개발 완료 보고서

제출일	2024년 4월 5일
팀 명	MVP희정쨩 4기
구성원	이준호, 이동준, 박희정

<table>
<tr><td>프로젝트 명</td><td colspan="4">LinEz 개발</td></tr>
<tr><td>활동 일자</td><td colspan="2">2024년 3월 29일
~ 2024년 4월 5일</td><td>장소</td><td>광주인력개발원
공학1관 드론융합실</td></tr>
<tr><td>주요 주제</td><td colspan="4">C언어의 제어문, 반복문, 난수, 배열, 함수, 포인터 동적할당을
이용한 프로그램 개발</td></tr>
<tr><td>개발 목적</td><td colspan="4">C 언어에 대해 학습한 내용을 바탕으로
프로그램의 요구 사항을 분석하여 기능을 구현하고자 함.
팀 프로젝트 경험을 통해 상호 간 의사소통의 필요성을 파악하며, 협업 능력 증진.</td></tr>
<tr><td rowspan="6">업무 분담</td><td rowspan="2">이준호</td><td>기획 업무</td><td colspan="2">개발계획서 및 구현 일정 계획표 작성과 업무 분담</td></tr>
<tr><td>개발 업무</td><td colspan="2">함수와 포인터를 이용하여 포탈 구현 및 필드 생성
게임 내 조작키 구현
잡화상, 성소, 강화소 구현
동적할당을 이용하여 마을, 포탈 이동 주문서 구현
코드 취합</td></tr>
<tr><td rowspan="2">이동준</td><td>기획 업무</td><td colspan="2">요구 사항 분석, 작업 진행 상황에 따른 유동적 업무 분담</td></tr>
<tr><td>개발 업무</td><td colspan="2">몬스터 관련 기본 정보 구조체 및 함수화
몬스터, 유저 전투 로직 구현
장비 강화 구현
전체 코드 취합 및 게임 내 오류 테스트 및 수정</td></tr>
<tr><td rowspan="2">박희정</td><td>기획 업무</td><td colspan="2">로직 순서도 및 계발 완료 보고서 작성</td></tr>
<tr><td>개발 업무</td><td colspan="2">함수 포인터 구조체를 이용하여 기본 아이템 생성
아이템 드롭 이후 인벤토리로 바로 저장할 수 있도록 구현
아이템 상점 구매 시, 인벤토리로 저장할 수 있도록 구현
인벤토리 내 아이템 장착 및 탈착 구현
코드 취합</td></tr>
<tr><td>개발 환경</td><td colspan="4">Ubuntu / Visual Studio Code / GitHub / Google Draw-io / Discord</td></tr>
</table>

일정표

항목		03 / 29		03 / 30		03 / 31		04 / 01		04 / 02		04 / 03		04 / 04		04 / 05	
회의	개발계획서 및 일정표 작성																
	요구분석서 및 순서도 작성																
	중간점검 및 병합																
구현	메인																
	로직 / 맵 구현																
	로직 / 포인터																
	로직 / 랭업																
	로직 / 전투																
	로직 / 게임 아이템																
	로직 / 게임 내 npc																
	로직 / 랜덤																
	테스트 및 디버깅																
제출	완료보고서 및 문서 작성																
	스크린샷 및 실행영상 촬영																

요구 분석서

□ 요구사항 분석서

유형 분류	세부 분류	요구 분석 내용	담당자	
			정	부
맵	이동	1. 50 * 50크기의 맵 6개 구현	이준호	이동준
		2. 1.1에서 시작해 50.50에서 다음 층으로 이동		
		3. 수동으로 마을 복귀 or 주문서를 통한 복귀및 좌표 기억		
		4. 방향키를 이용한 이동		
	조우	1. 마을에서 npc 접촉시 해당 npc 제공사항 시각화화	이준호	박희정
		2. 잡화상, 성소, 제련소 구현 및 상호작용		
전투		1. 용사의 체력 및 초기 상태.	이동준	이준호
		2. 맵 진입시 층에 맞는 몬스터 리젠		
		3. 일반 몬스터, 용사, 보스등장 확률		
		4. 적의 데미지 및 스테이터스 작성성		
		5. 전투 상황 구현		
아이템	강화	1. 각종 티어 아이템 제작 및 장착 여부	박희정	이동준
		2. 아이템들의 성능과 소모 아이템 성능		
		3. npc와 아이템 구매, 판매(1티어 장비와 물약)		
	강화	1. 장비의 인챈트(강화)	박희정	이준호
		2. 드랍 아이템 구현		

코드 설명	이준호	· #define SIZE 50 char map[SIZE][SIZE] 50*50 크기의 맵.
		· void vilage() 모드 함수가 모이고 게임이 진행되는 마을 함수.
		· void dungeon1_4(), 5() 던전 1층~4층, 5층을 맵을 구현한 함수.

