개발 완료 보고서

제출일 : 2019년 5월 6일

|  |  |  |  |
| --- | --- | --- | --- |
| **팀명** | 레드팀 | | |
| **참여인원** | 서의영, 김경훈, 양승호, 오금성, 조민권 | | |
| **개발프로젝트 소개** | | | |
| **프로젝트**  **주제** | 1 · 2인용 크롬 공룡게임 모작 구현 | | |
| **활동일시** | 4월 29일 18시  ~ 5월 6일 16시 | **장소** | 광주인력개발원 공학1관 IOT 융합 1실 |
| **주요 주제** | C언어와 스레드를 활용한 1 · 2인용 chrome T-rex 게임 모작 구현 | | |
| **개발목적** | - 기존 숙련되지 않은 기능(kbhit 등) 과 지난 주 학습한 내용(구조체, 파일입출력, 메모리 동적할당 등)을 활용하여 chrome T-rex게임을 재현한다.  - 멀티스레드 병렬프로그래밍을 이용해 기존 T-rex게임에 인원변경, 타이머 등을 구현한다. | | |
|
| **개인별**  **임무분담** | 서의영 | 2p 구현, 메인 취합, 코드 리팩토링 | |
| 김경훈 | 맵, 아이템 구현 | |
| 양승호 | 맵 이동, UI 구현 | |
| 오금성 | 키 입력 구현 | |
| 조민권 | 랭킹 구현(파일입출력, 정렬 알고리즘) | |
| **개발환경**  **&**  **요구조건** | - ubuntu 16.04 버전의 교육원 내 서버 상에서 vim, visual studio code 등 에디터 활용  - C언어를 활용하여 문자를 사용한 시각화 출력, 입력하는 값과 상황에 따른 결과 표출  - 쓰레드를 활용한 타이머, 동시 키입력, 플레이어별 각각 따로 화면 구현  - 파일입출력을 활용한 랭킹구현  - 버블정렬 알고리즘을 활용한 랭킹 내림차순 및 최고점수 표출 | | |
|
| **구현기능** | - 메뉴에서 1p, 2p, 랭킹보기, 종료를 선택 할 수 있다.  - 플레이어별 다른 키를 입력받아 화면상에 점프를 구현한다. 점프를 시도시 일정시간 후 다시 내려와야한다.  - 정해진 범위내에서 랜덤하게 장애물을 출력된다.  - 장애물에 닿으면 체력이 하나 감소되며, 최대 체력은 3이다.  - 아이템을 획득하면 아이템에 따라 속도변화, 체력증감, 상태변화 등이 일어난다.  - 1인용을 이용할때는 하나의 화면만, 2인용을 이용할 때는 두개의 화면이 출력된다.  - 한 플레이어가 먼저 게임 오버시, 해당 플레이어의 창은 꺼지지 않고, 다른 플레이어의 게임이 종료될 때 까지 달성 점수, 플레이타임이 출력되어 유지된다.  - 두 플레이어가 모두 게임오버 당해 게임 종료시 랭킹화면이 출력되며, 이번 플레이시 획득한 등수가 강조되어 표시된다.  - 랭킹은 획득점수 또는 플레이시간을 기준으로 내림차순으로 정렬된다. | | |
|
| **예상 문제점**  **및**  **해결 방안** | - 지난 주 새로운 기능의 학습에 있어 팀원별로 이해도가 다르다.  : 관련 부분에 대한 개인 학습을 통해 숙련도를 높인다.  - 전체적인 프로그램에 있어, 세세한 부분을 인원들이 나누어 구현하여, 기능에 따라 겹치거나 우선 적용되어야하는 기능들이 있다.  : 공유문서를 활용, 적극적인 커뮤니케이션을 통해 보완한다.  - 전 인원이 스레드를 다뤄본 경험이 없으며, 맡은 역할에 있어 스레드기능이 주가 되는 인원이 있다.  : 각자 본인의 파트에 중점을 두되 스레드 활용에 신경써, 서로 질의응답하여 이해도를 높인다. | | |
| **계획된 일정표** |  | | |
|
| **실제 일정표** |  | | |
