「Unitix Legends（UnitixLegends）」設計案

必須事項

1. ~~視点移動にはCinemachineを用いること~~
2. ~~Animatorの設定を行い、行動に合わせたアニメーションを再生せよ~~
3. 「クラスの継承」を意識せよ
4. ~~レベルデザインを行え（「ステージ３つに対して、Sceneも３つ作る」という従来の方法ではなくて「ステージ用の Scene は１つで、そこに各ステージのデータを読み込ませる」という風に、データベースと連動した応用を考えていけ）~~
5. ~~既存の学習内容を取り込め~~

ゲーム概要

タイトル	UnitixLegends
ジャンル	TPSアクションバトルロワイアル
モード	ソロ
環境	オフライン
舞台	住宅街
プレイヤー数	30（うち29がNPC）
ゲームの目的	最後の1人まで生き残ることを目指す
ウリ	全員で30人であるため、1試合が短くサクサク進められる （近年のバトルロワイアルゲームに見られる「試合時間の長期化に伴う暇な時間」をプレイヤーに与えない）
ゲームサイクル	• HPが0になったらゲームオーバー（他プレイヤーの観戦不可） • 最後の1人まで生き残ったらゲームクリア

ゲーム進行	1. 飛行機から降下 2. 地上でアイテムを収集 3. 戦闘（キルされたらゲームオーバー） 4. 最後の一人まで生き残る（ゲームクリア） 5. 画面が切り替わり、「1」へ移行（以降ループ）

アイテム

備考

- アイテムは全部で13種類ある
- アサルトとショットガンとスナイパーはそれぞれ2/9の確率で出現し、それら以外のアイテムはそれぞれ1/30の確率で出現する
- アイテムは最大5つまで保持可能で、画面下部のアイテムスロットにて「1~5」の数字をキーボードで入力することで、その数字（番号）に対応したアイテムを選択できる
- 捨てたアイテムは出現しない
- 弾を拾った場合、武器が無い状態でも弾の数は更新されて記録される

近接武器

名称	降下
Knife	一撃で相手のHPを100%削る
Bat	一撃で相手のHPを50%削る

飛び道具

名称	効果
Grenade	被爆者のHPを30%削る
TearGasGrenade	相手の動きを5秒間止める
Assault	1発で相手のHPを1%削る
Shotgun	1発で相手のHPを10%削る

Sniper	1発で相手のHPを50%削る

回復

名称	効果
Bandage（包帯）	HPが10%回復
MedicinalPlants（薬草）	HPが50%回復
Syringe（注射器）	HPが100%回復

弾

名称	内容
AssaultBullet	アサルト用弾（1000発）
SniperBullet	スナイパー用弾（10発）
ShotgunBullet	ショットガン用弾（100発）

スクリプト

スクリプト名	概要	処理内容	アウトプット内容

PlayerController.cs	プレイヤーの動作の制御を行う	~~移動(WASD)~~ ~~飛び降りる（Space）~~ ~~かがむ(E)~~ ~~アイテムの取得(Q)と廃棄(X)~~ ~~アイテム選択（1~5）と使用(左クリック)~~ ~~構える（右クリック）~~ ~~アニメーション(「走る」「かがむ」)~~	~~Animator~~ ~~enum を条件に利用した switch 文の実装~~
GameManager.cs	ゲーム全体の一連の制御を行う	~~ゲームスタート演出~~ ~~ゲームオーバー演出~~ ~~ゲームクリア演出~~	~~Startメソッドの戻り値を void から IEnumerator 型にして使用~~ ~~yield return StartCoroutine メソッドによる待機処理~~
CinemachineManager.cs	視点の制御を行う	~~飛行機からプレイヤーへの視点遷移~~ ~~視点操作~~	~~Cinemachine~~
BulletManager.cs	弾の生成と制御する	~~武器に応じた弾を選定~~ ~~弾を生成し発射~~ ~~発射後の弾の動きを制御~~	~~Transform.SetParent メソッド~~

