Day 4

09/01/2023

Planning for unreal Engine based chess game started planned how to create different classes and and actor that need to be created and how to have a single pc two player game with changing camera angle for different player

coupling and cohesion basically we want to keep low coupling that is the degree of interdependence between the modules we want to minimize as much of their knowledge about each other and how they do things the lower the coupling the better the easier just manage it easier it is to extend the game or make changes to the game without breaking things

cohesion on the other hand is a degree to which the elements are functionally related so once we're given a blueprint a purpose and a reason and to find that well then everything that is inside that blueprint

should be specifically about what it is what its purpose is and not much more we do not want to have duplication or replication of functionality across blueprints we want to keep them pretty well defined within each based on what their set of concerns are and that is known as concern driven development or separations of concern is is the pattern that were trying to follow that is we're going to separate the out two concerns and make sure that the blueprints are only acting on the concerns that they
should be acting on .

The main concepts discussed are encapsulation, low coupling and high cohesion, which are all important for creating organized, reusable, and easy-to-understand code in Unreal Engine's Blueprint system. Encapsulation involves bundling all of the data and code that make up a class into one place and hiding the details of how the class works, exposing only a contract or API to other classes. Low coupling means that the modules in the codebase have a minimal amount of interdependence, while high cohesion means that the elements within a module are functionally related and have a clear purpose. The use of events, functions and macros in blueprint also discussed, events allows for capturing incoming data and facilitating communication between blueprints, functions are used to encapsulate functionality in a blueprint and make it reusable, and macros are used to encapsulate functionality in reusable blocks of functionality that can be reused and shared across different blueprints.

created several blueprints such as the player pawn blueprint, player controller blueprint, game mode blueprint, board blueprint, board square blueprint, and a parent chess piece blueprint. You've also added an "initialize" function to each one of these blueprints so that you can call them later in the development process. You're also considering configuring the project settings so that the player controller blueprint and the player pawn blueprint are the defaults, as well as setting the default map and game mode in the project settings.

risus. Nullam blandit quis est vitae feugiat.