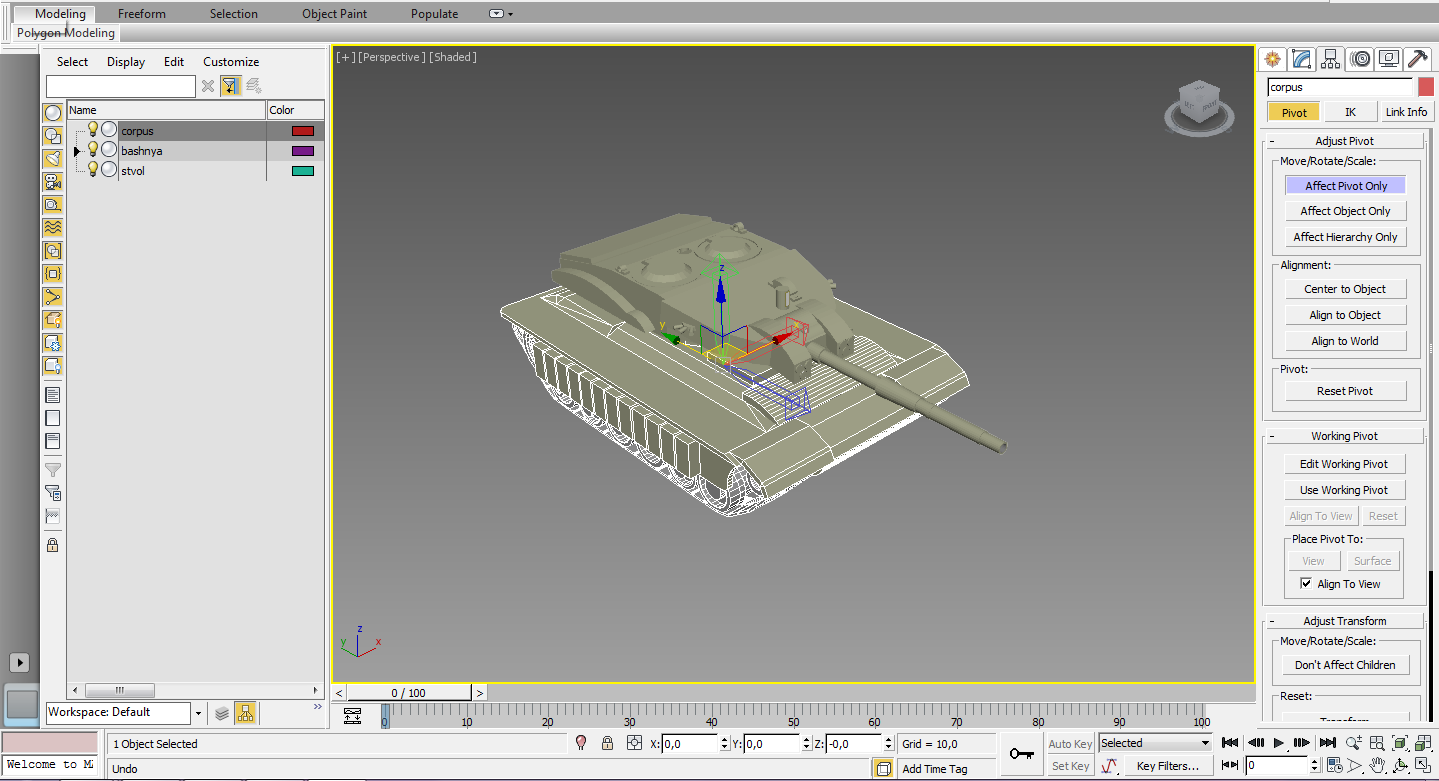
**Тема 9-10. Создание модели танка в 3ds Max и управление танком в среде Unity**

1. Создать в **3ds MAX** на основе примитивов (**Box, Cylinder**) модель танка из трех составляющих частей: корпус, башня, ствол и дать им по отдельности соответствующие имена.
2. Отредактировать поверхности модулей танка в режиме **Edit Mesh/Edit Poly** (щелкнуть правой кнопкой мыши по объекту) для придания им более реалистичный вид, похожий на представленный на рисунке ниже.



