

# 자기 소개서

## 직무 지원 동기

저는 '내가 만든 게임을 통해' 세계 각지의 사람들이 소통하고 교류하며 친구가 되는 경험을 하게 해주고 싶다'는 꿈을 가지게 되었고 이 꿈을 이루기 위해 게임 개발자가 되기로 결심했습니다. 이 꿈은 중학교 3학년 때, 친구의 추천으로 접한 *DayZ Standalone*이라는 스팀 게임을 통해 시작되었습니다.

이 게임에서 저는 모르는 외국인 친구들과 함께 파밍하고, 좀비를 잡으며 생존을 이어가며 게임을 즐겼습니다. 서로 얼굴도 모르고, 언어도 잘 통하지 않았으며, 문화적 배경도 전혀 달랐지만, 게임이라는 하나의 수단을 통해 언어의 장벽, 문화의 벽, 인종의 벽을 넘어 자연스럽게 소통하고 협력할 수 있었습니다. 특히 함께 위험을 헤쳐 나가는 과정에서 빠르게 유대감이 형성되었고, 그렇게 게임을 통해 친구가 되어가는 경험은 제게 매우 신선하고 강렬한 인상을 남겼습니다. 이 게임을 접하기 전에는 단순히 제 생활반경 안에 있던 사람들과만 소통하고 교류했지만, 게임 덕분에 제 교류 범위가 세계로 확장되어 다양한 사람들과 소통하고 교류하며 친구가 될 수 있었습니다.

이 경험을 통해, 단순한 오락 수단으로만 생각했던 게임이 세계 각지의 사람들과 소통할 수 있는 수단이 될 수 있다는 점에 크게 감명받았고 '나도 내가 느꼈던 이 감정과 경험을' 내가 만든 게임을 통해 다른 사람들도 느낄 수 있게 해주고 싶다'는 꿈이 생겼습니다. 이 경험 이후 저는 매년 학교에서 장래희망을 적는 시간마다 저는 빠짐없이 '게임 개발자'를 적어 제출하였고, 그 꿈을 이루기 위해 지금까지 꾸준히 노력해오고 있습니다.

## 성격의 장점

첫 번째 장점은 강한 책임감입니다. 맡은 일은 끝까지 책임지고 완수하는 성격입니다. CheeseRoll 게임 프로젝트 당시, 커스텀 엔진은 물론 상용 엔진도 처음이었지만, 막히는 부분이 생기면 매일 교수님을 찾아가 1:1로 조언을 구하며 끝까지 문제를 해결했습니다. 특히 Collision Resolution 구현이 뜻대로 되지 않았을 때도 끈질기게 질문한 끝에 교수님의 제안을 바탕으로 문제를 성공적으로 해결할 수 있었습니다.

당시 교수님은 영어권이셨고, 저는 영어도 익숙하지 않았고 질문을 자주 하던 성격도 아니었지만, 책임감을 갖고 앞뒤 재지 않고 도움을 요청하며 주어진 역할을 끝까지 수행했습니다. 이러한 경험을 통해, 저는 어떤 상황에서도 책임감을 갖고 끝까지 문제를 해결하는 태도를 실무에서도 발휘할 수 있다고 확신합니다.

두 번째 장점은 끈기입니다. SoulsLike 게임을 개발하던 중, 화면이 미세하게 떨리는 현상이 발생했는데, 대부분의 자료는 이를 해결하는 방법만 제시할 뿐, 원인과 원리는 제대로 설명하지 못했습니다. 하지만 저는 단순히 해결에 그치지 않고 왜 그런 현상이 발생했고, 어떻게 해결되는지를 끝까지 이해하고자 했습니다. 약 2주간 Unity 엔진의 내부 작동 방식과 보간 처리에 대해 집중적으로 공부한 끝에, 화면 떨림 현상의 근본 원인과 그 해결 원리를 스스로 명확히 이해할 수 있었습니다.

이러한 경험을 통해, 저는 실무에서도 예상치 못한 문제를 단순히 피하거나 임시방편으로 넘기지 않고, 끈기 있게 분석하고 근본적으로 해결해 나가는 태도를 갖췄다고 자신 있게 말할 수 있습니다.

제 성격의 세 번째 장점은 친화력입니다. 저는 어떤 환경에서도 주변 사람들과 큰 갈등 없이 원만하게 잘 어울리고 지낼 수 있는 능력을 가지고 있습니다. 그 예로, A Round Entertainment에서 근무하는 동안 팀원들과 빠르게 친해지며 원활한 협력 관계를 구축할 수 있었습니다. 이로 인해 팀 내에서 자연스럽게 신뢰를 쌓을 수 있었고, 감사하게도 대표님께서 저를

좋게 봐주셔서 그룹 리더를 맡겨주셨습니다. 저의 이러한 성격은 여러 개발자님들과의 협업에 큰 도움이 될 것이라 믿습니다.