ItemData SO.cs	アイテムのデータを管理	• ~~アイテムの名前~~ • ~~回復量~~ • ~~出現確率~~ • ~~弾の速さ~~ • ~~攻撃力~~ • ~~連射間隔~~ • ~~爆破までの時間~~ • ~~Sprite~~	• ~~クラス内に入れ子として enum を宣言して利用~~ • ~~リスト~~ • ~~スクリプタブルオブジェクト~~ • ~~クラス内に別のクラスを作成(入れ子クラス)~~ • ~~[System.Serializable（シリアライザブル）]属性~~ • ~~List を利用したスクリプタブル・オブジェクト内のデータの抽出処理~~
PlayerHealth.cs	プレイヤーのHPを管理	• ~~ダメージ処理~~ • ~~回復処理~~ • ~~回復アイテムの管理~~	
EnemyController.cs	敵の動作の制御を行う	• ~~付近のターゲットへの移動~~ • ~~アイテム取得と使用~~ • ~~ダメージ処理と演出~~ • ~~死亡処理~~ • ~~アニメーション~~	
EnemyGenerator.cs	敵の生成を行う	• ~~敵の生成~~ • ~~生成した敵の管理~~	• ~~リスト~~
ItemManager.cs	アイテムを制御	• ~~取得したアイテムの管理~~ • ~~アイテムの生成~~ • ~~アイテムの使用~~ • ~~アイテムの取得、破棄~~	• ~~GameObject型以外のインスタンス~~ • ~~リスト~~ • ~~Vector3.Scaleメソッド~~

GameData.cs	ゲームのデータを管理	~~落下速度~~ ~~キル数~~	~~シングルトン~~ ~~ゲームの設定関連の値を1つのスクリプトに集約し利用~~
StormController.cs	ストームを制御	~~ストームの収縮~~ ~~ストーム外に居るかどうかの判定~~	
AirplaneController.cs	飛行機を制御	~~飛行機の移動（43秒でマップを1周）~~ ~~飛行機のアニメーション（プロペラ）~~	
SoundDataSO.cs	音のデータを管理	~~音の種類~~ ~~音の名前~~	~~クラス内に入れ子としてenumを宣言して利用~~ ~~スクリプタブルオブジェクト~~ ~~リスト~~ ~~クラス内に別のクラスを作成(入れ子クラス)~~ ~~[System.Serializable（シリアライザブル）]属性~~ ~~Listを利用したスクリプタブル・オブジェクト内のデータの抽出処理~~
Soundmanager.cs	音の取得と再生処理	~~効果音の取得~~ ~~効果音の再生~~	

UIManager.cs	全てのUIを制御	~~ゲームスタート表示~~ ~~ゲームクリア表示~~ ~~ゲームオーバー表示~~ ~~フロート表示（取得アイテム、与えたダメージ）~~ ~~アイテムスロット表示（CanvasGroup）~~ ~~フレームレート表示（CanvasGroup）~~ ~~ミニマップ表示（CanvasGroup）~~ ~~残弾数表示（CanvasGroup）~~ ~~HP表示（CanvasGroup）~~ ~~照準（CanvasGroup）~~	~~Raycast Target機能（Rayの対象にするかどうか）~~ ~~Cull Transparent Mesh機能（透明時に描写しないかどうか）~~ ~~RawImage コンポーネント~~ ~~RenderTexture~~ ~~キャスト処理~~ ~~CanvasGroup コンポーネント~~ ~~DOFade メソッド~~ ~~DOText メソッド（GameOverで使用）~~ ~~for 文と Grid Layout Group コンポーネントを利用したインスタンシエイト処理~~

実装手順

1. ~~プレイヤーの動き（PlayerController.cs）~~ 8/11に実装済み
2. ~~視点の制御（CinemachineManager.cs）~~ 8/12に実装済み
3. ~~銃火器等の攻撃（BulletManager.cs）~~ 8/12に実装済み
4. ~~プレイヤーの体力を管理（PlayerHealth.cs）~~ 8/12に実装済み
5. ~~UI（UIManager.cs）~~ 8/13に実装済み
6. ~~アイテムのデータ（ItemDataSO.cs）~~ 8/14に実装済み

7. ~~アイテムの管理（ItemManager.cs）~~8/20に実装済み
8. ~~敵の動き（EnemyController.cs）~~8/21に実装済み
9. ~~敵の生成処理（EnemyGenerator.cs）~~8/21に実装済み
10. ~~ステージ~~8/21に実装済み
11. ~~ストームと飛行機の制御（StormController.cs、AirplaneController.cs）~~8/22に実装済み
12. ~~ゲームの進行（GameManager.cs）~~8/23に実装済み
13. ~~ゲームのデータの管理（GameData.cs）~~8/23に実装済み
14. ~~音の管理（SoundDataSO.cs）~~8/23に実装済み
15. ~~音の制御（SoundManager.cs）~~8/24に実装済み