## Stealth

(Wilde, 2012)

Creating an obstacle for the player to overcome. I will need to create a method that will both allow the player to know when they are and aren’t hidden as well as a way to allow stealth within the Unreal Engine.

For my stealth game I want Sound and Light to play a big roll in how they move through the environment. For Light I have come up with the following ideas on how they can be implemented.

## Collision Boxes

(Nicol, 2020)A picture containing graphical user interface

Description automatically generated

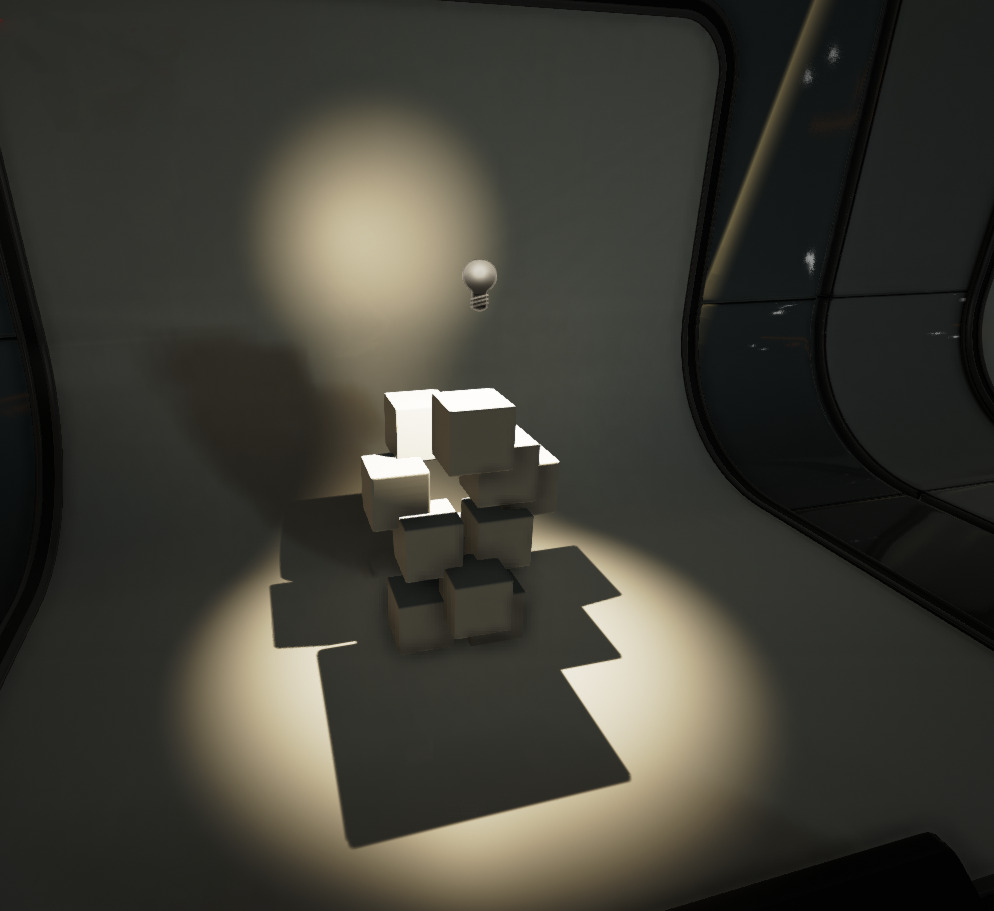
A simple way of adding in stealth within the game would be through using collision boxes around the environment to define where I want the player to be seen and invisible, however this method would only allow for hard defined spaces of visibility. It would also be very time consuming to place all the collision boxes everywhere. This method would be more usable within a box or bush hiding mechanic similar to Metal Gear Solid or Assassins Creed.

(Koshelev, 2017)

This method can be seen as a very inefficient method of creating a stealth system within the environment however it is a good way of creating safe zones for the player to use like the bushes within Assassins creed where unless the NPC walks into you, you’re practically invisible to everything.

So, using Collision boxes as a way to create stealth isn’t going to be useful to me for an overall stealth gameplay mechanic, however I can use it for creating usable stealth safe zones within the world for the player to use.

## Using Point Lights

(Point Lights, n.d.)

What you can do within the unreal engine is use existing light sources and their own light ray calculations to detect the player. From this we can get a either a true and false value or even a percentage-based value of the players current visibility.

A true or false setting for whether or not you’re currently visible or not is good enough for testing and then if time allows it a progression onto a percentage-based visibility method can be developed. Similar to the visibility system in the earlier Splinter Cell games.

(Jean, 2013)

## How can this be done in the engine?

One method of integrating point lights into the game within the engine is to have the player character blueprint having a event tick or a timer by event/function (which would allow for a less performance tanking) from this I can then have a for each loop from the array do a check for each light to see if they can see the player character through their AttenuationRadius, if this comes back as true, we can then have a line trace be spawned to check if the player is hiding behind a crate or box within the lights radius. We can also tell the line trace to ignore the player character and NPC’s. This will allow the line trace to make sure that the player character is in full view and not hiding.

Instead of having all this being calculated in the player through a tick event you can instead use the lights themselves do all the checking for the player. You can give each Point Light a collision box that’s the same size as their AttenuationRadius, from this collision box you can have it set of a timer by event which when it detects a player and from this a line trace can then be used to find and track the player within the lighting zone to check if they’re actually visible and not just behind cover. This method would allow for better performance by not having the check done continuously but only when needed.

# References

Jean, A. (2013, January 4). *Splinter Cell: Chaos Theory Retrospective*. Retrieved September 2020, 2020, from GameTactics: http://www.gametactics.com/2013/01/splinter-cell-chaos-theory-retrospective/

Koshelev, D. (2017, November 28). *assassins-creed-origins-review*. Retrieved September 28, 2020, from Root Nation: https://root-nation.com/en/games-en/games-review-en/en-assassins-creed-origins-review/

Nicol, H. (2020). *FYP Diagram.*

*Point Lights*. (n.d.). Retrieved September 28, 2020, from Unreal Engine.

Wilde, T. (2012, December 03). *stealth-game-design-mark-of-the-ninja*. Retrieved September 28, 2020, from PCGamer: https://www.pcgamer.com/stealth-game-design-mark-of-the-ninja/