예상 질문

여태까지 일했던 경력 어떻게 살리는지

- 회계 업무에서는 돈 관리하는 일을 맡았기 때문에 작은 실수도 큰 문제로 이어질 수 있다는 점과 정확하고 세세하게 관리하는 것이 얼마나 중요한지는 이전 직장에서의 경험으로 잘 알고 있습니다.

따라서 게임 개발을 할 때도 여태까지 일해서 얻은 경험을 살려 회사에 빠르게 적응해 정확하고 세세하게 게임을 개발하는 것이 제 목표입니다.

회계에서 왜 여기로 왔는지, 왜 처음엔 회계를 선택했고 왜 이제 와서 게임업계로 왔는지

- 회계를 선택한 이유는 그저 단순히 모든 기업에서 회계가 필요하니까 취직이 잘 될 것이라는 생각에 회계를 선택했습니다. 하지만 약 4년 간 회계 업무를 수행하면서 일에 대한 열정과 흥미를 점점 잃어버렸습니다.

그러던 중 제가 어릴 적에 즐겨했던 RPG 만들기 시리즈 게임들이 눈에 들어왔습니다. 그 툴로 자작게임도 만들어보고 남들이 만든 것도 즐기면서 다시 게임을 만들고 싶다는 생각이 들었습니다. 그래서 게임개발 업계에 취직하고자 C언어를 공부하면서 현재 게임업계에서 가장 많이 쓰고 있는 엔진인 유니티와 언리얼 중 선택해야 했는데 저는 캐주얼 게임 제작에 많이 쓰이는 유니티를 배웠습니다. 이런 경험을 통해 게임 업계로의 이직을 희망하게 됐습니다.

게임도 매너리즘에 빠질 거라는 생각은 안 하셨는지

- 정형화돼 세부적인 것만 바뀌는 회계와는 달리 게임은 사람의 상상력을 기반으로 만들어지기 때문에 사람의 상상력이 끝나지 않는 이상 게임은 계속해서 발전해 나갈 거라 생각합니다. 매일매일 새로운 것을 보고 체험하고 구현하는 그런 도전적인 삶을 원했기 때문에 쉽게 매너리즘에 빠질 거라 생각하진 않습니다.

존경하는 인물

- 상투적이라고 할 수 있지만 저는 아버지를 가장 존경합니다. 저의 아버지는 현재 하고 있는 일을 약 30년 넘게 우직하게 하고 계십니다. 거의 매일매일 일하시면서 앓는 소리 하나 안하시고, 쉴 때는 확실하게 쉬시면서 자기자신만의 워라밸을 챙기는 모습이 아들의 입장에서 봤을 때 정말 대단하다고 느꼈습니다.

그런 아버지의 삶의 자세가 제 성장과 발전에 큰 영향을 미쳤습니다. 저의 성격도 아버지를 보고 배워 하나를 정하면 우직하게 나아가는 성격이 되었으며, 앞으로 나아가는 것과 동시에 좀 더 보고 배우고자 노력하고 있습니다.

성격의 장단점

- 저의 성격 중 장점은 하나를 정하면 우직하게 나아가는 것입니다.

- 단점은 자신의 의견을 내세우지 못하는 점

지원동기

- 현재 모바일 게임의 시장 추세는 키우기 형태의 방치형 게임이나 일본에서 대흥행 하고 있는 블루아카이브 같은 스토리나 캐릭터성이 뛰어난 게임, 스킵권 같은 시간을 오래 쓰지 않으면서 캐릭터가 스토리적인 부분이나 게임적인 부분에서 눈에 띄게 성장하는 것을 중요시 여기는 유저들이 대다수입니다. 그런 시장 속에서 이번 콩스튜디오에서 만들려는 신규 캐주얼 게임의 정보를 봤을 때 저는 위 게임의 흥행 가능성을 높게 쳤습니다. 그런 게임에 신입으로써 지원을 해 경험도 쌓으면서 개발진 크레딧에 이름도 올릴 수 있다는 생각에 바로 지원하게 됐습니다.

기술 스택 관련 질문

1. C#의 기본 구문과 개념을 설명해 주세요.

- C#은 객체지향 프로그래밍 언어로, 캡슐화로 데이터를 보호할 수 있으며, 상속으로 부모클래스의 특징과 기능을

자식클래스가 물려받아 사용할 수 있습니다. 다형성으론 오버라이딩이나 오버로딩을 통해 상황에 따라 다른 결과를 낼 수 있습니다.

오버라이딩 = 부모클래스 메서드를 자식클래스에서 재정의하는 것.

오버로딩 = 한 클래스에서 메소드 이름은 같지만 파라미터 개수나 자료형을 다르게 하여 서로 다르게 동작하게 하는 것

2. C#에서 클래스와 객체의 차이는 무엇인가요?

- 클래스는 객체를 생성하기 위한 설계도이며, 객체는 해당 클래스로부터 생성된 실체입니다.