| **요구분석서** |  | | |
| **주요**  **코드**  **설명** | 서의영 | **void\* t\_timer(void\*)**  스레드에서 돌아가는 타이머 함수. 0.1초마다 반복되어 Player 구조체의 elapsedTime멤버에 저장된다. 1p, 2p 모두 게임오버시 스레드가 종료된다.  **void\* t\_game(void\*)**  Game 구조체의 게임스피드를 딜레이로 화면을 clear하고 ***PrintStatusBar***함수와 ***maps***함수를 반복 호출하여 맵의 이동, 스테이터스바 정보를 새로고침하여 보여준다. 1p, 2p 모두 게임 오버시 스레드가 종료된다.  **void Play(Game\* game)**  ***t\_timer, t\_keyInput1, t\_keyInput2, t\_game*** 네개의 함수를 매개변수로 갖는 스레드를 생성하고 종료시 join한다. 각각의 함수에 game 매개변수를 넣는다.  **void InitGame(Game\* g, int playerNum)**  Game 구조체와 플레이어 수를 매개변수로 받아 플레이어, 맵, 게임 정보를 초기화시킨다. 1p게임의 경우 2p를 처음부터 게임오버상태로 초기화하여 1p가 게임오버시 스레드에서 돌아가는 함수들이 잘 종료될 수 있도록 했다.  **void PrintHearts(Player p)**  Player 구조체를 매개변수로 받아 현재 남은 체력을 ♥로 표시하고 닳은 체력은 ♡로 표시해 준다.  **void PrintGameOver(Player p)**  Player 구조체를 매개변수로 받아 게임 오버시 획득 점수, 플레이시간을 표시해준다. 2p게임의 경우 다른 플레이어의 게임이 끝날 때 까지 표시된다. | |
| 김경훈 | **void maps(Game\* g, int player, int \* already\_obs)**  장애물, 아이템이 일정확률로 등장하며, usleep 시간 단위마다 한 칸씩 오른쪽에서 왼쪽으로 이동하는 것을 구현, 총을 쏘는 경우 반대방향으로 이동하며, 장애물과의 충돌 까지 usleep 시간 한 단위에 이동하게 끔 구현  **void get\_Item(Game \*g, int player)**  플레이어와 아이템이 배열 상에서 같은 칸을 차지해 획득했다고 판정시 상호작용  **void damage(Game \* g, int player)**  플레이어와 장애물이 같은 칸을 차지했다고 판정시 체력 소모 상호작용  **void Itm\_heart(Player\* p)**  하트아이템 획득 시 체력변화  **void Itm\_speed\_up(Game\* g)**  속력 증가 아이템 획득 시 usleep되는 시간에 변화  **void return\_speed(Game\* g, int player)**  속력 증가 이후 일정시간 이후 usleep되는 시간 원상복귀 | |
| 양승호 | **void maps(Game\* g, int player, int \* already\_obs)**  배열을 활용하여 맵의 전반적인 기능을 구성, 아이템이 붙어서 생성되지 않도록 최소간격 설정 구현, 제작 초기의 기초 틀을 제작 | |
| 오금성 | **void\* t\_keyInput1(void\*)**  **void\* t\_keyInput2(void\*)**  **(getch(), \_\_fpurge(stdin), pthread\_mutex)**  쓰레드로 구현하여 1p 키(↑) 입력시 플레이어 점프하며 2p 키(w) 입력키와 동시 반응하도록 하였음  **void jump(Game \*g, int player)**  점프키 입력시 플레이어의 위치가 이동하도록 함 | |