		· int height = 7, width = 3;
		int **memory;
		int i;
		memory = (int **)malloc(sizeof(int *) * height);
		for (i = 0; i < height; i++)
		memory[i] = (int *)malloc(sizeof(int) * width);
		순간이동주문서, 유저 좌표 7 개까지 저장 가능하며
		2차원 배열에 동적할당을 이용하여 구현.

		· for (i = 0; i < height; i++)
		{
		free(memory[i]);
		}
		free(memory);
		순간이동주문서 2차원 배열 동적할당 해제.

		· if (control == 'w' && y >1){} else if (control == 'a' && x>1){} else if(control == 's' && y < SIZE - 2){} else if(control == 'd' && x < SIZE - 2){} 'w, s, a, d' 방향키 조작, 맵 밖을 넘어가지 않도록 로직 구현.

		· else if (item == 2){
		printf("마을이동 주문서 칸입니다₩n");
		int stage = 0;
		int x = 1, y = 1;
		village(stage, x, y, player, memory, m_count);}
		마을 이동 주문서

		· if (player->hp <= 0)
		player->hp = player->total_hp * 0.1;
		int stage = 0;
		int x = 1, y = 1;
		village(stage, x, y, player, memory, m_count);
		유저 체력 0이하 시 10% 체력과 함께 마을로 귀환 구현.

코드 설명	이동준	· #include “getch.h” 사용자에게 하나의 키를 입력받고 즉시 반환하는 함수 이동키를 눌렀을때 엔터 입력하지않고 바로 움직이기 위해 적용 리눅스환경에서는 getch()에 해당하는 함수가 없어 사용자 정의 헤더 파일로 만들어 작동 시켰습니다.
		· typedef struct {}monster; 몬스터의 hp, 공격력, 드랍골드, 추가 hp 구조체 구현
		· typedef struct {}named; 적 영웅의 hp, 공격력, 드랍골드, 추가 hp 구조체 구현
		· typedef struct {}boss; 보스의 hp, 공격력, 드랍골드, 추가 hp 구조체 구현
		· void moninfo(int, int, monster *); 적의 정보를 불러오는 함수
		· void heroinfo(int, named *); 적 영웅의 정보를 불러오는 함수
		· int attack(int, int, int, int); 사용자 데미지 산출 값을 리턴해 주는 함수.
		· int nomal(int, int, int); 던전1~4층에서 출몰하는 몬스터를 구분해 주는 함수.
		· void boos1(int, Boss *); 바포메트의 기본 스텟 및 드랍 골드를 구현해 주는 함수.
		· void boos2(int, Boss *); 찐보스의 기본 스텟 및 드랍 골드를 구현해 주는 함수.
		· void boos3(int, Boss *); 찐찐보스의 기본 스텟 및 드랍 골드를 구현해 주는 함수.
		· void battle(); 각 몬스터별 전투 현황에 따라 승리와 패배 시 결과 구분, 유저 행동 1번 후 적 공격, 유저의 걸음수 5 미만시 조우 불가, 아이템 사용, 도주, 적 공격 분류, 찐 보스와 찐막 보스는 이전 보스 처치 플래그가 있어야 등장하도록 모든 전투를 모은 함수.
		· void sanctuary(status *player_ptr() 성소, 유저 체력 회복 함수.
		· void upgrade_li(); 제련소 함수, 주문서 구입 및 장비 강화 구현 함수.

