|  |  |  |  |
| --- | --- | --- | --- |
| [Cranding(크랜딩)] | | | |
| **4주차** | **2024.01.14~ 2024.01.20** | **작성자** | **신동엽** |
| **주간 회의** | | | |
| **회의 날짜** | | 2024.01.17 | |
| 1. 맵 제작  * 각각의 맵을 제작하고 씬전환 하기 * 기지 맵/자연 맵/얼음 맵/불 맵 * 기지 맵 제작시 시네마틱 효과를 위해 아래와 같이 맵 제작할것      * 각 지형 이동은 불가능하게 두고, 다른 지형으로 이동시 꼭 기지를 거쳐서 이동하도록 합시다. * 기지 → 각 씬으로 이동시 아래와 같은 화면이 보이도록 하는것이 목표      * 객체 랜덤 배치의 방법은 계산만 많아진다고 생각해서 보류   <일반 외계인, 보스 외계인, 플레이어의 크기 비교>     * 믹사모에서 찾아야하는 애니메이션 * 검 애니메이션 3개 → 연속 공격시 애니메이션 나오게 하기 * 기절 → <https://www.mixamo.com/#/?page=1&query=crawl&type=Motion%2CMotionPack> * 부활해서 일어나는 애니메이션 → <https://www.mixamo.com/#/?page=7&query=idle&type=Motion%2CMotionPack> * 총쏘는 애니메이션 → <https://www.mixamo.com/#/?page=1&query=gun&type=Motion%2CMotionPack> * 검으로 공격하는 애니메이션 1번 → <https://www.mixamo.com/#/?page=2&query=sword&type=Motion%2CMotionPack> * 검으로 공격하는 애니메이션 2번, 3번 → <https://www.mixamo.com/#/?page=2&query=sword&type=Motion%2CMotionPack> * 총 들고 있는거 → <https://www.mixamo.com/#/?page=3&query=idle&type=Motion%2CMotionPack> * 총 들고 이동하는거 → <https://www.mixamo.com/#/?page=1&query=rifle+run&type=Motion%2CMotionPack> * 검 들고 있는거 → <https://www.mixamo.com/#/?page=1&query=sword+idle&type=Motion%2CMotionPack> * 검 들고 이동하는거 → <https://www.mixamo.com/#/?page=2&query=sword&type=Motion%2CMotionPack> * 상하체 애니메이션 분리를 통해 총/검 들고 이동하는 애니메이션도 진행 | | | |
| **이번주 한 일** | | | |
| 1. 신동엽(서버)   서버 프레임워크   * 우선 클라이언트와 서버가 공동으로 네트워크용을 사용될 것들을 관리하는 ServerCore라는 라이브러리를 만들었다. 이를 통해 클라이언트와 서버 간의 통신을 하게 될 용도 * 따라서 ServerCore를 위주로 작업 * 소켓을 관리하는 SocketUtils 클래스를 구현 * 소켓의 옵션을 설정해주는 함수들과 네트워크 연결에 필요한 Listen Bind,를 함수로 따로 만들어서 사용하기 편하게 만들어 주었다. * Init에선 소켓을 초기화 해주고, SetLinger오 ReuseAddress에서는 소켓의 옵션을 설정한다. * 본격적인 IOCP * 우선 IOCP를 생성하고 등록시키고 Cp를 관리하는 Dispatch를 만들었다.   Extern을 사용해서 일단 전역으로 IOCPCore객체를 선언했지만 후에는 수정이 필요할 것 같다.  Session 역할을 하는 Listener 클래스를 생성     * OVERLAPPED EX 용도로 사용할 IocpEvent 클래스 구현   전체적인 흐름은 이렇다.  Listener(session)을 생성하고 주소와 포트번호를 입력받아 StartAccept함수를 실행한다.  accept함수에선 위에서 만든 SocketUtils를 통해 소켓을 생성하고 그 소켓을 iocp에 등록하고 네트워크 연결을 준비한다.    소켓을 리슨상태로 만든후 accept이벤트 상태를 만들어서 비동기 방식인 acceptex를 실행하며 일감을 iocp에 등록한다.  