|  |  |  |
| --- | --- | --- |
| **우정연** | 서버 | * 플레이어와 몬스터, 코인, 열쇠 사이의 충돌을 체크하고 그에 따른 오브젝트의 정보를 업데이트한다. (점수 증가, 오브젝트 삭제 등) * ~~게임 시작 시 서버 시간을 기록하고, 클라이언트로 전송한다.~~ * ~~플레이 시간을 계산해 클라이언트로 전송한다.~~ * ~~서버 시간을 통해 게임 종료 여부를 판단하고, 클라이언트로 전송한다.~~   **시간 관련은 이세민이 작성**   * 클라이언트에서 캐릭터 선택 여부 정보를 받아 캐릭터 중복 선택을 막고, 접속 인원 수를 판단해 게임 시작 여부를 클라이언트로 전송한다. * 몬스터 초기화 정보 전송한다. * 서버 동기화 |
| 클라 | * **플레이어 클래스를 구현한다.** * 몬스터 클래스를 구현한다. * 충돌처리를 위한 ~~구조체~~와 함수를 구현한다. * 서버로부터 플레이 시간 정보를 받아 화면에 출력한다. * 게임 플레이 전 캐릭터 선택 기능을 구현한다. * **선택한 캐릭터에 맞춰 캐릭터 애니메이션 변경** * **서버에서 받은 초기 정보를 기반으로 렌더링을 진행한다.** * 리소스 제작 * 캐릭터 선택창 |
| **이세민** | 서버 | * 클라이언트에게 ID 정보를 받아 기록한다. * 클라이언트의 키보드 입력을 받아 플레이어의 이동을 구현한다. * 플레이 시간 업데이트를 구현하고, 플레이 시간을 클라이언트로 전송한다. * 플레이 시간 및 승패, 게임 오버 여부로 게임 종료를 판단하고 승패 결과를 클라이언트로 전송한다. * **1/30초마다 클라이언트에게 게임 데이터 전송을 구현한다.** * **서버 동기화** |
| 클라 | * 플레이어 클래스를 구현한다. * 플레이어별 ID 입력 화면을 구현하고 ID 정보를 서버에 전송한다. * 클라이언트의 키보드 입력 정보를 서버에게 전송한다. * 게임 종료 판단을 서버에게 받고, 스코어에 따른 승패 여부를 화면에 출력한다. * 리소스 제작 * ID 입력 화면 리소스 * 승패 결과 화면 리소스 * **게임플레이화면 배경** |
| **정가온** | 서버 | * ~~발판, 코인, 몬스터, 캐릭터 정보를 초기화하고 클라이언트로 전송한다.~~ * **발판, 코인, 캐릭터 정보를 초기화하고 클라이언트로 전송한다.** * 클라이언트에게 받은 정보 수신한다. * **서버 동기화** * **클라이언트에게 받은 정보 수신하는 Recv\_Thread를 구현한다.** * 클라이언트에게 정보를 전송하는Send\_Thread를 작성한다.   서버에서 갱신된 데이터들을 클라이언트에게 전송한다.   * 몬스터 위치와 생존 유무 * 코인 획득 유무 * 코인 수에 따른 열쇠 렌더링 유무 * 키입력에 따른 플레이어 위치 |
| 클라 | * 코인 클래스를 구현한다. * 플레이어의 위치에 따라 카메라 스크롤링을 구현한다. * 플레이어 사망시 생존한 플레이어 시점으로 카메라 전환 기능 추가한다. * 서버에서 받은 초기정보를 기반으로 렌더링을 진행한다. * 코인 * 발판 * 플레이어 * 서버에서 받은 코인정보를 기반으로 렌더링 구현한다. |

변동사항

* 1. 스레드 관련 구현 뒤로 미룰 것
  2. 타이머 동기화 관련 구현 뒤로 미룰 것
  3. 데이터 구조체 변경 필요함

몬스터를 벡터로 관리할건데, 몬스터마다 일련번호가 있어서IsCollided 처리되면 몬스터 일련번호 반환. 코인 역시 벡터로 관리해 충돌 시 코인 일련번호 반환.