코드 설명	박희정	· struct Defensive() 장착 가능한 아이템을 만들 수 있는 구조체
		· void status_li() main에 선언된 Defensive 구조체의 주소값을 받아 유저의 기본 스텟을 설정하는 함수
		· struct medicine 물약 아이템을 만들 수 있는 구조체
		· struct items 물약 외 소모 아이템을 만들 수 있는 구조체
		· struct Inventory 장비 아이템을 보관할 수 있는 구조체
		· struct S_Inventory 소모 아이템을 보관할 수 있는 구조체
		· void cl_get_2ti_20() 몬스터 처치 시 20% 확률에 따라 2 티어 아이템이 드롭되며 인벤토리 안으로 저장할 수 있는 함수
		· void cl_get_3ti() 몬스터 처치 시 지정한 확률에 따라 3티어 아이템이 드롭되며 인벤토리 안으로 저장할 수 있는 함수
		· void cl_get_4ti() 몬스터 처치 시 지정한 확률에 따라 4티어 아이템이 드롭되며 인벤토리 안으로 저장할 수 있는 함수
		· void status_li() main에 선언된 구조체의 주소값을 받아 유저의 기본 스텟을 설정하는 함수
		· struct Equip_Items 인벤토리 안에 있는 아이템을 장착할 수 있는 구조체
		· void print_equip() 장착 아이템을 출력하는 함수.
		· void store() 상점에서 아이템을 구매하면 유저 인벤토리 안으로 아이템이 보관되며, 그 가격만큼 유저 골드가 차감되는 함수.
		· void put_on() 유저 스텟, 인벤토리, 장착 구조체 등 모든 아이템의 구조체를 주소로 받아 해당 함수를 통하여 유저의 기본 스텟 출력, 현 보유 중인 아이템 리스트 인벤토리 내 장착 가능한 아이템 탈착을 구현한 함수.

프로젝트 후기	이준호	세 번째 팀 프로젝트 “LinEz’를 마무리하게 되었습니다. 이번 팀 프로젝트에서는 맵 파트, 마을 이동 주문서, 순간 이동 주문서, 1차 병합을 맡았습니다. 이 전의 팀 프로젝트에서 맡지 못했던 맵 파트를 맡으며 게임의 전체적인 틀을 구현하였고, 동적할당과 포인터 함수를 통해 이동 주문서를 구현하면서 익숙지 않았던 동적할당과 포인터 함수 쓰임에 대해 알게 되었습니다. 또한 1차 병합을 맡으며 원병합의 어려움을 알게 되었습니다. 하지만 팀원분들의 코드를 하나씩 이해가며 실시하니 성공적으로 마무리할 수 있었습니다. 세 번째 같은 팀을 하며 궂은 일을 마다하지 않고 코드를 구현해주시는 동준형님, 이번 팀 프로젝트의 최대 복병인 아이템 부분을 맡아 성공적으로 진행해주신 희정누님 고생 정말 많으셨습니다.
	이동준	이번 프로젝트를 진행하며 많은점을 느꼈습니다. 이전보다 길어진 코드들로 인해 병합하는 과정에서 시간을 많이 사용했습니다. 팀 프로젝트에서는 자신의 코드를 완벽히 아는것은 물론 다른 팀원의 코드를 분석하고 이해하는 과정 또한 중요하다는 것을 느낀 프로젝트였습니다. 코드를 저만 보는것이 아닌 누가 보더라도 확실히 인지할 수 있도록 변수명을 명확하게 작성하고 함수 작성시 크게 한 묶음보다 작은 함수들로 나눠 만드는것이 활용성이 높고 효율적인점을 배웠습니다. 이후 프로젝트에서는 가독성이 높으면서 이용하기 편한 함수들로 코드를 작성해보겠습니다.
	박희정	어느정도 포인터와 구조체를 이해했다는 생각으로 아이템을 선택으나, 실제로 코드를 작성해 보니 오만했다는 것을 알게 되었습니다. 몬스터 아이템 드롭 이후 인벤토리 창으로 바로 넣고, 장착형 아이템과 소모성 아이템을 구별하는 부분과 인벤토리 안 아이템을 꺼내 장착하는 부분을 로직 구상하는 게 가장 어려웠지만 어떻게든 완성해서 성취감을 느꼈습니다. 다만, 취합하는 단계에서 예상 못한 변수가 많이 나와 다시한번 곰곰히 생각해 보니, 개인 프로젝트가 아닌, 팀 프로젝트이기 때문에 구상 단계부터 팀원들과 많은 소통이 필요했다는 사실을 뒤늦게 알게 되었습니다. 예를 들어 어디에서 메인을 쓸 것인지, 구조체를 어떻게 사용할 것인지, 아이템 적용 및 드랍은 어떻게 될 것인지, 맵은 어떤방식으로 구현할 것인지등등.. 그래도 그 덕분에 소통의 필요성을 빠르게 알게 되었으며, 같이 버그를 고쳐나가는 방법도 알게 되었습니다. 다음 프로젝트를 진행한다면 구상 단계에서 팀원들에게 내가 어떤 방식으로 코드를 짤 것이며, 어떤 부분에서 병합하면 되는지를 미리 말을하고 프로젝트를 시작해야겠습니다. 추가로 저에게 충분한 시간을 주면서 믿어준 저희 팀원들이 있어 너무 고마웠습니다!