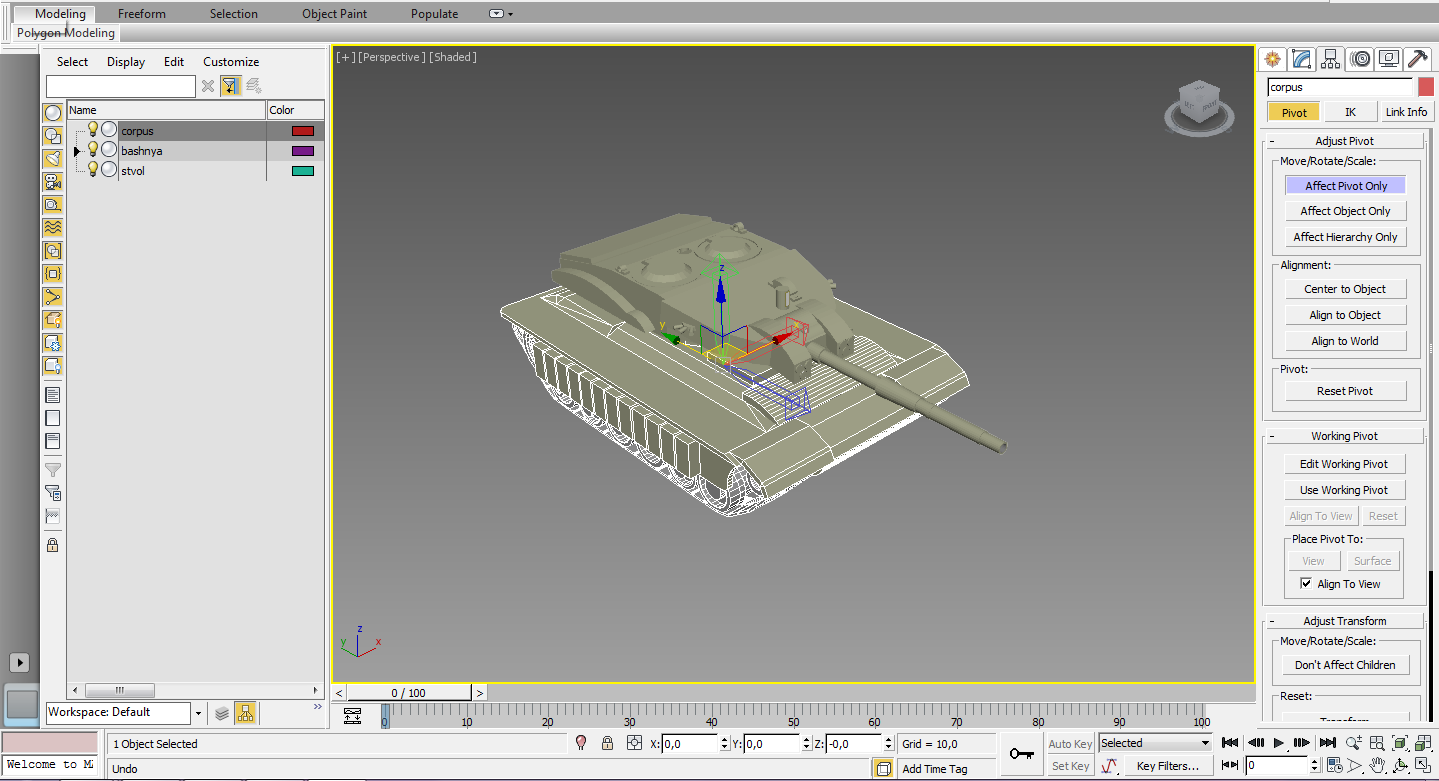
**Тема 14. Управление 3D-моделью танка в среде Unity**

1. Отредактировать в **3ds MAX** созданную по **Теме 9** на основе примитивов модель танка из трех составляющих частей: корпус, башня, ствол и дать им по отдельности соответствующие имена.
2. При создании танка необходимо изменить направление осей координат точки привязки **Pivot** всех его частей в соответствии с рисунком для корпуса и аналогично для башни и ствола, чтобы направление оси **Y** совпадало с осями в редакторе **Unity** (подробнее см. **Лекцию 13**).



