|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **기초 프로젝트 최종 보고서** | | | | | | | | | | |
| 프로젝트 분야/주제 | 사용자 데이터를 이용한 산성비 게임 | | | | | | | 총 투입인원 | | 4명 |
| 프로젝트 명 | 국문 | | | | 사용자 데이터를 활용한 텍스트 매칭 게임 | | | | | |
| 영문 | | | | Text matching game using user data | | | | | |
| 수행기간 | 2018. 05. 29 ~ 2018. 6. 18 | | | | | | | | | |
| 참여자 | 소속 | | | 참여자성명 | | 학년 | 학번 | 휴대전화 | 이메일 | |
| 컴퓨터공학과 | | | 백승한 | | 3 | 2014244115 | 010-6307-3054 | hameli@naver.com | |
| 컴퓨터공학과 | | | 이권우 | | 2 | 2015244070 | 010-4183-8548 | kenu12@naver.com | |
| 컴퓨터공학과 | | | 성소연 | | 2 | 2017243060 | 010-4758-5638 | sos980621@naver.com | |
| 컴퓨터공학과 | | | 임동범 | | 2 | 2017243122 | 010-8719-7100 | ehdqja1020@naver.com | |
|  | | | | | | | | | | |
| **개발결과요약** | |  | | | | | | | | |
|  | |  | | | | | | | | |
| 1. 개발 결과물에 대한 요약   * 해당 프로그램은 한컴타자연습의 미니게임 중 하나인 ‘산성비’ 게임을 모티브로 제작됨 * 파일입출력을 통한 데이터 파싱으로 사용자 문자열을 이용한 게임 진행이 가능함(코드 수정이 아닌 텍스트 파일 수정을 통한) * 쓰레드를 통해 사용자 입력과 문자의 출력을 독립시킴으로 원활한 게임 진행을 가능하게 함 * 최종적으로 사용자의 분당 타자수를 알려줌으로 사용자가 효율적으로 타자연습을 수행할 수 있도록 도와줌   2. 요구사항 분석 대비 핵심 개발 사항   |  |  |  | | --- | --- | --- | | 순번 | 계획서 상 요구사항분석 내용 | 요구사항 대비 개발 사항 | | 1 | 파일 입출력을 통해 읽어온 데이터를 연결리스트에 저장 | 읽어온 데이터를 성공적으로 연결리스트에 저장하였으며 사용자 데이터 또한 코드의 수정이 아닌 텍스트 파일을 일부 수정하여 통해 즉시 시현 가능함 | | 2 | 읽어온 데이터를 바탕으로 메뉴를 출력하는 기능 | 윈도우 API를 이용하여 CLI화면이 출력된 상태에서 특정 위치에 메뉴를 출력하고 역할을 다하면 해당 부분만 삭제함으로써 불필요한 화면갱신을 최소화시킴 |  |  |  |  | | --- | --- | --- | | 순번 | 계획서 상 요구사항분석 내용 | 요구사항 대비 개발 사항 | | 3 | 사용자가 입력한 데이터를 바탕으로 텍스트 파일을 읽어오는 기능 | 사용자의 입력에 따라 메뉴의 데이터가 담긴 파일에서 읽어온 데이터에 ‘.txt’를 추가하여 해당 파일을 검색함  세부 기능은 1번 기능과 중복됨 | | 4 | 게임 진행 과정에서 사용자 입력을 기다리는 기능 | 콘솔창에서 출력 커서가 1개로 제한되는 문제로 인해 입력된 데이터를 저장하는 기능과 화면 상에 입력된 글자를 나타내는 기능을 분리함 | | 5 | 사용자의 입력과 생성되어 있는 글자를 비교하는 기능 | 사용자의 입력 데이터와 생성되어 있는 글자를 비교하는 과정에서 서로 같은 연결리스트를 공유하기 때문에 쓰레드 동기화 문제가 발생하여 해당 문제점을 해결하기 위해 c++의 mutex 키워드를 사용하였고 이를 이용해 문제점을 해결함 | | 6 | 텍스트 파일에서 읽어온 데이터를 바탕으로 글자 노드를 생성하는 기능 | 텍스트 파일에서 읽어온 데이터를 바탕으로 이중 연결리스트를 생성하여 읽혀온 문자열과 해당 노드가 화면에 출력될 위치정보를 저장함  이 과정에서도 쓰레드 동기화 문제를 해결하기 위해 mutex 키워드를 사용함 | | 7 | 분당 타자수를 계산하고 출력해주는 기능 | 분당 타자수를 계산함에 있어 영어의 경우에는 문제가 없었으나 한글의 경우 입력 횟수와 글자의 길이가 일치하지 않는 문제가 발생하여 해당 문제를 해결하기 위해 입력 데이터를 유니코드와 시켜 해당 데이터를 자모음 단위로 잘라 워드의 개수를 셈으로 인해 해당 문제를 해결함 | | | | | | | | | | | |
