**이 력 서**

**개인신상**

성 명 : 최 동 현 ( 영문 :Choi Dong Hyeon )



생년월일 : 1994년 01월 10일

주 소 : 서울 송파구 성내천로6길 1-6 (오금동)

핸 드 폰 : 010 - 6745 - 4317

E-Mail : [MicrosoftMainuser@gmail.com](mailto:MicrosoftMainuser@gmail.com)

GitHub : [github.com/Choi-Dong-Hyeon](﻿https://github.com/Choi-Dong-Hyeon)

병역사항 :군필(해군 병장) 2013. 1 ~ 2015. 1

**학력사항**

2009~2012 : 상일미디어고등학교 [서울] **졸업**

**Unity 프로젝트**

|  |  |
| --- | --- |
| **기간** | **내용** |
| 2023.09.01  ~  2023.09.15 | **– Potfolio 소개.** |
| - **Figma Tool을 이용한 Class Logic 의 흐름을 도식화.** |
| - **JsonData 를 FireBase 의 DB 동기화 / Login Auth 규칙 설정.** |
| - **Code 해설.** |
| - **Google Play Store 서비스 출시 과정 소개.** |
| -  **느낀 점과 깨달은 점.** |

**자격사항**

2009. 10. 03. 정보처리기능사 (한국산업인력공단) **최종합격**

2011. 07. 06. 정보기기운용기능사 (한국산업인력공단)  **최종합격**

2019. 10. 18. 1종보통운전면허 (경찰청(운전면허시험관리단)  **최종합격**

**교육과정**

2023. 01 ~ 2023. 08 SBS 게임아카데미 [유니티 프로그래머 실무 교육과정] **수료**

2023. 02 ~ 2023. 05 인프런 [유니티 클라이언트 기초 과정] **수료**

2023. 03 ~ 2023. 05 인프런 [C#프로그래밍 기초 과정] **수료**

|  |  |  |
| --- | --- | --- |
| **기간** | **동아리명** | **학교** |
| 2009~2012 | 여울컴 | 상일미디어고등학교 |

**자기소개서**

**[개발 직무 지원동기]**

수업 중 웹사이트를 만드는 경험을 통해, 평소 하던 게임이 어떻게 동작하고 만드는지 호기심이 생겼습니다.

게임은 프로그래밍 언어로 구성된다는 것을 알게 되었고 학원을 등록해 C언어를 학습했습니다.

콘솔 창에 테트리스를 만드는 것이 목표가 되어 꾸준한 학습 과 도전 끝에 완성하여 성취감을 느껴보았습니다.

고교 진학 때 스스로 주변 고등학교 중에 프로그래밍 수업이 있는 학교를 찾아, IT 미디어과에 진학했습니다.

평소 프로그래밍 수업에 열심히 하여 생활기록부 에 좋은 평가를 받았습니다.

**[몰입하여 이뤄낸 성과]**

방과후 컴퓨터 공부에 몰입하여 정보처리기능사,정보기기운용사 자격증을 단기간에 취득하는 성과를 이뤘습니다.

**[성장하는 게임을 통한 경험]**

학원을 수료하고 여러가지 게임을 만들었습니다. 재밋는 요소를 추가하고 싶어서 꾸준히 공부했습니다.

게임이 완성 된 후에 리팩토링 작업을 하며 스스로 발전한 경험을 했습니다.

여러 게임을 만들어 보며 실무에 대한 준비를 했습니다.

**[본포지션 을 선택한 이유]**

어릴 적부터 게임에 대한 관심과 애정을 갖고 있습니다 게임을 플레이하는 것뿐만 아니라

그 속에서 어떻게 동작하는지,어떻게 만들어지는지 알고싶은 갈증을 느꼈습니다

본포지션에 대한 학습이 갈증을 해소 시켜주었습니다.

**[즐거웠던 경험]**

처음접한 게임 일랜시아 는, 레벨업을 통해 강해지는 즐거움도 있지만 유저들과 파티플레이를 하면서 친근한 관계가 되어,소소한 일상대화를 했던 것이 즐거움 이였습니다. 앞으로 즐거운 경험을 주는 게임을 만들고 싶습니다.