마지막 장점은 빠른 적응력과 학습 능력입니다. A Round Entertainment 인턴십 당시 JavaScript와 React Native는 처음이었지만, 회사는 학교가 아닌 실제 고용 환경인 만큼 즉시 업무에 투입될 수 있도록 사전 준비가 중요하다고 생각했습니다. 업무 전 미리 학습하고, 실무 코드에 빠르게 익숙해지기 위해 검색엔진, 유튜브, AI 도구 등을 적극 활용해 관련 자료를 찾아보며 퇴근 후에도 코드를 분석했습니다. 그 결과, 첫 업무를 일주일 내에 성공적으로 마무리할 수 있었고, 낯선 기술 환경에서도 빠르게 적응해 실무에 기여할 수 있다는 자신감을 얻었습니다.

## 문제 해결 능력

Finite State Machine으로 몬스터 AI를 구현하던 중, 액션 전환 시 애니메이션이 제대로 종료되지 않고 반복 실행되는 문제가 발생했습니다. FSM에서는 OnEnter와 OnExit 메서드를 통해 상태 전환 시점을 명확히 파악할 수 있어, 애니메이션의 초기화와 정리가 가능했습니다. 그러나 BT(Behavior Tree)에는 이러한 구조가 없어, 행동의 진입과 실행 상태를 구분하기 어려운 것이 문제의 원인이었습니다.

이 문제를 해결하기 위해, FSM에서 사용하던 개념을 BT에도 적용하고자 ActionManager를 직접 설계했습니다. 각 액션 노드에 OnEnter, Execute, OnExit, ExecuteInFixedUpdate 메서드를 정의하고, 현재 실행 중인 액션을 추적해 진입 시점에만 초기화 코드가 실행되고, 전환 시에만 정리되도록 구현했습니다.

또한, BT는 기본적으로 우선순위가 높은 행동을 먼저 실행하는 구조입니다. 그러나 실제 게임 플레이에서는 이러한 구조가 전투 흐름을 부자연스럽게 만드는 문제가 있었습니다. 예를 들어 같은 부모를 가진 노드들 중 우선순위가 높은 원거리 공격 노드와 우선순위가 낮은 근거리 공격 노드가 있을 때, 근거리 공격이 실행 중임에도 우선순위가 높다는 이유로 원거리 공격이 이를 중단시키고 실행되는 상황이 발생했습니다. 이를 해결하기 위해, 저는 실행 중이던 자식 노드를 기억하고, 그 노드가 완료되기 전까지는 같은 부모 노드 내에서는 다른 노드로 전환되지 않도록 제어하는 커스텀 노드를 구현했습니다. 덕분에 공격이 중간에 끊기지 않고 끝까지 자연스럽게 이어지며, 해당 부모 노드에서 아무 행동도 실행 중이 아닐 경우에는 우선순위에 따라 동작합니다. 단, 당연히 상위 노드에서 피격 같은 인터럽트 상황이 발생하면, 현재 행동은 즉시 중단되고 흐름이 상위 노드로 전환됩니다. 이로써 BT의 우선순위 구조는 유지하되, 불필요한 전환 없이 더 자연스럽게 몰입감 있는 전투 흐름을 구현할 수 있었습니다.

이러한 설계는 단순한 문제 해결이 아니라, FSM을 구현하면서 익혔던 구조적 개념과 흐름 제어 방식을 BT라는 새로운 구조에 맞게 적용한 사례입니다. 이전 경험을 그대로 반복한 것이 아니라, 새로운 문제에 맞춰 구조를 재해석하고, 필요한 기능을 직접 설계하고 확장하며 문제를 해결했습니다.

이처럼 직접 구현하며 쌓은 경험은 새로운 문제에 유연하게 적용하고 재구성할 수 있는 응용력으로 이어졌습니다. 이러한 문제 해결 능력은 협업 과정에서도 기존 방식으로 해결되지 않는 문제에 대해 창의적인 해법을 제시하고, 팀에 실질적인 기여를 할 수 있을 것이라 생각합니다.

## 관련 경험 및 역량

첫째, A Round Entertainment에서 소프트웨어 엔지니어 인턴으로 근무하며 Firebase, React Native, Android Studio를 활용해 데이트 앱 Joopi의 온보딩 화면과 사용자 취향 설정 화면을 개발했습니다. 두 화면의 데이터를 Firebase에 저장하고 동기화를 구현했으며, 팀 협업을 통해 프로젝트를 성공적으로 마무리하고 그룹 리더로 선발되었습니다.

