|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **주차** | 4-2 18주차 | **기간** | 2023.10.24  2024.10.30 | **지도교수** | (서명) |
| 이번주 한일 요약 | - 애니메이션 적용 TEST2  - codingtest 공부 | | | | |

작성자(학번 이름): 2018182037 조상준

팀명:

<작업 내용>

애니메이션 적용 및 테스트 완료 -> 테스트 후 아쉬운 부분 수정 필요

- 애니메이션 보충(다시 부활하는 부분, 무기Idle, 점프 등)

- 애니메이션 괴리감 수정(idle,walk 등)

스크린샷, 3D 모델링이(가) 표시된 사진

자동 생성된 설명

스크린샷, 예술이(가) 표시된 사진

자동 생성된 설명





- CodingTest 공부

- 약 40문제 정도 풀이 진행

- 난이도는 실버3정도까지

* Codingtest중 알게 된 내용

codingTest 공부를 하던 중에 **‘C++을 사용하고 있고 cin/cout을 사용하고자 한다면, cin.tie(NULL)과 sync\_with\_stdio(false)를 둘 다 적용해 주고, endl 대신 개행문자(\n)를 쓰자. 단, 이렇게 하면 더 이상 scanf/printf/puts/getchar/putchar 등 C의 입출력 방식을 사용하면 안 된다.’** 라는 글이 있길래 왜 그런지 알아보니 c의 stdio와 cpp의 iostream이 동기화가 되어있어 iostream과 stdio 버퍼를 모두 사용한다고 하여 딜레이가 발생

-> std::ios::sync\_with\_stdio(false); 추가하면 동기화를 비활성화시켜주어 c++만의 독립적인 버퍼가 생성되어 c의 코드와 병행하여 사용할 수 없지만 사용하는 버퍼의 수가 줄어들어서 실행 속도가 빨라지게 된다고 합니다.

std::cin.tie(NULL); 는 cin의 동기화를 해제하여 cout이 flush(플러시)되지 않게 하여 cout이 cin보다 먼저 flush되지 않아 성능이 향상된다고 합니다.

그래서 cin/cout을 사용하고자 하면

std::ios::sync\_with\_stdio(false);

std::cin.tie(NULL);

이 두 줄을 추가하고 시작하는 것이 성능 향상에 도움이 된다고 합니다.

|  |  |  |  |
| --- | --- | --- | --- |
| **문제점 정리** |  | | |
| **해결방안** |  | | |
| **다음주차** | 4-2 19주차 | **다음기간** | 2023.10.31~ 2024.11.6 |
| **다음주 할일** | * 중점연구 이어서 * Codingtest 공부 | | |
| **지도 교수**  **Comment** |  | | |