| 조민권 | **void SaveScore(Game \*g)**  플레이 기록을 파일입출력을 사용하여 텍스트파일에 저장  **void PrintRanking(Game \*g, SaveData \*sd, int sortby)**  텍스트파일에 저장된 랭킹을 읽어들여서 버블정렬기능을 추가한 랭킹 출력  **void PrintRankerASCII()**  랭킹 아스키아트 출력  **int GetHighScore(SaveData \*sd)**  플레이 마다 최고점수를 출력해야하므로 파일입출력과 버블정렬을 이용하여 랭킹 텍스트파일에서 최고점수 반환 | |
| **팀후기**  **&**  **총평** | - 미숙했던 스레드를 구현해보면서 익숙해질 수 있어서 만족합니다. 다만 사용법 뿐만 아니라 뮤텍스 등의 개념에 대한 이해를 확실히하는게 필요할 것 같습니다.  - 롱-텀 프로젝트에 대한 부담감이 있었지만, 1주차에 학습한 내용을 2주차에 구현하면서 바로 적용할 수 있어서 좋았습니다.  - 이번부터 팀프로젝트와 병행하여 파티프로젝트가 진행되었는데 그에따라 효율적인 일정관리를 할 수 있도록 신경써야겠습니다. | | |
| **개인 소감** | 서의영 | 스레드 학습과 구현에 집중하느라 팀원관리, 일정관리 등의 팀장역할에 부족한 부분이 많았지만, 팀원들이 성실하게 맡은 기능을 잘 구현해주어서 무난하게 프로젝트가 완료된 것 같습니다. | |
| 김경훈 | 이리저리 시간을 많이 사용했지만, 파일입출력이나 스레드구현에 있어 아직 개인적으로는 이해나 활용이 부족함을 느꼈습니다.  후에 팀 혹은 파티 프로젝트에서 어떤 역할을 맡아 구현하게 될 지는 모르지만 부족한 부분 없이 팀에 차질이 생기지 않도록 학습해야겠습니다.  스레드 구현 단계, 또 마무리 단계에서 의영이형이 많이 고생하셨는데 실력의 부족함이나 또 강의실 미개방 등의 사유를 핑계삼아야 하는 것에 아쉬움이 많이 남습니다. 좀 더 노력해야겠습니다. | |
| 양승호 | 맡은 파트가 형들과 같이 하는 파트라 그런지 대부분의 작업을 같이 해야했으나, 팀플에서도 해야하는게 있어서 그런지 많이 못붙어서 작업했던것 같습니다. 그래도 이것저것 배울수 있어 좋은 기회가 되었다고 생각합니다.  레드팀만 되면 누구 하나씩 사라지는거같은데 레드팀을 없애고 핑크팀으로 개명했으면 좋겠습니다. | |
| 오금성 | 동시 다중 입력이라는 것이 생각만큼 많은 문제가 생겨서 쓰레드를 다양한 방법으로 사용해보려고 많이 시도해봤던 것 같습니다. 그리고 구조체나 전체적인 메뉴의 함수화를 이런식으로 할 수 있다는 것을 많이 배웠습니다.  동시에 두 프로젝트를 진행함에 있어서 공룡게임에 많은 시간을 투자했지만, 투자한 시간 대비 효율이 안나와서 많이 아쉬웠습니다. | |
| 조민권 | 파일입출력을 이전에 몇번 사용하여, 어느정도 이해간 상태라 엄청 크게 막히지는 않았습니다. 하지만 이전에 사용하였던 방식과는 조금 다르게 응용하여 사용하다보니 버그부분을 쉽게 찾지못하였고, 수정하는데 쵸큼 시간이 많이 걸렸지만, 잘 끝낸것같아 다행입니다.  다른 분들이 맡은 파트에 비하면 제가 맡은 파트의 난이도는 쉬워서 조금 아쉽습니다.  직접 해보고, 시행착오를 겪어보며 본인의 것으로 만드는 건데, 아직 실력도 없으며, 맡아서하기 무서워서 쉬운 랭킹을 고른게 조금 후회됩니다.  프로젝트를 두개를 병행하여 하다보니, 아직 익숙치않아 한 쪽으로 크게 비중이 치우치게 되어 그 부분이 요번 두 프로젝트의 아쉬움입니다. | |
| **비고** |  | | |