둘째, DigiPen Institute of Technology에서 TA로 활동하며 60명의 학생들에게 C#/C++을 지도하고 UI 매니저용 액션 리스트 작성을 도왔습니다. 과제 채점과 질문 응답을 통해 커뮤니케이션 역량과 효과적인 질문의 중요성을 깨달았습니다.

셋째, CheeseRoll 팀 프로젝트에서 물리 프로그래머로서 C++로 커스텀 엔진의 Object class, Object Manager 및 Component Pattern을 구현하고, SAT를 사용한 OBB 충돌 검사와 Collision Resolution을 개발했습니다. 이를 통해 게임 엔진의 구조에 대한 깊은 이해를 얻었습니다.

넷째, Unity에서 FSM, Rigidbody, ProjectOnPlane, 애니메이션 이벤트를 사용해 3D 이동과 콤보 공격 시스템을 구현하고, 타겟 락온 카메라 및 자체 제작한 FSM의 특성이 적용된 Behavior Tree로 다양한 몬스터 AI를 개발했습니다. 또한, 카메라 진동 문제는 WaitForFixedUpdate를 사용해 물리와 렌더링 간 보간을 개선하여 해결했습니다. 이를 통해 유니티 렌더링 프레임과 물리 프레임의 간극과 렌더링 프레임과 물리 프레임 간 보간 로직에 대해 깊은 이해를 하게 되었습니다.

다섯째, Hollow Knight, Super Meat Boy, Celeste 등 유명 2D 플랫폼어의 조작감을 분석하여, Unity 2D Physics와 FSM 기반의 컨트롤러를 구현해 높은 상호작용성과 조작감을 완성했습니다.

여섯째, Unity FPS 프로젝트에서는 Vector3.Lerp, Slerp, 삼각함수를 사용한 절차적 반동 시스템과 Game Feel을 위해 Sin, Cos을 이용하여 총기 Sway, tilt를 구현하였습니다. 이를 통해 Unity에서의 회전 처리와 Vector3.Slerp와 Quaternion.Slerp의 차이 및 Slerp의 특성을 학습했습니다.

## 입사후 포부

저는 항상 제가 상상하고 만들고 싶었던 기능들을 언제든지 마음껏 만들 수 있는 게임 클라이언트 개발자가 되고 싶었습니다. 사소한 일부터 시작해 제가 목표했던 다양한 구현을 할 수 있는 개발자가 되겠습니다. 또한, 학업 중에는 인력과 리소스가 부족해 구현하지 못했던 것들을 실무에서 직접 경험하고 실현함으로써 회사에 기여하고 싶습니다. 회사에서 맡은 기능을 구현하는 과정에서 막히는 상황이 분명 발생할 것 입니다. 이러한 경우, 회사 스케줄에 지장을 주지 않도록 사수님들과 동료분들과 소통하며 도움을 구해 함께 해결하겠습니다. 또한, 게임 디자이너분들과 협업할 때, 해당 기능이 구현 가능한지를 논의해야 하는 상황이 생길 수 있습니다. 이때, 디자이너분들이 원하는 기능들을 최대한 구현할 수 있도록 항상 공부하고 배우겠습니다.

1년 차 목표는 다음과 같습니다. 사수분들이 시키시는 일을 최대한 빠르고 정확하게 수행하도록 노력하겠습니다. 또한, 맡은 일이 끝난 후에는 주체적으로 필요한 작업을 찾아서 하는 개발자가 되겠습니다. 또한 퇴근 후에도 자기 개발을 게을리하지 않고, 지속적으로 공부하며 새로운 기술을 습득하고 성장해 나가겠습니다.

3년 차까지의 목표는 다음과 같습니다. 새로운 신입 개발자분들이 들어오면, 그분들이 잘 적응할 수 있도록 도와주고, 언제나 도움이 될 수 있는 개발자가 되겠습니다. 이를 위해 지속적으로 제 역량을 키우고, 팀과의 원활한 소통을 통해 더욱 성장하겠습니다.

5년 차 목표는 최신 기술과 트렌드를 지속적으로 학습하고 이를 프로젝트에 적용하여 회사의 기술력을 높이는 것입니다. 이를 통해 회사의 경쟁력을 유지하고, 더 재미있는 게임을 만들 수 있도록 기여하겠습니다. 또한, 신입 및 저년차 개발자들의 멘토로서 그들이 성장할 수 있도록 적극적으로 지원하고 지도하겠습니다. 이를 통해 팀의 전체적인 역량을 향상시키고, 협업 문화를 강화하는 데 이바지하겠습니다. 이러한 목표들을 통해 개인의 성장뿐만 아니라 팀과 회사의 발전에도 기여하겠습니다. 또한, 제가 상상하고 만들고 싶었던 기능들을 언제든지 마음껏 만들 수 있는 게임 클라이언트 개발자가 되어있도록 노력하겠습니다.