성공적으로 리스너가 생성되고 연결이 되었고 등록을 했다면 일을 처리할 쓰레드들을 생성하고 쓰레드들에게 일감을 처리하는 Dispatch함수를 반복 실행하여 일감이 있는지 계속 찾으며 처리한다.        차례로 함수가 불려 실행되면 Process함수까지 불려서 일감을 처리하게 된다.  이번주에 느낀 점 : 이 코드는 인프런의 서버강의를 기반으로 작성한 코드이다.  기존의 내 프레임워크는 main문 안에 연속적으로 작성하여 코드가 지저분하고 가독성이 많이 떨어진다고 생각했고 좀더 객체지향적으로 서버를 관리해야 효율적이라고 생각하여 서버 수업을 수강하며 같이 프레임워크를 따라 제작하는 중인데, 엄청 어려운 것 같다. 이렇게 프레임워크를 제작해야 하는지도 고민이 된다.  5주차   * 22일 ~ 25일 까지 일본여행.. * 26일 27일 알바 12시간 * 실제로 작업을 할 시간이 적은 관계로 가장 해낼수 있는것들을 먼저 해보자 * Connect가 오류가 나고 있는데 이부분을 해결해야한다. 어떤문제? 클라이언트에선 커넥트완료 표시가 나오지만 서버쪽에서 클라이언트 소켓을 식별하지 못하는 것 같다. 컨테이너에 들어오지 않음 * Connect 오류 해결 이후, Session을 제작하고 많이 진행 하면 동기화 3인 띄우기  1. 정경서(클라이언트)  * C++ 공부   상속 → 다른 클래스의 정보를 받아와서 사용할 수 있게 해주는 것      위에서 Derived의 생성자 호출 부분을 보면 기반의 생성자를 먼저 호출하여 기반이 되는 클 래스를 생성하고 Derived가 생성되도록 합니다.  위의 코드를 실행하면 먼저 기반 클래스가 생성되고 이후 파생 클래스를 생성할때 “기반 클 래스”가 먼저 출력된 뒤 “파생 클래스”가 출력되고 이후 Derived에서 Base 클래스의 what 함수를 호출하게 되는데 이때 Base 클래스의 s는 “기반”으로 정의되었기에 “기반”이 출력됩니다.  파생 클래스는 기반 클래스의 모든 정보를 상속받았기 때문에 기반 클래의 함수를 호출할 수 있습니다.  void what() { std::cout << s << std::endl; }  Derived 클래스에도 위와 같이 what 함수를 정의하고 코드를 실행 시킨다면 “기반” 대신 “ 파생”이 출력될것 입니다.  서로 다른 클래스에 정의 되어 있기에 다른 함수로 구분이 되고, Derived의 생성자가 같은 클래스에 있는 what 함수를 호출하기 때문입니다. 이를 오버라이딩이라고 합니다.  Derived의 what 함수가 Base의 what 함수를 오버라이딩 했다.  접근 지시자  protected → 상속받는 클래스는 접근 가능  private → 자기 클래스만 접근 가능  public → 어디든 접근 가능  class A : 000 B  000 자리에 들어가는 접근 지시자에 따라 기반 클래스의 정보들이 작동하는것에 영향을 줍니다.  protected → 파생클래스에서 public만 protected로 바뀜  private → 모든 접근 지시자들이 private가 됨  public → 기반 클래스의 접근 지시자들에 영향이 없다  파생 클래스에는 기반 클래스의 정보가 포함되어 있어서 오버라이딩을 할때 파생 → 기반 클 래스로 캐스팅 하는 것을 업캐스팅이라 부른다  다운 캐스팅 시에는 기반 → 파생 클래스로 캐스팅을 시도하게 되는데 기반 클래스에 파생 클래스의 내용은 없기때문에 오류가 발생할수 있어서 다운캐스팅은 함부로 시도하지 않는다   * 애니메이션   애니메이션을 적용하고 추출하는 문제에서 근본적인 원인을 해결하고 싶어서 시간을 조금 더 투자했다.  먼저 뼈 이름을 겹치는 부분이 없도록 모두 바꾸어주었다.  이후 프로젝트에서 애니메이션을 적용한 모델을 추출하였더니 모델이 찌그러지면서 그려지게 되는 문제가 발생했다.  이름을 바꾸는 방식에 문제가 있었다고 생각을 해서 잘 출력되는 플레이어 모델의 이름을 빌려 사용하였는데 계속 같은 문제가 발생하고 있었다.  또한 애니메이션이 제대로 출력되지 않는 경우도 발생하였는데, 이는 애니메이션 정보에 뼈 이름이 달라져서 그런거라고 생각되었다.  객체를 3D 오브젝트 위에 띄울때는 별다른 문제가 발생하지 않았기에 팀원들과 상의 후에 해당 부분은 넘어가기로 했다.     1. 