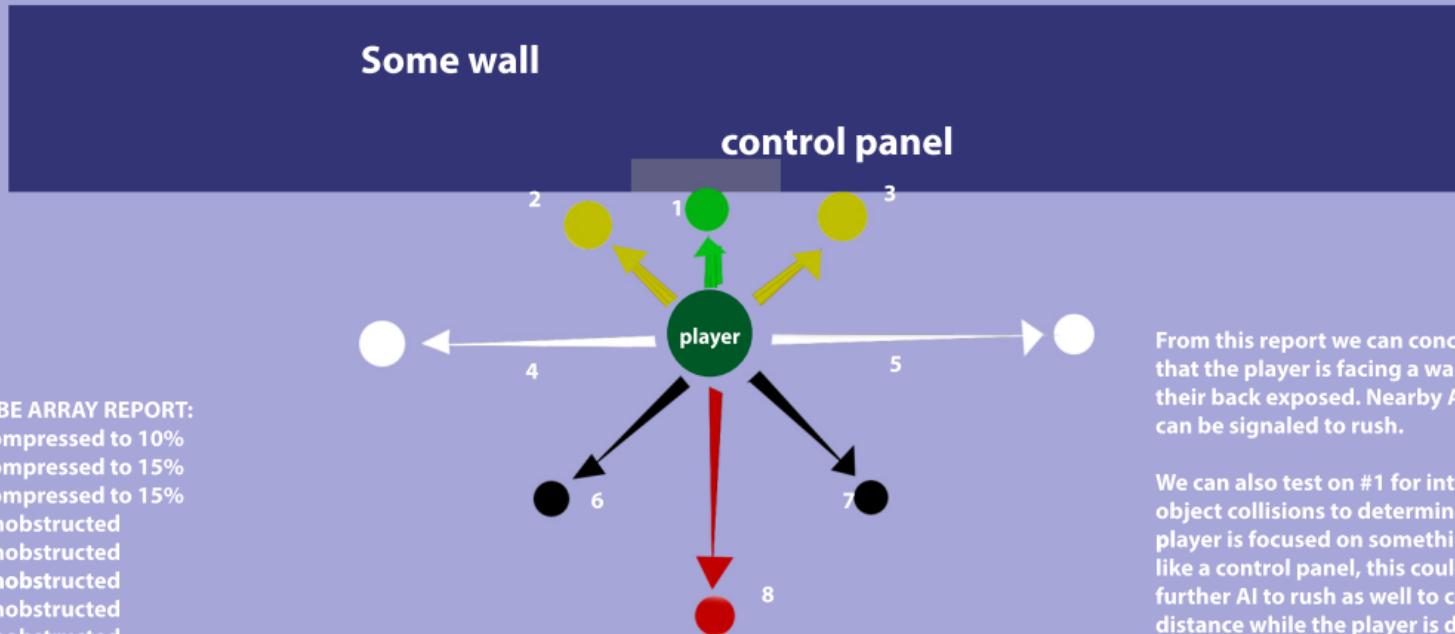
PROBE ARRAY REPORT:

- 1: Unobstructed
- 2: Unobstructed
- 3: Unobstructed
- 4: Compressed to 20%
- 5: Compressed to 20%
- 6: Compressed to 20%
- 7: Compressed to 20%
- 8: Compressed to 20%

We can use the data from this probe report to determine that the player is hiding in a nook, looking outward, most likely waiting from enemies to approach.

We can signal the enemy AI to do some behaviors like:

*Stop moving collectively and start howling and screaming until the player moves at least X distance from their hiding location, and then go silent and strike all at once.



From this report we can conclude that the player is facing a wall with their back exposed. Nearby AI can be signaled to rush.

We can also test on #1 for interactive object collisions to determine if the player is focused on something like a control panel, this could tell further AI to rush as well to cover more distance while the player is distracted.

Alternatively we could alert AI to sneak up on the player as quiet as possible if they are focused on something else.