| **상세개발내용** | |  | | | | | | | | |
|  | |  | | | | | | | | |
| 1. 상세 개발 내용  -> 이번 프로젝트에 인해 가장 중요한 키워드 3개를 뽑자면 다음과 같은데 해당 키워드가 코드 내에서 어느 부분에서 사용되었고 사용된 이유에 대해 설명하고자 함   * 이중 연결리스트(Double Linked List) * 쓰레드(thread) * 분당 타자수 계산 * 이중 연결리스트(Double Linked List)   -> 이중 연결리스트는 단일 연결리스트의 문제점을 해결하기 위해 만들어진 것으로 단일 연결리스트의 비해 데이터의 크기 자체를 더 크나 단방향 탐색이 아닌 양방향 탐색이 가능하기 때문에 탐색이 잦은 코드 내에서 효과적으로 사용이 가능함.  해당 프로그램 내에서는 크게 두 부분에서 사용되었는데 첫번째는 사용자 입력에 해당하는 데이터를 읽어온 후 데이터를 저장하기 위해서 사용됨.  일단 텍스트 파일내의 데이터의 개수를 측정하는 것이 불가능함으로 연결리스트를 사용해야 하지만 화면에 나타낼 데이터를 선정하는 과정에서 단방향으로 탐색할 경우 그만큼 단어의 손해가 발생하기 때문에 효율적이지 못함. 이를 보완하기 위해 이중 연결리스트를 활용하여 데이터를 저장하고 무작위로 결정된 값에 의해 짝수일 경우 순방향으로, 홀수일 경우 역방향으로 무작위 횟수만큼 이동하여 도착한 위치의 데이터를 글자 노드 생성을 위해 넘겨주게 됨.  두번째는 생성된 글자 노드를 관리하기 위해 사용되었는데 해당 부분의 경우 서로 다른 쓰레드를 통해 매번 새로운 탐색과 노드의 빈번한 삭제 및 생성이 발생하기 때문에 단방향 연결리스트를 이용할 경우 코드의 복잡도 향상 및 효율성 감소를 겪을 수 있으므로 이중 연결리스트를 활용함    (좌)읽어올 데이터 셋 (우)읽혀온 데이터를 바탕으로 생성된 글자 노드   * 쓰레드(thread)   -> 쓰레드는 보통 한 개 이상의 기능을 동시에 실행시키기 위해 사용이 되는데 이번의 경우에는 게임 진행중 이루어지는 사용자의 ‘입력’과 프로그램에 의한 ‘출력’을 독립적으로 작동시키기 위해 사용함.  해당 프로그램은 콘솔창의 한계로 인해 출력 커서 하나가 콘솔화면 내를 이동하며 데이터를 출력해주는 방식으로 이루어지는데 사용자의 입력과 프로그램에 의한 출력이 동시에 이루어지게 되면 엉뚱한 위치에 텍스트를 출력하는 문제가 발생함. 해당 문제를 해결하고자 사용자를 통한 키입력과 화면에 입력된 정보를 출력하는 것을 분리하여 해당 문제를 해결하였으며 서로 다른 쓰레드가 한 개의 연결리스트를 공유하기 때문에 데이터 동기화 문제가 발생하여 해당 문제를 보완하기 위해 동기식으로 쓰레드를 관리하기 위한 키워드인 mutext를 활용하여 쓰레드간 데이터 동기화 문제를 해결함    사용자 입력과 글자를 출력하는 것이 독립적으로 작동 해야함   * 분당 타자수 계산   -> 분당 타자수 계산하는 방법은 단순히 일정시간 내에 몇번의 입력이 있었는지를 이용하여 계산하는데 영어의 경우 입력 한번에 하나의 글자가 사용되기 때문에 문제가 없으나 한글의 경우 2~3회의 입력으로 한글자가 입력되어 타자수 계산을 위해서는 이 부분을 해결해야함.  해당 문제를 해결하기 위해 입력된 한글을 유니코드로 변환하고, 변환된 데이터를 자모음으로 분해하  여 분해된 개수를 세어 줌으로써 해당 문제를 해결함  2. 팀원 별 개발 역할 및 개발내용     |  |  |  | | --- | --- | --- | | 팀원명 | 상세역할 | 주요 기여 내용 | | 백승한 | 게임 및 쓰레드 구현 | 전반적인 게임 시스템 제작 및 쓰레드 관리 | | 이권우 | 데이터 파싱 및 저장 | 파일 입출력을 통한 데이터 파싱과 이중 연결리스트를 이용한 데이터 저장 | | 성소연 | CLI구현 및 사용자 데이터셋 제작 | CLI(Console Line Interface) 구현 및 사용자 데이터 셋을 활용한 메뉴화면 개발 | | 김동범 | 분당 타자수 측정 구현 | 한글 자모음 분리를 통한 한글&영어 분당 타자수 측정 알고리즘 구현 | | | | | | | | | | | |
| **과제수행방법** | |  | | | | | | | | |
|  | |  | | | | | | | | |