홍예나(클라이언트) 2. 자료구조 vector, deque, list   <vector>     * 생성시 heap에 생성 & 동적으로 할당됨 * 배열과 같이 연속적이기 때문에 iterator와 position index로도 접근 가능함 * 동적으로 확장, 축소가 가능한 dynamic array로 구성됨 * 원소는 할당, 복사가 가능하다면 어떠한 타입도 가능함 * 랜덤 엑세스 반복자이므로 stl의 모든 알고리즘 사용 가능함 * 맨 끝의 원소 삽입, 제거 성능은 좋으나, 중간에 원소의 삽입, 제거 성능은 안좋음   (push\_back, pop\_back / insert, erase)   * 컨테이너의 크기 확장 시 비용이 크기 때문에 reserve와 같은 함수로 재할당은 피하는 것이 좋음 * 원소 값을 제거하기 위해서는 stl 알고리즘 사용해야함   <deque – double ended queue>     * 양방향에서 데이터를 처리(원소 추가, 삭제)할 수 있는 queue형 자료구조 * Rancom access iterator를 통해 개별 원소에 대해 접근 가능 * 개별 원소를 position index로 접근 가능 * 컨테이너의 동적 크기 조절 가능 * 원소의 앞, 뒤에서의 삽입, 제거가 모두 빠름 * 어떤 순서로든 원소의 순회가 가능 * 중간 위치에서의 삽입, 제거 수행시 성능이 떨어짐(list보다) * 메모리 공간이 연속적으로 할당되지 않아서(not contiguously) 원소들간의 포인터 연산이 불가능   <list – doubly linked list>     * 단순 연결리스트와 다르게 노드가 이전 노드와 다음 노드 두 가지로 구성되어있음 * 양방향 탐색이 가능하다는 큰 장점이 있음 * 랜덤 엑세스 지원 x * 원소의 position index로 직접 접근이 불가 * 이동 삭제 삽입이 빠름!! : 포인터만 이동만 요구해서 * 특정 원소를 찾기 위해서는 처음이나 끝에서 선형 탐색을 해야함 * 컨테이너의 어느 위치에서도 삽입, 제거가 빠름 * 원소들의 컨테이너 내 순서 이동이 빠름  1. 애니메이션 블렌딩   애니메이션 두 개를 동시에 enable하게 설정하면 렌더링되지 않는 오류가 발생함  결론적으로 원인은 한 번 바꿔준 bone weight를 다시 1.0f로 돌려놓지 않았기 때문인 것 같다  내가 애니메이션 블렌딩을 하려고 시도하고 있음에도 내가 생각하는 애니메이션 블렌딩의 개념이 정확한 것인지에 대한 확신이 부족했기에 개념을 다시 공부해 봄 + 코드 분석  <animation sets, animation track, key frame, bone frame>    <이번 회의 때 얘기가 나왔던 애니메이션 분할>    <애니메이션 블렌딩 개념>     * 두 애니메이션을 섞는 것에 집중하기 전에 1번 애니메이션의 끝나는 부분과 2번 애니메이션의 시작 부분의 보간에 집중하는 것이 더 좋을 듯 * GetSRT 함수를 이용하면 되지 않나????? * 코드 분석   <class CAnimationSet>         * Class CAnimationSets : animationset 클래스 모음 + boneframe 정보 포함   <class CAnimationTrack>   * 애니메이션 하나의 enable여부, speed, position, weight, callbackKey 등을 포함   <advance time 함수> | | | |
| **다음주 할 일** | | | |
| 1. 신동엽(서버)  * - 22일 ~ 25일 까지 일본여행.. * 26일 27일 알바 12시간 * Connect가 오류가 나고 있는데 이부분을 해결해야한다. 어떤문제? 클라이언트에선 커넥트완료 표시가 나오지만 서버쪽에서 클라이언트 소켓을 식별하지 못하는 것 같다. 컨테이너에 들어오지 않음 * Connect 오류 해결 이후, Session을 제작하고 많이 진행 하면 동기화 3인 띄우기      1. 정경서(클라이언트)  * 애니메이션 관련 프레임워크 수정 * 애니메이션 관련 추출 코드 수정  1. 홍예나(클라이언트)  * 애니메이션 블렌딩 구현 * 상속 추가 공부 | | | |
| **비고** | | | |
|  | | | |