1. После создания модели танка необходимо изменить направление осей координат точки привязки **Pivot** всех его трех модулей, для чего выполнить команду **Hierarchy/Affect Pivot Only** в соответствии с рисунком для корпуса и аналогично для башни и ствола, чтобы направление оси **Y** в **3ds MAX** (горизонтальное, как показано на рисунке) совпадало с направлением оси **Y**, принятом в редакторе **Unity** (вертикальное), для чего нужно при включенном режиме вращения повернуть оси **Pivot** на 900 вокруг оси **X** (подробнее см. **Лекцию 4**).
2. Отредактировать поверхности отдельных частей модели танка для придания схожести с реальным современным танком. Наложить соответствующие текстуры на объекты, составляющие модель танка.
3. Экспортировать в среде **3ds MAX** модель танка в формате **fbx** для последующего импорта в среду **Unity** командой **Export>**, а также сохранить исходник в формате **max** для возможной коррекции исходной модели танка.
4. Создать в среде **Unity** плоскую поверхность из объекта **Plane** и нанести на нее соответствующую текстуру (песок, трава, …).
5. Импортировать модель танка в среду **Unity,** скопировав файл с танком в формате **fbx** 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 плоскости и анести на все его части соответствующую структуру.

*Если импортированный танк не соответствует размерам созданной плоскости, необходимо отредактировать значение* ***Scale Factor*** *танка, выбрав предварительно его в окне* ***Assets****.*

*Если текстура легла неправильно для объекта (например, повторяется много раз), необходимо изменить значение* ***Tilling*** *в материале текстуры, выбрав ее в окне* ***Assets****.*

1. 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.
2. Добавить танку в Инспекторе компонент **Rigidbody** и задать ему массу – параметр **Mass**около 1000.
3. 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 материалы **Лекции 4**.
4. Запустите приложение и проверьте работу созданного сценария управления танком при движении его по плоскости (для реалистичности передвижения и поворотов возможно будет необходимо отрегулировать их скорости).
5. Для организации ***взаимодействия танка с объектами сцены*** добавьте на плоскости напротив танка три объекта окружения: тонкую высокую «стенку» (**Box**), вытянутые цилиндр (**Cylinder**) и капсулу (**Capsule**). Добавьте каждому из объектов компоненту **Rigidbody** и раскрасьте их разными цветами или наложите на них текстуры.
6. Перед «стенкой» разместите триггерный коллайдер и добавьте для него скрипт на основе функции **OnTriggerStay**(), который заставляет вращаться стенку вокруг вертикальной оси, если в триггере находится танк.
7. Разместить на сцене два точечных источника света типа **Point** красного и синего цвета с начальной интенсивностью **Intensity** равной 0 и увеличенной **Range** областью действия равной примерно 100.
8. Разместите по краям сцены два триггерных коллайдера и создайте для каждого из них по скрипту с использованием двух функций **OnTriggerEnter()** и **OnTriggerExit()**, которые при входе в них танка будут включать красный или синий цвета дополнительной освещенности сцены с заданием для них параметра **Intensity** равным примерно 10 и соответственно выключать при выходе танка из соответствующих триггерных коллайдеров с заданием параметра **Intensity** равным 0, создавая тем самым вспышки красного или синего цвета на всей сцене при входе танка в соответствующий коллайдер.
9. **Этот пункт задания выполнить факультативно**. Добавьте для танка скрипт, который генерирует падение «бомб» на танк при нажатии на определенную клавишу, например **Q.** Используйте для генерации «бомб» из префаба примитив **Capsule** красного цвета со скриптом исчезновения при достижении поверхности плоскости. Для расчета области падения задайте площадку размером вдвое больше размеров танка с разбросом мест падения случайным образом с помощью конструкции **Random.Range()** ицентром вблизи танка (ориентируйтесь на начальные координаты центра танка на сцене в компоненте **Transform/Position** танка) и расположите ее прямо по ходу движения танка вперед. Запустите игру и, генерируя нажатием заданной клавиши падение «бомб», попробуйте объехать зону бомбежки.
10. Добавьте звуки на сцену: 1) фоновый звук сцены, 2) долгий звук при движении танка по сцене, 3) краткий звук при падении «бомб» на плоскость. **Ориентируйтесь на материалы Лекции 5а.**