**몬스터 충돌 처리 시**

1. 머리를 짓밟았다(몬스터 사망 판정) -> 몬스터 일련번호를 서버에 넘기고 -> 점수도 올려야함

2. 머리가 아니다(플레이어 피격 판정) -> 체력을 깎고 -> 바뀐 체력을 서버에 넘겨야 함

따라서 몬스터 일련번호와 플레이어의 체력, 점수가 구조체에 포함되어야 함

**코인 충돌 처리 시**

충돌한 코인의 일련번호를 서버에 넘기고 -> 점수도 올려야 함

따라서 코인의 일련번호와 플레이어의 점수가 구조체에 포함되어야 함

**키 충돌 처리 시**

플레이어가 키를 가졌는지 여부를 서버에 넘기고 -> 점수도 올려야 함

따라서 플레이어 키 소지 여부, 점수가 구조체에 포함되어야 함

**포탈 충돌 처리 시**

플레이어가 포탈에 닿았는지 여부를 서버에 넘겨야 함

따라서 플레이어 포탈 충돌 여부가 구조체에 포함되어야 함

* 1. 구조체의 이름 변경 건

텍스트이(가) 표시된 사진

자동 생성된 설명

TODO

* + 1. 윈도우 포커싱 나가는 것 해결
    2. 구조체 협의해 이름과 형태 바꿔야 함
    3. 캐릭터 이미지 리소스 편집해야 함

**2019180025 우정연**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **11월** | | | | | | |
| **일** | **월** | **화** | **수** | **목** | **금** | **토** |
|  | | 1 | 2 | 3 | 4 | 5 |
|  |  | **기획서 1차제출** |  | 몬스터 클래스 수정 |
| 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 캐릭터 선택창 리소스 제작 |  | **기획서 2차 제출** |  | **기획서 3차 제출** | 충돌처리를 위한 ~~구조체~~ 함수 생성 | **기획서 3.5차 제출** |
| 13 | 14 | 15 | 16 | 17 | 18 | 19 |
|  |  |  | 15일까지  각 팀원들 구현한 내용들  검토 및 Merge | IsReady() |  | ~~AllReady()~~  스레드 구현 후 |
| 20 | 21 | 22 | 23 | 24 | 25 | 26 |
| ~~RecordTime()~~ | ~~SendThread~~ | ~~UpdateTime()~~  ~~서버 동기화~~ | 22일까지 각 팀원들 구현한 내용들 검토 및 Merge |  | IsCollidedMonster(), CollideMonster() | IsCollidedCoin(), CollideCoin() |
| 27 | 28 | 29 | 30 |  | | |
| IsCollidedKey(), CollideKey() | IsCollidedPortal(),CollidePortal() | ~~CheckGameOver~~  **SendThread** | **서버 동기화** |
| 12월 | | | | | | |
| 일 | 월 | 화 | 수 | 목 | 금 | 토 |
|  | | | | 1 | 2 | 3 |
| 문제 분석 및 디버깅 | | |
| 4 | 5 | 6 | 7 | 8 |  | |
| 최종 디버깅 | | | | 제출일 |

**2020182032 이세민**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **11월** | | | | | | |
| **일** | **월** | **화** | **수** | **목** | **금** | **토** |
|  | | 1 | 2 | 3 | 4 | 5 |
|  |  | **기획서 1차 제출** | 플레이어 클래스 정리,  UpdatePlayerInput() 키 이벤트 | UpdatePlayerLocation() 이동 |
| 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|  | ID 입력 화면 리소스 제작 | **기획서 2차 제출** | SendPlayerData 구조체 구현 | ~~UpdateSendData() 구현~~ | KeyInput 구조체 통신 구현 | **기획서 3.5차 제출** |
| 13 | 14 | 15 | 16 | 17 | 18 | 19 |
|  | ~~ID 정보를 담은 SendPlayerDdata 송수신 구현~~  ~~UpdateSendData() 구현~~ | ID 정보를 담은 SendPlayerData 송수신 구현 | 15일까지  각 팀원들 구현한 내용들  검토, merge  전송 시간 계산 함수 구현 |  | 승패 결과 화면 리소스 제작, ~~UpdateTime() 구현~~ | 각종 테스트 및 오류 수정 |
| 20 | 21 | 22 | 23 | 24 | 25 | 26 |
|  |  | **클라이언트에게 bIsPlaying 전송, 서버 동기화** | 22일까지 각 팀원들 구현한 내용들 검토 및 Merge |  | IsPlayerWinner() 승패 판별 함수 구현 및 송수신 | 승패 결과 화면 렌더링 구현 |
| 27 | 28 | 29 | 30 |  | | |
|  | 열쇠 획득 CheckGameOver() 구현 | **타임 오버에 따른 CheckGameOver() 구현** | 모든 플레이어 게임 오버 시 CheckGameOver() |
| **12월** | | | | | | |
| **일** | **월** | **화** | **수** | **목** | **금** | **토** |
|  | | | | 1 | 2 | 3 |
| 문제 분석 및 디버깅 | | |
| 4 | 5 | 6 | 7 | 8 |  | |
| 최종 디버깅 | | | | 제출일 |

**2020180034 정가온**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **11월** | | | | | | |
| **일** | **월** | **화** | **수** | **목** | **금** | **토** |
|  | | 1 | 2 | 3 | 4 | 5 |
|  |  | 기획서 1차 제출 | 코인클래스 구현  class Coin{} | 플랫폼 클래스 구현  platform{} |
| 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|  |  | ~~ScrollingCamera()~~;  **기획서 2차 제출** |  | **기획서 3차 제출** |  | **기획서 3.5차 제출** |
| 13 | 14 | 15 | 16 | 17 | 18 | 19 |
|  | InitPlatform() 구현 |  | 15일까지  각 팀원들 구현한 내용들  검토 및 merge. | PlayerMgr구조체 구현,  InitPlayer()  구현 | initMonster()구현  InitCoin() 구현 | initServer()  서버를 생성하여 플레이어의 초기 위치를 선정하고 서버 데이터를 초기화 |
| 20 | 21 | 22 | 23 | 24 | 25 | 26 |
| **클라이언트에게 SendGameData**  **정보전송** | ~~InitPlatform() 구현~~  **Recv\_Thread 구현** | ~~클라이언트에게 SendGameData~~  ~~정보전송~~  **서버 동기화** | 22일까지 각 팀원들 구현한 내용들 검토 및 Merge | 수신받은 데이터를 토대로 클라이언트 렌더링 수정 |  |  |
| 27 | 28 | 29 | 30 |  | | |
|  | ~~ChangeCamera()~~  ~~플레이어 사망시 카메라 전환 구현~~ | **Send\_Thread 구현** | 29일까지  각 팀원들 구현한 내용들  검토 및 merge. |
| **12월** | | | | | | |
| **일** | **월** | **화** | **수** | **목** | **금** | **토** |
|  | | | | 1 | 2 | 3 |
| 문제 분석 및 디버깅 | | |
| 4 | 5 | 6 | 7 | 8 |  | |
| 최종 디버깅 | | | | 제출일 |