

Problem Definition

The purpose of this project is to create a computer simulation and navigation algorithm to move a virtual spacecraft through a simulated asteroid field in a 3D environment. The asteroids will be modeled as Unreal Actor objects of different sizes traveling in random directions with random speeds, some of which may be in the spacecraft's path. In this virtual world, the spacecraft will have full asteroid awareness, allowing it to compute many navigation options that will help it to avoid hitting the asteroids. We plan to use Unreal Engine 5 for the visualizations and C++ and/or Unreal Blueprints for the programming [2].

Problem Solution

We plan to solve this problem by creating an Unreal level with Actor objects representing the spacecraft, asteroids, and waypoints. The spacecraft will use simple Unreal physics for the ship and the asteroids will spawn in certain spots moving at fixed speeds. The main challenge will be to calculate waypoint actors for the ship to follow so that it will not hit an asteroid. We believe we can start our algorithm development for making waypoint actor locations by limiting the possibilities of the locations to zones in space and time that the ship is capable of moving to, we'll then see if any asteroids occupy this space at this future time and then determine an area that the ship can pass through in order to avoid hitting the asteroid. Unreal Engine provides many useful features for this project such as object collision detection, vector math functions, and nice visuals. Additional work includes making adjustments to the spacecraft's pitch, yaw, roll, and thrust. This spacecraft will have to have a closed loop control of these aspects to get the spacecraft to move to the waypoints.

Progress

Our progress up to this point started with downloading a spacecraft model (with asteroids) from the Unreal marketplace. We created a basic Unreal project and added a spacecraft APawn object, an asteroid AActor object, and a waypoint AActor object [6]. Unreal makes C++ coding somewhat simpler because it uses C++'s object inheritance feature [4]. With object inheritance you simply choose an Unreal base class that best meets the characteristics of the type of object you want to place in your level. Using Unreal's vectors and rotators of the waypoint and asteroids, we have managed to set up a pitch and yaw error for the spacecraft to help the ship orient to the waypoint [6]. Finally, our spacecraft has collision components and a static mesh in the same way as the asteroids.

We put debug spheres around the asteroid and waypoint, so they are easier to see and lines that point toward the waypoint and asteroid.

Results

The results that we expect to get are that our spacecraft will be able to navigate through an asteroid field to its objective all by itself. This means that the spacecraft, using C++ and Blueprints will be able to find its own way through the asteroid field to achieve the objective. This may help us in the future to understand the way of coding and how to make a spacecraft navigate in the many years to come.

Sources

1. Dan's Game Dev Channel. (2023, December 29). Unreal Engine Tutorial Vectors 02: Position Vectors. YouTube. <https://www.youtube.com/watch?v=SLUCENNoBV8>
2. Forsythe, Alex. (2023, December 31). Blueprints vs. C++: How They Fit Together and Why You Should Use Both. YouTube. <https://www.youtube.com/watch?v=VMZftEVDuCE>
3. Forsythe, Alex. (2023, December 31). The Unreal Engine Game Framework: From int main() to BeginPlay. YouTube. <https://www.youtube.com/watch?v=laU2Hue-Apl>
4. Introduction to C++. (1997). Microsoft Corporation Document No. X03-09163.
5. UPIDController. (2023, December 27). Github. <https://github.com/robco-iai/UPIDController>
6. Zhenyu, George Li. (2023). Unreal Engine 5 Game Development with C++ Scripting. Birmingham: Packt Publishing Ltd.