

## Add fail condition (失败条件)

在 TowerDefenseGameState.cpp 中 BeginPlay 函数增加代码:

```
if (GameResultWidgetInstance) {  
    GameResultWidgetInstance->RemoveFromParent();  
    GameResultWidgetInstance = nullptr;  
}
```

在 Damage 函数中增加代码:

```
if (Health <= 0) {  
    if (UClass* DialogClass = LoadClass<UUserWidget>(nullptr, TEXT("WidgetBlueprint'/Game/TwinStick/DialogClass'"), nullptr)) {  
        if (APlayerController* PC = GetWorld()->GetFirstPlayerController()) {  
            GameResultWidgetInstance = CreateWidget<UUserWidget>(PC, DialogClass);  
            if (GameResultWidgetInstance) {  
                GameResultWidgetInstance->AddToViewport();  
            }  
        }  
    }  
}
```

在 TowerDefenseGameState.h 中增加代码:

```
UUserWidget* GameResultWidgetInstance = nullptr;
```

这几行代码主要是为了实现,游戏的结果,如果赢得话会弹出 You Win 的一个对话框,输了的话会弹出 You Lose 的一个对话框,然后判别方法 就是根据玩家的属性 Health(生命值)。

## Upgrade to UE 4.25 and add projectile test

将虚幻 4 游戏引擎的版本,从原来 4.23 版本升级到 4.25 版本,然后增加了炮弹的功能

新增 ArrowProjectile 类

继承虚幻 4 自带 AActor 类，主要利用了一写虚幻 4 自带的功能。然后还包含了 Init, BeginPlay, Tick 函数。

完善 EnemyBase 类

Cpp 中增加代码：

```
float AEnemyBase::TakeDamage(float DamageAmount, FDamageEvent const &DamageEvent, AController *EventInst
{
    Health -= (int32)DamageAmount;
    if (Health <= 0) Destroy();
    return DamageAmount;
}
```

如果敌人生命值减为 0，destroy

头文件 中增加代码：

```
int32 Health = 100;
```

```
virtual float TakeDamage(float DamageAmount, FDamageEvent const &DamageEvent, AController *EventInst
```

完善 StrongEnemy 类

```
Health = 80;
```

完善 ArrowTower 类

```

void AArrowTower::Tick(float DeltaTime)
{
    Super::Tick(DeltaTime);

    UE_LOG(LogTemp, Warning, TEXT("CNM"));
    if (Timer <= 0) {
        // 搜索范围内的敌人
        for (TActorIterator<AEnemyBase> It(GetWorld()); It; ++It) {
            FVector Dir = It->GetActorLocation() - GetActorLocation();
            UE_LOG(LogTemp, Warning, TEXT("Cnm %s"), *It->GetName());
            if (Dir.Size() <= Range * 100) {
                // 发射子弹
                GetWorld()->SpawnActor<AArrowProjectile>(GetActorLocation(), Dir.GetSafeNormal().Rotation);
                break;
            }
        }
    }
}

```

在 Upgrade 函数中增加代码：

```
GetWorld()->GetGameState<ATowerDefenseGameState>()->Money -= Cost;
```

减去升级所需的费用

完善 BigEnemy 类

设置 BigEnemy 属性 Health = 200, Speed = 60, Damage = 3;

这里的 Damage 指的是它对玩家的伤害，不是防御塔。

## Finish arrow projectile

完善 ArrowProjectile 类

C++ 中修改代码：

将原来子弹初速度 300.f 改成 500.f

```

ProjectileMovement->InitialSpeed = 500.f;
ProjectileMovement->MaxSpeed = 500.f;

```

原来 3.0 改成 1.5

```
InitialLifeSpan = 1.5f;
```

完善 ArrowTower/BombTower 类

限制了塔的升级：

```
int32 RangeArr[8] = { 0, 150, 200, 250, 300, 350, 500 };  
int32 CostArr[8] = { 0, 10, 12, 15, 18, 22, 10000000 };
```

修改了升级的花费

完善 BigEnemy 类/smallEnemy 类/StrongEnemy 类

修改了 BigEnemy 的生命值，从原来 200 改为 50，因为太 bug 了

增加了 smallEnemy 的属性，生命值

修改了 StrongEnemy 的生命值，从原来 80 改为 20

**Finish bomb projectile**

新增 BombProjectile 类，这个是为 BombTower 准备的子弹。

这个和 ArrowProjectile 类类似，利用了一些虚幻 4 的功能，建立模型，创建网格，然后把模型放进网格中……

包含了 BeginPlay, EndPlay, Tick, Init 函数。

完善 BombTower 类

和 ArrowTower 类一样,增加了搜索范围类的敌人,发射子弹的功能。

限制了炸弹塔的等级上限

修改初始状态 TowerDefenseGameState

初始金币从 10 修改为 100

**Fix money display bug**

修复了 money 展示的 bug

在 AArrowTower/BombTower 的 Upgrade 函数中

将原来的代码改成这样

```
GetWorld()->GetGameState<ATowerDefenseGameState>()->Cost(Cost);
```

新增 EnemySpawner 类

这个类是用来生成敌人的。

```
private:
    float Timer = 3;
    char EnemySeq[32] = { 'a', 'a', 'a', 'a', 0, 0, 0, 0, 'b', 'b', 0, 0, 0, 0, 'a', 'a', 'c', 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 };
    int32 NextEnemyIt = 0;
```

每隔 3 秒出一个兵, a 表示 smallEnemy, b 表示 bigEnemy, c 是

StrongEnemy

完善 TowerDefenseGameState 类

新增代码：

```
if (bCanWin && !TActorIterator<AEnemyBase>(GetWorld())) {
    if (UClass* DialogClass = LoadClass<UUserWidget>(nullptr, TEXT("WidgetBlueprint'/Game/TwinStick/")) {
        if (APlayerController* PC = GetWorld()->GetFirstPlayerController()) {
            GameResultWidgetInstance = CreateWidget<UUserWidget>(PC, DialogClass);
            if (GameResultWidgetInstance) {
                GameResultWidgetInstance->AddToViewport();
            }
        }
    }
}

void ATowerDefenseGameState::Cost(int32 Cost)
{
    Money -= Cost;
    ((UPlayerInfoUserWidget*)PlayerInfoWidgetInstance)->Money = Money;
}

void ATowerDefenseGameState::Win()
{
    bCanWin = true;
}
```

判断：如果最后一个敌人已经出了，然后场景里的所有怪都死了，玩家的生命还大于 0 则弹出对话框胜利……

在头文件中将初始金币从 100 改成 50，然后新增了 cost 函数和 win 函数

Finish strong enemy attacking tower

给 strongenemy 增加了攻击的功能

新增 StrongEnemyBullet 类

和前面的子弹类基本相似

完善 StrongEnemy 类

和前面箭塔炸弹塔相似，包含了搜索范围内的防御塔，发射子弹的功能

**Fix game lose bug**

基于上面，游戏基本成型，但是在测验游戏的时候发现了一个 bug：敌人全部通过玩家的基地时，玩家已经输掉游戏，但是当最后一个敌人通过基地时，还是会出出现 You Win 的对话框。

将 TowerDefenseGameState 中 45 行代码修改成如下代码：

```
if (bCanWin && !GameResultWidgetInstance && !TActorIterator<AEnemyBase>(GetWorld()))
```

**Add sound**

增加 音效加入了新的头文件 “Kismet/GameplayStatics.h”，然后还有一些虚幻 4 的操作