Primeiramente, o documento de design de jogo eh muito importante para que tudo seja bem definido, desde coisas simples, como a historia, ate coisas mais complexas como requisitos tecnicos, ajuda a manter uma visao clara do jogo, facilitando a comunicacao entre todos os que trabalham para que o o projeto saia do papael, como os designers, programadores, artistas, produtores, garantido a consistencia durante seu desenvolvimento.

O material serve de apoio, de um guia de referencia.

Analisando os 3 documentos, penso que cada contem o conceito do jogo bem detalhadamente, no caso do DOOM, eh inovador no mundo de terror e acao, Valkior um jogo de tiro, cuja inspiracao vem da mitologia nordica e street fighter 4, moderniza o classico jogo de luta, a parte interessante e pensar em tudo que esta por tras do que vemos quando sentamos e ligamos o video game, tive oportunidade de jogar o street of fighter, era um dos meus jogos favoritos, agora construindo alguns elementos de jogos na unreal engine posso ver como e trabalhoso, a parte artistica, a historia dos personagens, vejo como tudo e pensado de forma cuidadosa pelos desenvolvedores, algo que antes não tinha dimensao.

Esses documentos explicam como o jogo funciona, incluindo aspectos tecnicos, descreve o publico albo, como o jogo pode atrair os jogadores, pela nostalgia, inovacao, acessibilidade.

O documento sobre DOOM eh muito longo e extremamente detalhado, descrevendo precisamente os personagens, fases, sequencias cinematrograficas, enquanto Valkior eh mais focado na historia e na viabilidade comercial, mostrando como o jogo seria destaque no mercado e por sua vez, street fighter 4 explica os sistemas de combate, incluindo duracao de animacoes e jogabilidade online.  
  
Achei incrivel refletir sobre isso, porque ate entao, não havia analisado profundamente o quanto de coisa que esta envolvida na criacao de um jogo, tal como ler documentacao para os codigos, para linguagens de programacao, ler o documento de conceito foi enriquecedor, trazendo uma visao abrangente de um mundo totalmente novo.   
Ler esses documentos de design de jogos me mostrou como eh importante ter o meu proprio, para o primeiro assignment por exemplo, cuja a primeira tarefa consistia em elaborar um documento falando sobre niveis, jogabilidade, cores, personagens, pensar nisso tudo faz parte da criacao.

First of all, the game design document is very important to ensure that everything is well defined, from simple aspects like the story to more complex ones like technical requirements. It helps maintain a clear vision of the game, facilitating communication among everyone working to bring the project to life, including designers, programmers, artists, and producers, ensuring consistency throughout development.  
The material serves as a support and a reference guide.

Analyzing the three documents, I think each one contains the game concept in great detail. In the case of *DOOM*, it is innovative in the world of horror and action. *Valkyrior* is a shooter game inspired by Norse mythology, and *Street Fighter 4* modernizes the classic fighting game. The interesting part is thinking about everything that happens behind the scenes of what we see when we sit down and turn on the video game. I had the opportunity to play *Street Fighter*, which was one of my favorite games. Now, as I am building some game elements in Unreal Engine, I can see how much work is involved—the artistic aspects, the character stories—I now realize how carefully everything is thought out by the developers, something I had never fully understood before.

These documents explain how the game works, including technical aspects, describing the target audience and how the game can attract players through nostalgia, innovation, or accessibility.  
The *DOOM* document is very long and extremely detailed, precisely describing characters, levels, and cinematic sequences. Meanwhile, *Valkyrior* focuses more on the story and commercial viability, showing how the game would stand out in the market. *Street Fighter 4*, on the other hand, explains the combat systems, including animation duration and online gameplay.

I found it amazing to reflect on this because, until now, I had never deeply analyzed how much goes into the creation of a game. Just like reading documentation for programming languages and code, reading the concept document was enriching, giving me a broad view of a completely new world.  
Reading these game design documents showed me how important it is to have my own, for example, for my first assignment, where the task was to create a document covering levels, gameplay, colors, and characters. Thinking about all these elements is an essential part of game creation.

Lab 2

Para esse exercicio, optei em utilizar a segunda opção, que consiste em arrays e loops, escolhi essa opção por ter mais familiaridade com esses elementos em detrimento da primeira opção. Entao, criei um blueprint que gerou uma grade com cubos, usando loops for e um componente de malha estatica instanciada. Usei os loops para gerar cubos organizadamente e configurei variaveis que me permitiram ajustar o tamanho e espaçamento da grade.

Meu primeiro e maior desafio foi saber como utilizar os loops corrementamente, precisei criar dois, um para dimensão x e outro pra a dimensão y. A principio, os cubos estavam um cima do outro e demorei muito tempo até conseguir descobrir o que estava errado e encontrar uma solução, fiz algumas pesquisas e a solução mais plausivel para mim foi multiplicar o indice do loop pelo espaçamento.

Criei variáveis editaveis, que me permitiu modificar a grade dentro do Unreal Engine.

Estudei e pesquisei sobre o instanced static mesh component para ajustar o custo do processamento.

Como um dos pontos a serem melhorados, creio que criar um grade em 3d possa evoluir o conceito da grade que estava apenas no eixo X e Y, poderia adicionar mais um loop.

Consegui entender melhor alguns conceitos importantes no unreal engine, por exemplo, como usar loops for, a importancia de usar instanced static meshes, como tornar algumas variáveis editáveis, o que me permitiu fazer ajustes sem modificação do código.

Acredito que esse aprendizado me mostrou que futuramente consigo criar mecanicas de jogos baseados em grid, por exemplo, xadrez ou alguns mapas, conseguirei de acordo com a criatividade exigida.

Aprendi a trabalhar com loops e otimização de desempenho, o que é muito importante para automação e eficiência, criar objetos de forma manual é mais demorado, e se, por acaso, o programador precisar de centenas ou milhares de obejtos, colocar um por um não é viável, então com loop conseguimos gerar uma grande quantidade de objetos rapidamente, poupando tempo.

For this exercise, I chose to use the second option, which consists of arrays and loops. I selected this option because I am more familiar with these elements compared to the first option. So, I created a Blueprint that generated a grid of cubes using for loops and an Instanced Static Mesh Component. I used the loops to generate cubes in an organized manner and configured variables that allowed me to adjust the size and spacing of the grid.

My first and biggest challenge was understanding how to use loops correctly. I needed to create two loops:

One for the X dimension

Another for the Y dimension

At first, the cubes were stacked on top of each other, and it took me a long time to figure out what was wrong and find a solution. After researching, the most plausible solution for me was to multiply the loop index by the spacing value.

I created editable variables, which allowed me to modify the grid directly inside Unreal Engine.

I also studied and researched the Instanced Static Mesh Component to optimize processing costs.

One of the improvements I identified is that creating a 3D grid could evolve the concept beyond just the X and Y axes. By adding one more loop, I could extend the logic to the Z dimension, making the grid three-dimensional.

I was able to understand some important concepts in Unreal Engine, such as: How to use loops efficiently, the importance of using Instanced Static Meshes for performance optimization, how to make some variables editable, allowing adjustments without modifying the code manually.

This exercise showed me that, in the future, I will be able to create grid-based game mechanics, such as chess or strategy maps, depending on the required creativity.

I also learned how to work with loops and performance optimization, which is crucial for automation and efficiency. Creating objects manually is time-consuming, and if a developer needs hundreds or thousands of objects, placing them one by one is not viable. With loops, we can generate a large number of objects quickly, saving time and improving workflow