1. Отредактировать поверхности отдельных частей модели танка для придания схожести с реальным современным танком. Наложить соответствующие текстуры на объекты, составляющие модель танка.
2. Сохранить в среде **3ds MAX** модель танка в формате **3DS** (или **FBX)** для последующего импорта в среду **Unity** командой **Export>**, а также исходник в формате **max** для возможной коррекции исходной модели танка.
3. Создать в среде **Unity** холмистую поверхность **Terrain** с плоской частью в центре и холмами (горами) по краям.
4. Импортировать модель танка в среду **Unity,** скопировав файл с танком в формате **3DS** в папку **Assets** проекта**,** при этом в **Assets** будет создан объект в виде префаба танка**,** и затем разместить его перетаскиванием префаба танка на сцену в начале плоской части территории.
5. Создать иерархию подчиненности частей танка в **Unity** наложением в окне **Hierarchy** при перетаскивании мышью объектов: ствол на башню, башню на корпус в соответствии с теми именами, которые были присвоены этим объектам в **3ds MAX** перед импортом в среду **Unity**. При этом можно убрать из иерархии объект – весь танк, т.е. имя префаба танка, оставив только составляющие его компоненты: корпус, башню и ствол.
6. Добавить танку в Инспекторе компонент **Rigidbody** и задать ему массу *(****Mass****)* около 1000.
7. Добавить скрипт как компонент в модель танка на верхний объект иерархии, в нашем случае это его корпус и создать в нем программный код для управления движением танка и вращением его отдельных компонент, используя для этого конструкцию **Input.GetAxis()** для клавиш и курсора мыши, а также материалы **Лекции 14**.
8. Запустите приложение, нажав кнопку **Play,** и проверьте работу созданного сценария управления танком. Для реалистичного передвижения возможно будет необходимо отрегулировать скорость и массу танка.
9. Добавьте на сцену в горизонтальной части поверхности напротив танка 3-5 объектов окружения из примитивов Unity и раскрасьте их разными цветами или наложите на них различные текстуры.
10. Добавьте для объекта башня скрипт, который генерирует падение «бомб» на танк, действуя аналогично работе по **Теме 12**.

*Используйте для генерации растянутую сферу – примитив* ***Sphere****, для масштабирования объекта применяйте метод* ***localScale****.*

*Задайте для генерируемых объектов соответствующий цвет.*

*Для расчета области падения выберите площадку размером вдвое больше размеров танка с разбросом случайным образом с помощью конструкции* ***Random.Range()*** *и**центром вблизи танка (ориентируйтесь на координаты центра танка на сцене в компоненте Инспектора* ***Transform/Position****).*

1. Запустите игру и, генерируя нажатием заданной клавиши падение «бомб», попробуйте объехать зону бомбежки.
2. Подчините камеру башне, подберите для ее положения (сзади и чуть выше башни, возможно поверните в вертикальной плоскости) выгодный ракурс просмотра и «поездите» танком по поверхности расталкивая объекты-цели.
3. Добавьте к камере скрипт, который при попадании из камеры луча (направление будет соответствовать повороту башни в сторону «цели», поскольку камера подчинена башне) в «цель» - цилиндр или капсулу, будет заставлять их «убегать» (сдвигаться) от танка.

*Используйте для этого скрипт, рекомендованный в* ***Лекции 13*** *для обработки попадания луча от камеры в «стенку» с генерацией в месте попадания сферы, но теперь вместо генерации сферы в месте попадания луча должен запускаться метод, заставляющий цель двигаться. Для преобразования кода скрипта используйте материалы* ***Лекции 13****.*

*Для обработки попадания луча в несколько целей необходимо добавить для каждой цели скрипт обработки, например,* ***run1, run2****, ... и создать в скрипте камеры переменные типа* ***target1, target2,*** *… с запуском для каждой цели обработки ее поведения на основе обращения к скрипту, соответствующему каждой цели* ***run1, run2****, ...*

1. Запустите игру и заставьте цели «убегать» при попадании их в «прицел» танка.