**[본인의 장점 + 채용해야 하는 이유]**

우선 아르바이트를 하며 사회생활을 경험해 봄으로 위계질서가 있는 어느 조직에 있어도 화합이 됩니다.

주어진 업무에 책임감을 가지고 해나가는 장점이 있으며, 업무를 해낸 후 얻는 성취감에 대해 알고 있습니다.

지시사항에 대해 여러번 고민해 보고 목표하는 결과물에 최대한 근접하려 노력합니다.

일손이 모자라 어려운 상황에 동료를 위해 휴가 반납을 한 경험이 있습니다.

꾸준한 자기계발 을 하며 긍정적인 사고를 가지고 있습니다.

**[경험으로 보이는 본인의 장점]**

미용실에서 아르바이트를 한 적이 있습니다 원장님은 매출이 저조하여 걱정이 있으셨습니다.

오피스텔 상권 으로,퇴근후 방문을 원하는 요구를 파악해 마감시간을 2시간 늦춰 보는 것이 어떨지 제안했습니다.

흔쾌히 그러자 하셨고 월 매출 15% 상승을 기록했습니다.

개선하여 더 좋은 결과를 만들고 실천하는 것이 저의 장점입니다.

**감사합니다.**

**포트폴리오**

**개인 프로젝트**

|  |  |  |
| --- | --- | --- |
| **개발 기간** | 2023.09.01 ~ 2023.09.15 | |
| **개발 환경** | Unity 2021.3.21f1 / GitHub / VisualStudio 2022 | |
| **개발 인원** | 1명 | |
| **게임 제목** | BasicMMORPG | |
| **장 르** | MMORPG | |
| **플 랫 폼** | 안드로이드 | |
| **비즈니스 모델** | 와우 / 리니지 / 마비노기 / 로스트아크 | |
| **로그인 씬** | | **로비 씬** |
|  | |  |
| **게임씬** | | **전투씬** |
|  | |  |
| **[ Figma 를 이용한 필요한 기능과 로직 분석 ]**  ● 전역변수의 클래스에 객체 생성된 메서드들.    ● 전역변수의 클래스에 객체 생성된 메서드들의 로직 흐름.  **BaseController / UI\_Base / BaseScene / Json / Define**    **[ FireBase DB / Login Auth 규칙설정 과 동기화]**  **- Authentication / Realtime Database -**   |  |  | | --- | --- | | **Code** | **해설** | |  | ● 로그인 사용자만 데이터에 접근할 수 있는  규칙 설정. | |  | ● 로그인 정보를 DB에 저장. | |  | ● 리얼타임 데이터베이스 구조.  ● Json데이터 실시간 동기화. | |  | ● FireBase 필수 플러그인 임포트 |   **[ State패턴 으로 Animation 상황에 맞는 메서드 선언 ]**  **- BaseController : MonoBehaviour -**     |  |  | | --- | --- | | **Code** | **해설** | |  | ● 기본클래스의 맴버변수. | |  | ● Animation 상황에 맞는 메서드 선언. | |  | ● State패턴에 사용할 가상메서드 정의. | |  | ● 초기화 메서드 호출. |   **- 로직의 흐름 도식화 -**    **[ State 상황에 맞는 메서드 구현 ] (Player)**  **- PlayeControllerr : BaseController -**     |  |  |  | | --- | --- | --- | | **Code** | **해설** | | |  | ● 맴버변수 선언. | | |  | ● 댈리게이트 OnMouseAction에 구독. | | |  | | | | ● InputManager의 대리자에 구독할 메서드. | | | |  | | ● 공격 모션에 맞춰 이벤트 메서드 호출. | |  | | ● 가상메서드 를 재졍의하여 상태 구현. |   **[ State 상황에 맞는 메서드 구현 ] (Monster)**  **- MonsterControllerr : BaseController -**     |  |  | | --- | --- | | **Code** | **해설** | |  | ● Start 메서드 에 호출할 메서드 정의. | |  | ● Idle상태 메서드 정의. | |  | ● Run상태 메서드 정의. | |  | ● Attack상태 메서드 정의. | |  | ● 공격시 발생하는 이벤트 메서드 정의. |   **[ FireBase DB 에 동기화 시킬 JsonData ]**  **- JsonData -**   |  |  | | --- | --- | | **Code** | **해설** | | **Player** | | |  | ● 플레이어의 스탯 데이터. | | **Monster** | | |  | ●몬스터의 스탯 데이터. |   **- 로직의 흐름 도식화 -**    **[ JsonData를 Dictionary 형태로 저장 ](Player)**  **- PlayerStat -**       |  |  | | --- | --- | | **Code** | **해설** | |  | ● JsonData를 받아올 맴버변수. |        |  |  | | --- | --- | | **Code** | **해설** | |  | ● JsonData를 딕셔너리에 저장.  ● JsonData -> List -> Dictionary |   **[ JsonData를 Dictionary 형태로 저장 ](Monster)**  **- MonsterStat -**     |  |  | | --- | --- | | **Code** | **해설** | |  | ● JsonData를 받아올 맴버변수. | |  | ● JsonData를 딕셔너리에 저장.  ● JsonData -> List -> Dictionary |   **[ 공통적인 스탯속성 데이터 선언 ]**  **BaseStat : MonoBehaviour**     |  |  | | --- | --- | | **Code** | **해설** | |  | ● 기본클래스의 맴버변수.  ● 모든스탯의 부모가 될 클래스. | |  | ● 초기화 메서드. | |  |   **[ 공통적인 데이터를 상속받아 스탯속성 구현 ](Player)**  **- PlayerStat -**     |  |  | | --- | --- | | **Code** | **해설** | |  | ● 경험치 변동시 JsonData의  Total경험치에 따라 레벨조정. | |  | ● JsonData를 맴버변수에 대입 |   **[ 공통적인 데이터를 상속받아 스탯속성 구현 ](Monster)**  **- MonsterStat -**     |  |  | | --- | --- | | **Code** | **해설** | |  | ● UI의 HP스크롤바 생성. | |  | ● JsonData를 맴버변수에 대입 |   **[ Singleton 패턴으로 전역에서 사용을 위한 여러 메서드 인스턴스화 ]**  **- Managers : MonoBehaviour -**     |  |  | | --- | --- | | **Code** | **해설** | |  | ● 전역변수 선언. | |  | ● 싱글톤 변수 인스턴스화. | |  | ● 전역에서 호출할 클래스 인스턴스화. | |  | ● 싱글톤 게임오브젝트 생성. | |  | ● 대리자 리스너. |   **[JsonFile 파싱 / Dictionary에 저장]**  **- DataManager -**     |  |  |  | | --- | --- | --- | | **Code** | | **해설** | |  | | ● JsonData의 정보를 저장할  Dictionary 메서드를 구현 인터페이스. | |  | | ● JsonData 파싱. | |  | | | | ● Player / Monster 의 JsonData -> Dictionary 저장후 반환 | | | | **- Managers : MonoBehaviour -** | | | |  | ● Game씬 시작시 Init 메서드 호출. | |     **[ GameObject의 객체 생성 / 삭제 ]**  **- ResourceManager -**     |  |  | | --- | --- | | **Code** | **해설** | |  | ● 생성한 오브젝트를 Dictionary에 저장. | |  | ● 한번이상 생성하는 오브젝트를  Dictionary에서 반환. | |  | ● 게임오브젝트 삭제. |   **[ Listener 패턴 으로 구독된 메서드 CallBack 호출 ]**  **- InputManager -**     |  |  | | --- | --- | | **Code** | **해설** | |  | ● 대리자 선언. | |  | ● 대리자 에 구독된 메서드 CallBack호출. | | **- PlayerController : BaseController -** | | |  | ● 구독신청. | | **- Managers : MonoBehaviour -** | | |  | ● 리스너. |     **[씬 전환시 불필요한 데이터 초기화]**  **- SceneManagerEx -**     |  |  | | --- | --- | | **Code** | **해설** | |  | ● 씬 전환 에 불필요한 데이터 삭제. |   **[ 씬에 필요한 리소스 호출 / 불필요한 데이터 Clear ]**  **- BaseScene : MonoBehaviour -**     |  |  | | --- | --- | | **Code** | **해설** | |  | ● EventSystem생성.  ● 씬전화시 불필요한 데이터 Clear. | | **- LoginScene : BaseScene -** | | |  | ● Login씬에 필요한 리소스 로드.  ● 로드시 불필요한 데이터 Clear. | | **- GameScene : BaseScene -** | | |  | ● Game씬에 필요한 리소스 로드.  ● 로드시 불필요한 데이터 Clear. |   **- 로직의 흐름 도식화 -**    **[ BGM / Effect 로 구분 하고 Dictionary에 저장하여 호출]**  **- SoundManager -**     |  |  | | --- | --- | | **Code** | **해설** | |  | ● 변수선언. | |  | ● Effect / BGM 게임오브젝트 생성. | |  | ● enum타입으로 Effect / BGM 을  구별하여 사운드 재생하는 메서드. | |  | ● 씬전환 에 불필요한 데이터 제거. |   **[ UI를 팝업 / 고정 UI 를 구분 하여 생성 ]**  **- UIManager -**   |  |  | | --- | --- | | **Code** | **해설** | |  | ● 팝업UI / 고정UI 를 생성. | |  | ● 팝업UI 생성시 소팅오더 관리.  ● 팝업UI Stack 자료구조 에 저장. | |  | ● 팝업UI 닫기 / Stack에서 해제. |   **- 로직의 흐름 도식화 -**    **[ UI의 원하는 Componenet를 찾아 매핑 ]**  **- UI\_Base : MonoBehaviour -**     |  |  | | --- | --- | | **Code** | **해설** | |  | ● UI컴포넌트 타입을  enum 상수 네임 으로 저장. | |  | ● Dictionary에 매핑된 타입을 찾음. | | **- UI\_Popup : UI\_Base -** | | |  | ● 팝업으로 띄울때 소팅오더 카운팅. | | **- UI\_Scene : UI\_Base -** | | |  | ● 고정UI경우 소팅을 카운팅하지 않음. |     **[ UI에 Event 발생시 등록된 메서드 CallBack호출 ]**  **-UI\_EventHandler : Monobehaviour ,IPointerDownHandler ,IDragHandler ,IEndFragHandler ,IDropHandler-**   |  |  | | --- | --- | | **Code** | **해설** | | **\** | ● UI이벤트를 CallBack 호출 할 대리자. | |  | ● CallBack메서드 실행 조건.  . |     **[ 한눈에 관리할 수 있도록 하나의 Class에 모든 상수 선언 ]**  **- Define -**     |  |  | | --- | --- | | **Code** | **해설** | |  | ● enum상수들. |   **[ App서비스 출시 ]**  **- Google Play Store 서비스 등록 과정 -**   |  |  | | --- | --- | |  | **해설** | |  | ● 프로젝트 네임 설정 / App 아이콘 설정. | |  | ● App 번들 설정. | |  | ● API33 이상 설정. | |  | ● 버전 업데이트시 필요한 Key 설정. | |  | ● 확장자 .aab 빌드. | |  | ● 개인정보 처리방침 생성. | |  | ● App 번들 업로드. | |  | ● 등록 App 검토. | |  |  | | | |
| **프로젝트를 진행하면서 어려웠던 점** | **1. 데이터 연동시 서버에 대한 개념을 학습하는데 어려움이 있었습니다.**  **2. 반복되는 코드가 많아지면서 가독성도 떨어졌으며 코드 변경 시 불편함이 있었습니다.** | |
| **깨달은 점** | **1. 기초 부터 차근차근 학습하면 새로운 개념들을 쉽게 습득 할 수 있다는 것이었습니다.**  **2. 리팩토링 과정을 거치며 가독성을 높이고 / 모듈화된 메서드로 다듬는 법을 터득했습니다.** | |

**감사합니다**