| \* 개발 방법 용약   * 주 2회 이상 오프라인 회의를 통해 현재 개발상황을 및 문제점을 토의하고 해당 사항에 대해 해결책을 제시함 * 버전관리 시스템 중 하나인 Git과 Github를 사용하여 개발 과정과 버전을 관리하여 소스 코드 작성의 효율성을 향상시킴   \* 참여개발자간의 역할분담 및 추진 일정   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | 주요내용 | 참여자 | 추 진 일 정 (주) | | | | | | 5/31 | 6/5 | 6/7 | 6/12 | 6.14 | | 게임 및 쓰레드 구현 | 백승한 |  | ○ |  | ○ | ○ | | 데이터 파싱 및 저장 | 이권우 |  |  | ○ | ○ | ○ | | CLI구현 및 사용자 데이터 셋 제작 | 성소연 | ○ | ○ |  | ○ |  | | 분당 타자수 측정 및 구현 | 김동범 |  |  | ○ | ○ | ○ | | | | | | | | | | | |
| **최종결과물** | | |  | | | | | | | |
|  | | |  | | | | | | | |
| 1. 메인화면     - 사용자 이름을 입력 받아 출력해주고 파일 입출력을 통해 읽어온 메뉴를 화면에 출력해줌    - 메뉴에 존재하지 않는 수를 입력하였을 경우 오류 출력   1. 실행화면     - 게임이 시작되면 상단에서 단어가 비처럼 떨어짐  - 매 라인에 1개씩의 단어가 존재하며 떨어지는 속도는 사용자의 진행에 따라 점차 빨라짐  - 우측에 현재 체력과 점수를 출력해줌    - 단어를 입력하여 맞출 경우 점수와 콤보가 증가하고 우측에 타자수가 표기됨  - 단어가 하단 끝에 도달하면 점수를 잃고 콤보가 0으로 초기화됨   1. 종료 화면   - 최종적으로 사용자에 대한 점수와 평균 타자수를 출력해줌으로 게임을 종료함 | | | | | | | | | | |
| **결과활용계획** | | |  | | | | | | | |
|  | | |  | | | | | | | |
| \* 타이핑 속도가 느린 사람들로 하여금 해당 프로그램을 통해 자연스럽게 게임 진행과 같이 타자 연습을 실행함  \* 프로그래머 들을 대상으로 사용자 데이터 셋을 활용하여 프로그래밍 언어의 예약어 및 자주 쓰는 함수명 등을 타이핑 연습하여 타자가 느려 코드 제작에 오래 걸리는 경우를 줄임  \* 분당 타자수와 스코어를 표기해줌으로서 목표를 설정하고 목표를 성취하는 과정에서 학습에 재미를 붙일 수 있도록 유도함  \* 차후 기능 개선을 통해 다양한 프로그래밍 언어의 예약어 및 함수명을 학습함에 있어 도움이 될 것이며 나아가 어떤 단어를 입력할 때 특히 오래 걸리는지 등의 사용자 데이터 및 습관을 표시 할 수 있도록 함 | | | | | | | | | | |
| **애로사항 및 건의사항** | | |  | | | | | | | |
|  | | |  | | | | | | | |
| 1. 애로사항   \* 쓰레드의 작동 과정과 효율적으로 작업을 쓰레드로 나누기 위한 방법에 대해 공부가 필요하였음  \* 스레드 함수 구현에 있어 C언어의 경우 WinApi와 pthead를 통해 스레드를 구현할 수 있는데 각각 함수의 복잡성이 높고 리눅스 환경에서만 사용되기 때문에 프로젝트 진행에 앞서 어떤 것을 사용할지 결정할 것이 필요하였고 사용법이 간단하고 효율적인 C++의 thread 클래스를 사용함  \* 분당 타자수를 계산하는 과정에서 한글의 원활한 타자수 계산을 위해 유니코드로 변환하고 자모음을 분리하는 방법을 찾아야 했음  \* 이중 연결 리스트 사용에 있어 단일 연결 리스트와 차이점을 파악하고 코드에 적용하고 이중 연결 리스트가 단일 연결 리스트에 비해 어떤 부분에서 효과적인지 이해가 필요했음  \* 단순한 점수 계산 과정에서 사용자가 흥미를 잃을 가능성이 있으므로 콤보 제도를 도입하는 등의 방법으로 사용자가 보다 높은 점수를 받기위한 목표를 설정하였음   1. 건의사항   \* 프로젝트 간 자신의 실력이 어느 정도인지 파악할 수 있는 척도가 필요함  -> 자기가 잘한다고 생각했지만 실제와 다른 경우도 있고 반대의 경우도 있었습니다.  \*단순히 시험 점수나 과제 점수가 아니라 무언가를 직접 만들어본 경험이 있는 학생들을 조장으로 배정하는 것이 적합함  -> 제 개인적인 의견입니다만, 시험과는 별도로 코딩을 잘하는 애들이 섞여 있어 이들을 조장으로 배정하는 것이 좋다고 생각합니다. 제작 경험의 유무에 따라 최종 개발 결과물의 차이가 많을 것 같습니다. | | | | | | | | | | |