### Прототип Танков

#### Задача

Используя материал первого модуля, и ассеты, предоставленные для разработки. Создать репозиторий с проектом в Unity, в котором собрать по ТЗ прототип танков.

#### Техническое задание

Репозиторий:

1. Создать учебном репозитории ветку для разработки прототипа танка;
2. Добавить в ветку правильно настроенный .gitignore, пустой 3D Unity проект;

Проект:

1. Установить в проект пакеты TextMeshPro и InputSystem, разработанные Unity Technologies;
2. Установить в проект пакет Zenject, размещённый в AssetStore;
3. Импортировать основу прототипа танка;

Префаб танка:

1. Собрать префаб танка, используя модель из пакета;
2. Добавить на корневого объект танка компоненты:
   1. Rigidbody;
   2. MeshCollider с режимом Convex;
   3. TankController и провести настройку;
   4. PlayerInputController и BattleRam (используется для добавления очков при таране танком). Провести настройку;
3. Настройка объекта TankFree\_Tower:
   1. Разместить дочерним объектом TankFree\_Canon;
   2. Добавить в TankFree\_Canon объект FirePoint. Корректно спозиционировать относительно визуальной части модели;
   3. FirePoint объекту добавить LineRenderer и ProjectileTracker компоненты. Провести настройку;
   4. Добавить в FirePoint объект SmokeEffect и MuzzleFlash01 экземпляры префабов. Корректно спозиционировать относительно визуальной части модели;
   5. Добавить компонент TurretController к TankFree\_Tower, провести настройку;
4. Настройка ходовой части:
   1. В корень танка добавить объект Wheels. Вложить внутрь четыре объекта с названиями LeftFront, RightFront, LeftRear, RightRear;
   2. Каждому вложенному объекту добавить компонент WheelCollider, провести настройку и ориентацию относительно визуальной модели;
   3. На каждый объект с визуалом колеса добавить компонент Wheel, провести настройку;
   4. Создать пресет TransmissionSettings в папке Assets/Settings. Провести настройку минимум для трех передач;
5. Добавить в корневой объект новый объект с компонентом TankAudioSystem, провести настройку:
   1. Добавить AudioSource и прокинуть ссылку на него в TankAudioSystem;
   2. Установить в поле WorkingAudioClip ссылку на звук MotorAudio;
   3. Выставить поле MaxTankSpeed в значение чуть ниже максимальной скорости танка;

Префаб снаряда:

1. Используя модель, собрать префаб снаряда;
2. Добавить на корневого объект снаряда компоненты:
   1. SphereCollider;
   2. Rigidbody с настройками Interpolate и Continuous Dynamic;
   3. Projectile и настроить;
3. Вложить в корневой объект визуальную модель снаряда;
4. Добавить визуальной модели компонент TrailRenderer и настроить;
5. В префаб Installer/ProjectilePool в компоненте ProjectilePool установить ссылку на префаб снаряда;

Камера:

1. Установить пакет Cinemachine, разработанный Unity Technologies;
2. Добавить на камеру компонент CinemachineBrain;
3. Создать пустой объект и добавить компонент CinemachineVirtualCamera;
4. Провести настройку CinemachineVirtualCamera:
   1. Follow - следит за башней танка;
   2. Look At - следит за FirePoint ствола танка;
   3. Камера должна располагаться и ориентироваться за танком так, чтобы при наведении модель танка не перекрывала большую часть обзора;

Управление:

1. Для управления должен использоваться сгенерированный файл TankControls.cs;
2. Настройки управления необходимо расположить в папке Assets/Settings;
3. InputActions должен содержать карты:
   1. Tank содержит инпуты:
      1. Movement типа Vector2 с настройкой срабатывания на WASD;
      2. Handbreak с настройкой срабатывания на Space;
   2. Turret содержит инпуты:
      1. Focus типа Vector2 с настройкой срабатывания на delta Mouse;
      2. Fire с настройкой срабатывания на Q;

Спидометр:

1. Создать корневой объект с именем Speedometer на основе Image компонента. Добавить компонент Speedometer;
2. Разместить внутри корневого объекта TextMeshPro объект и установить ссылку на него в компоненте Speedometer;
3. В качестве задания с плюсиком, можно реализовать циферблат со стрелкой и аналоговой шкалой скоростей, в противном случае, для корректной работы Speedometer - создать пустой игровой объект внутри корневого объекта и прокинуть ссылку на него в поле Arrow компонента Speedometer;
4. Расположить спидометр слева снизу экрана, корректно привязать к нижнему левому краю;

Счетчик разрушения:

