|  |  |
| --- | --- |
| 게임명 : 날아라 날아라 | |
| 개발엔진 : Unity, 2022.3.4f1  장르 : 2D, 러닝 액션, 슈팅  개발기간 : 2024-7 ~ 2024-8  스크립트 URL : <https://github.com/h6011/Port1> | |
| 게임설명 | 투사체를 쏘거나 아이템을 써서 앞에 적을 물리치며 최대한 버티는 게임  (**드래곤 플라이트** 라는 게임을 영감을 받아서 만들어졌습니다)  플레이어는 **목숨**을 3개 가지고 게임을 시작합니다  wasd 키를 이용하여 움직입니다  **# 적 종류**  1. 기본 적  2. 보스  기본 적들이 1가지 패턴을 가지고 내려옵니다  기본 적들은 약 4 이상의 체력을 가지고 있습니다  그리고 투사체 같은건 쏘지 않고 그대로 내려옵니다  **# 기본 적들의 패턴**  1. 가로 일자형식 (ㅡ)  플레이어는 **투사체**를쏘아서 앞에 적을 잡으며 앞으로 나아갑니다  만약에 플레이어가 적에 닿는 다면 **목숨** 한개를 잃고  2초 동안 **무적** 시간을 가지게 됩니다  기본 적을 **쓰러트린다면**  **점수** 200점을 얻고  기본 적이 **죽는 다면**  총 **3가지의 재화**와 **아이템 2가지**를  다음과 같은 확률로 떨어 트립니다  **# 기본 적 재화 확률**  1. 코인 100% 확률로 3~8개를 떨어트림  2. 루비 40% 확률로 2~4개를 떨어트림  3. 다이아몬드 10% 확률로 1~3개를 떨어트림  4. 에메랄드 5% 확률로 1~2개를 떨어트림  **# 재화 종류**  1. 코인 (1원)  2. 루비 (10원)  3. 다이아 몬드 (20원)  4. 에메랄드 (30원)  **# 아이템이 뜰 확률 (5%)**  **# 아이템**  1. 투사체 강화 - 4 초 동안 투사체 **데미지 2배** 강화  2. 쉴드 - 플레이어에게 **추가 체력** 느낌으로 목숨을 하나 더 얻습니다  최대치는 1개입니다  3. 자석 - 6 초 동안 플레이어 주변에 재화를 **끌어 당깁니다**  **# 운석 장애물**  내려오기전에 경고 표시후 내려옵니다  운석은 적이 스폰될때나 보스가 스폰 됬을때  두가지 경우 모든 상황에서 내려옵니다  얼마나 많이 버텼냐에 따라 속도나 개수가 변경되면서 내려옵니다  적을 잡고 버티다 보면 1분 30초 마다  **보스 스테이지**가 시작 됩니다  보스는 다른 적들과 다르게 많은 체력과  **3가지의 총알 패턴**을구사합니다  **# 보스의 총알 패턴**  1. 라이플 방식  ㄴ 플레이어를 향해 약 10~15 발 를 발사합니다  2. 샷건 방식  ㄴ 플레이어를 향해 60 각도로 8~12 발을 발사합니다  3. 폭탄 방식  ㄴ 플레이어를 향해 조금 큰 2~3 개의 총알을 발사후  발사했을때의 플레이어의 위치에 도달 했다면 터지며  4~6개의 총알이 **랜덤**으로 퍼집니다  보스을 **쓰러트린다면**  **점수** 3000점을 얻고  게임이 계속 굴러 갑니다  이런식으로 **적과, 보스** 를 만나며 계속 **반복**되지만  점점 적들이 강해지는 식으로 게임이 진행됩니다  만약에 목숨을 모두 잃는 다면 **랭킹**에 점수와 시간이 기록되고  게임이 끝이 납니다 게임 재시작 버튼을 눌러 **다시 시작**을 할수 있습니다  (시간이 남는 다면 더 많은기능을 추가할 예정입니다)  1. firebase 데이터 랭킹  2. 상점 시스템 |
| 개발목표 | 1. 보스 총알 패턴 시스템  2. 랭킹 시스템  3. 아이템 시스템  4. 플레이어나 적이 화면 양옆 사이드 밖으로 나가지 않는 시스템  5. 웨이브 시스템  6. 재시작 시스템  7. 랭킹 데이터 난독화 |
| 인게임 이미지 사진 | |
| 기술 요약 | 관심을 끌기 위한 기술 소제목 |
| 알고리즘 설명 | |
| 발생했던 이슈 | 1.  2. |
| 해결 방안 | 1.  2. |
| 기술 요약 |  |
|  | |
| 발생했던 이슈 | 1.  2. |
| 해결 방안 | 1.  2. |

|  |  |
| --- | --- |
| 기술 요약 |  |
|  | |
| 발생했던 이슈 | 1.  2. |
| 해결 방안 | 1.  2. |
| 기술 요약 |  |
|  | |
| 발생했던 이슈 | 1.  2. |
| 해결 방안 | 1.  2. |

|  |  |
| --- | --- |
| 기술 요약 |  |
|  | |
| 발생했던 이슈 | 1.  2. |
| 해결 방안 | 1.  2. |
| 기술 요약 |  |
|  | |
| 발생했던 이슈 | 1.  2. |
| 해결 방안 | 1.  2. |

|  |  |
| --- | --- |
| 기술 요약 |  |
|  | |
| 발생했던 이슈 | 1.  2. |
| 해결 방안 | 1.  2. |
| 기술 요약 |  |
|  | |
| 발생했던 이슈 | 1.  2. |
| 해결 방안 | 1.  2. |

|  |  |
| --- | --- |
| 기술 요약 |  |
|  | |
| 발생했던 이슈 | 1.  2. |
| 해결 방안 | 1.  2. |
| 기술 요약 |  |
|  | |
| 발생했던 이슈 | 1.  2. |
| 해결 방안 | 1.  2. |