클래스는 객체의 속성과 동작을 정의하는데 사용되며, 객체는 클래스의 인스턴스입니다. 클래스는 여러 객체를 생성하기 위해 사용될 수 있습니다.

3. 유니티에서 C# 스크립트를 작성할 때 사용되는 MonoBehaviour의 기능에 대해 설명해 주세요.

- MonoBehaviour 클래스는 유니티에서 스크립트를 작성할 때 사용되는 기본 클래스로, 게임 오브젝트의 동작을 제어하기 위해 사용됩니다.

MonoBehaviour를 상속받은 스크립트는 Start(), Update() 등의 메서드를 사용하여 초기화 및 업데이트 로직을 구현할 수 있습니다.

4. C#에서 상속과 다형성에 대해 설명해 주세요.

- 상속은 한 클래스가 다른 클래스의 속성과 동작을 이어받는 것을 말하며,

다형성은 같은 이름의 메서드나 속성이 서로 다른 동작을 할 수 있는 성질을 말합니다.

상속을 통해 부모 클래스의 속성과 동작을 자식 클래스에서 재사용할 수 있으며,

다형성을 통해 코드의 유연성과 재사용성을 높일 수 있습니다.

5. 유니티에서 코루틴(Coroutine)이 무엇이며, 어떻게 사용되나요?

- 코루틴은 코드 내에서 구문 실행 도중에 처리를 대기시키거나 순차처리에 함수를 병렬로 동시에 처리할 수 있는 것.

코루틴은 yield 문을 사용하여 일시 중지하거나 지정된 조건이 충족되면 다시 실행됩니다.

주로 시간에 따른 작업이나 연산량이 많은 작업을 분할하여 처리하는데 사용됩니다.

6. C#에서의 예외 처리에 대해 설명해 주세요.

- C#에서 예외 처리는 try-catch 문을 사용하여 수행됩니다.

try 블록 내에서 예외가 발생할 수 있는 코드를 작성하고, catch 블록에서 예외를 처리합니다.

finally 블록은 예외 발생 여부와 관계없이 항상 실행되는 코드를 작성하는데 사용됩니다.

예외는 프로그램의 실행 중에 발생하는 오류를 처리하는데 사용됩니다.

7. C#에서의 제네릭(Generic)에 대해 설명해 주세요.

- 제네릭은 메서드나 클래스를 작성할 때 데이터 형식을 지정하지 않으며 실제 사용하는 시점에서 데이터 형식을 지정할 수 있도록 하는 기능입니다.

animator = getcomponent<Animator>();

8. 유니티 생명주기

- Awake - OnEnable - Start - FixedUpdate - Ontrigger - OnCollision - 마지막은 OnDestroy

9. C#에서 사용되는 LINQ(Language Integrated Query)에 대해 설명해 주세요.

- LINQ는 C# 언어에서 쿼리 기능을 사용하는 것입니다. 여기서 쿼리란 데이터를 조회나

조건에 맞는 데이터를 추출, 조작하는 명령어입니다. 제가 만든 포폴에서도 중복값을 제거할 때 사용했습니다.

10. 유니티에서의 프로퍼티에 대해서 설명해주세요.

- 필드에 대한 접근자(getter)와 설정자(setter)를 제공하여 데이터를 읽고 쓸 수 있도록 합니다.

get과 set을 둘 중 하나씩만 사용해 읽기 전용이나 쓰기 전용 등 기능을 만들 수 있습니다.

11. 유니티에서의 네임스페이스에 대해서 설명해주세요.

- 클래스, 구조체, 인터페이스 등의 형식을 그룹화하고 관련된 코드를 함께 묶는 컨테이너.

12. 구조체에 대해서 설명

- 클래스와 같은 사용자 정의 데이터형. 구조체와 클래스는 비슷한 부분이 많지만 가장 큰 차이는 클래스는 참조 형식이지만,

구조체는 값 형식이라는 점이 가장 큰 차이.

13. 배열에 대해서 설명

- 데이터 형식이 같은 변수를 한꺼번에 묶어서 처리하기 위해 사용되는 것.

14. 이벤트에 대해서 설명

- 이벤트는 델리게이트를 이용하여 객체의 상태 변화나 특정 액션 발생을 다른 객체에 알리는 기능을 제공.

15. 열거형에 대해서 설명

- enum 키워드를 사용해 정의하며 정수형 상수에 이름을 붙여서 이해하기 쉽게 해주는 역할

16. 델리게이트에 대해서 설

- 델리게이트는 메서드를 참조하는 개체로, 다른 메서드를 호출할 수 있는 객체이며 주로 이벤트 처리 등에 사용.

회사에 대해 질문할 것

- 주 3~4일 출근이라고 했는데 5일 출근은 불가능한지?

- 혹시 무리가 되지 않는다면 가디언 테일즈의 전작이었던 던전 링크처럼 이번 신작 캐주얼 게임도 가디언 테일즈의 후속작인지 궁금합니다.

저녁 때쯤 합격 여부