1. Создать пустой корневой игровой объект с именем CarnageIndicator. Добавить компоненты CarnageIndicator и AudioSource. В поле AudioClip у AudioSource прокинуть ссылку на звук TickerSum\_Sound. Установить низкую громкость и низкий приоритет источника звука;
2. Разместить в корневом объекте Image объект с именем IndicatorBackground. В качестве источника использовать спрайт progressBar4;
3. Разместить в объекте IndicatorBackground объект Indicator с компонентом Image. В качестве источника использовать спрайт progressBar4;
4. Корректно настроить заполнение Indicator в IndicatorBackground.Шкала должна заполняться слева-направо;
5. В каждой визуальной области шкалы разместить TextMeshPro по форме и размеру области. Корректно и одинаково настроить размер шрифта. Каждому объекту дать названия Step1, Step2 и т д;
6. Под шкалой разместить объект Message с компонентом TextMeshPro. Корректно настроить визуальное отображение сообщения под шкалой. Добавить компонент Animation;
7. Расположить IndicatorBackground по центру сверху экрана, корректно привязать к середине верхней границы канваса;
8. Создать Image объект с именем PointsRoot в корневом объекте. Использовать в качестве источника спрайт circlefillRy86px512px. Корректно настроить отображение с урезанным краем, так чтобы текст располагался в разрезе;
9. Создать два TextMeshPro объекта с именами SumText и CurrentPointsText и добавить их в PointsRoot. Расположить их друг под дружкой в визуальном разрезе объекта PointsRoot;
10. Создать в Assets/Settings скриптабл с названием Indicator, используя контекстное меню Create/Settings/Indicator. Для настройки сеттинга:
    1. Создать шесть элементов;
    2. Каждому элементу настроить поле Mult по принципу “последующий всегда больше предыдущего”;
    3. Первому элементу поля Text и Clip оставить пустыми;
    4. Со второго элемента в поля Text прописать любые короткие текстовые сообщения. Последнему элементу установить поле Text в значение “CARNAGE!!!”;
    5. Для всех элементов, начиная со второго, создать анимации, используя Animation редактор в Unity. Все анимации, не считая, последней, должны содержать минимум ключи изменения свойства вращения вокруг оси Z;
    6. Для последнего элемента анимация должна минимально содержать ключ вращение вокруг оси Z и ключ изменения цвета от белого/желтого до красного цвета шрифта;
11. Настроить компонент CarnageIndicator, используя ссылки на созданные объекты из пунктов выше. Массив IndicatorUpdateClips заполнить ссылками на звуки whoosh1\_Sound и whoosh2\_Sound. Корректно настроить IndicatorGradient и прочие поля;

Индикатор перезарядки:

1. Создать новый канвас с названием Canvas3D. Поле RenderMode задать значением World Space;
2. Разместить в канвасе Image, в качестве источника задать спрайт circlefillRy32px256px. Корректно настроить заполнение изображения по часовой стрелки;
3. Поместить на канвас компонент ReloadIndicator. Корректно настроить:
   1. Поле Indicator задать компонент Image;
   2. В поле Color задать градиент, со следующими ключами: 0% = (1f, 0f, 0f), 50% = (1f, 1f, 0f), 100% = (0f, 1f, 0f), где в скобках значения RGB;
4. Расположить канвас на башне танка таким образом, чтобы при запуске игры, индикатор был хорошо различим;

Префабы уровня:

1. Создать префаб SingleElement и расположить в Assets/Prefabs/Blocks. Настроить:
   1. Визуал в виде куба базового цвета;
   2. Компонент BoxCollider с установленным физматериалом;
   3. Параметры физматериала: FrictionCombine = Average, BounceCombine = Average, Bounciess >= 0.3, StaticFriction = 0.6, DynamicFriction <= 0.6;
   4. Компонент Rigidbody с параметрами: Mass >= 10, Drag & AngularDrag >= 0.1, Interpolate = Interpolate, CollisionDetection = Continuous Dynamic;
2. Создать в той же директории составной префаб GreenBlock из SingleElement:
   1. Четыре SingleElement прилегают друг к дружке по одной высоте;
   2. Перекрасить элементы в зелёный цвет;
3. Создать в той же директории составной префаб RedBlock из SingleElement:
   1. Четыре SingleElement располагаются один за другим по одной координатной оси;
   2. Перекрасить элементы в красный цвет;
4. Собрать на основе созданных блоков какую-либо конструкцию, которая суммарно будет насчитывать не менее 40 (сорока) SingleElement. Конструкция может быть произвольной формы (в виде башни, моста, домика или стены) и не должна содержать отдельных SingleElement (только в составе других префабов);

Уровни для тестирования:

1. Создать уровень для тестирования ходовой части, на котором:
   1. Присутствует пол, вытянутый по оси Z не менее чем на 500 метров;
   2. Танк располагается по направлению трассы на одном её краю;
   3. Вдоль трассы располагаются какие-либо легко различимые объекты. Их задача - упростить визуальное ориентирование относительно уровня при движении танка на большой скорости по трассе;
2. Создать уровень для тестирования боевых возможностей, на котором:
   1. Присутствует ограниченный пол, минимально размером 500х500 метров;
   2. Танк располагается ближе к центру и ориентирован на заполненную объектами часть уровня;
   3. На уровне должна располагаться композиция из префабов блоков. Суммарное кол-во SingleElement должно быть не меньше 150 штук