

# Top-down Space Shooter Tutorial

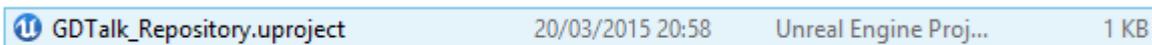
Supporting documents for Unreal 4 tutorial by Matt Blickem & Ben Wilson

This document accompanies a tutorial for building your first game in Unreal 4. I've printed out some copies for you but there's also hyperlinks that you might find useful. To find the digital version of this document go to <http://bit.ly/1Bnp3x9>.

## Opening the tutorial project

You've been provided with a jump-start project for this tutorial! This project has all the assets to build our game with, except for the blueprints that define how our game will behave. This includes the level, materials and models. Here's how to open it:

1. Download the project: <http://bit.ly/1LYrlru>
2. Unzip the project files to C:\Users\\Documents\Unreal Projects\ (or anywhere)
3. Double click *GDTalk\_Repository.uproject* to open the project in Unreal 4.7

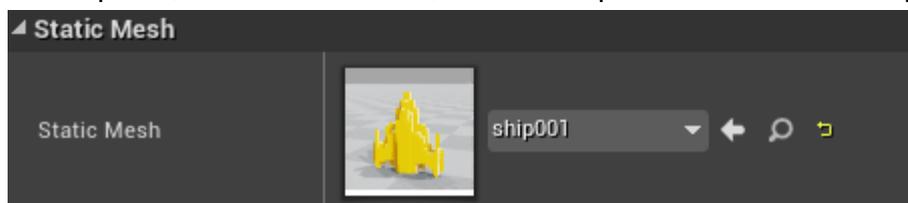


## Building the player

The player is made up of two blueprints: a **Pawn** and a **PlayerController**. In general terms with the [Unreal Framework](#), player input is handled in a PlayerController blueprint which possesses a Pawn. Pawns can be possessed by PlayerControllers or AIControllers.

For this tutorial we're building a player Pawn and an associated PlayerController, but for simplicity all the player input work that we do will be in the Pawn blueprint instead of the PlayerController.

1. Create a folder in the Content Browser at called Content/Blueprints
2. Create a new Blueprint Class, and in All Classes search for and choose **DefaultPawn** as the parent class.
3. Name the blueprint something like **BP\_PlayerPawn**
4. Open your new Pawn blueprint (double-click)
5. In the Components panel we have a MeshComponent. Click this component to view and edit it's properties. We want to use a spaceship for our static mesh.
6. In the Details panel, under Static Mesh, use the dropdown list to select **ship001** as your static



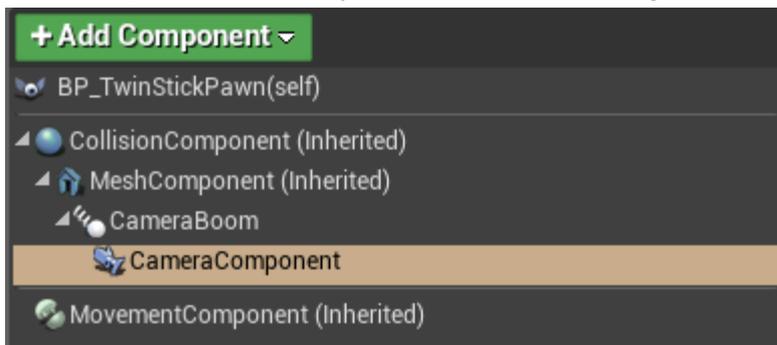
mesh.

7. The ship scale and rotations are wrong! You can blame Matt for that. To fix this set the Transform properties to: Location(10, 0, -40), Rotation(90, 0, -90) Scale (1.0, 1.0, 1.0)
8. One last tweak to the MeshComponent: set Collision Presents to NoCollision.
9. We also need to tweak some settings in the MovementComponent:
  - a. FloatingPawnMovement>Acceleration: 8000
  - b. FloatingPawnMovement>Deceleration: 3500
  - c. PlanarMovement>Constrain to Plane: True
  - d. PlanarMovement>Snap to Plane at Start: True
  - e. PlanarMovement>Plane Constraint Axis Setting: Z

After we've defined player input you might enjoy tweaking the acceleration and deceleration.

Next we're going to attach a top-down chase camera to the ship. We'll do this by holding the camera above the ship with a springy arm.

1. Add a new component to BP\_PlayerPawn, search for and select Spring Arm. Name the component something like CameraArm.
2. Drag and drop CameraArm onto MeshComponent in the Components panel (this attaches the arm to the ship).
3. In the Details panel set the following properties for the CameraArm
  - a. Location(0, -4, 0), Rotation(0, -80, 0), Scale(0.2, 0.2, 0.2)
  - b. Target Arm Length 750
  - c. Do Collision Test False
  - d. Inherit Pitch, Yaw, Roll all False
  - e. Enable Camera Lag True
  - f. Camera Lag Speed 20
4. Next, we need to attach a camera to the end of the CameraArm. Add a new component, search for and select Camera. Name the component something like CameraComponent.
5. Drag and drop the CameraComponent onto the CameraArm in the Components panel.
6. Set the following properties for the Camera Component
  - a. Location(0, 0, 0), Rotation(0, 0, 0), Scale(1, 1, 1)
  - b. All other components should be their default values. You might want to play around with rendering effects in this panel another time.
7. Your component hierarchy should look something like the following screenshot:

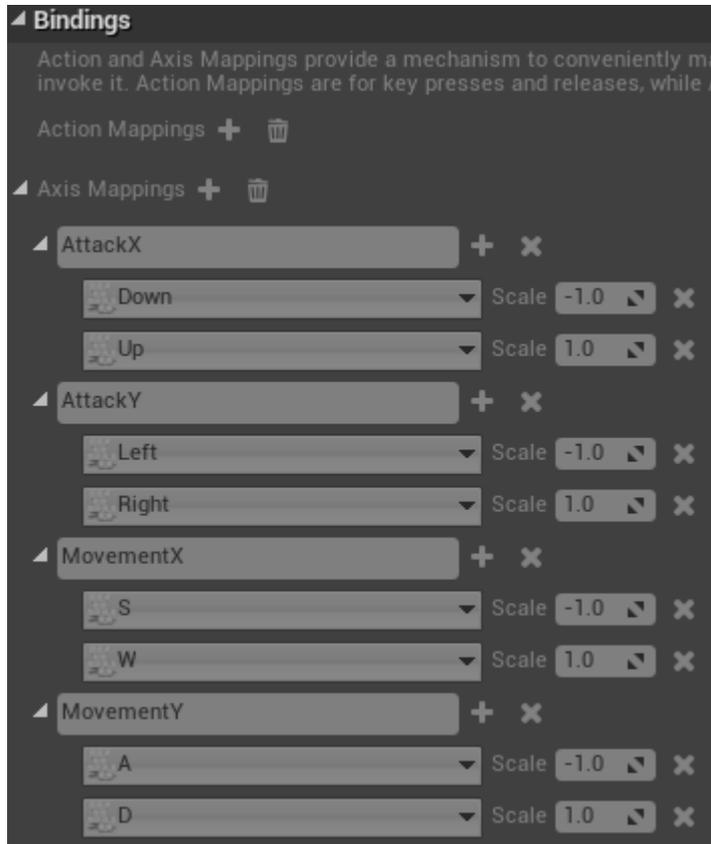


8.

## Input Settings

We need to set up some names for our player input.

1. Go to Edit>Project Settings
2. Navigate to Engine>Input and setup the following Axis Mappings



3.

## Player Pawn Blueprint Event Graph

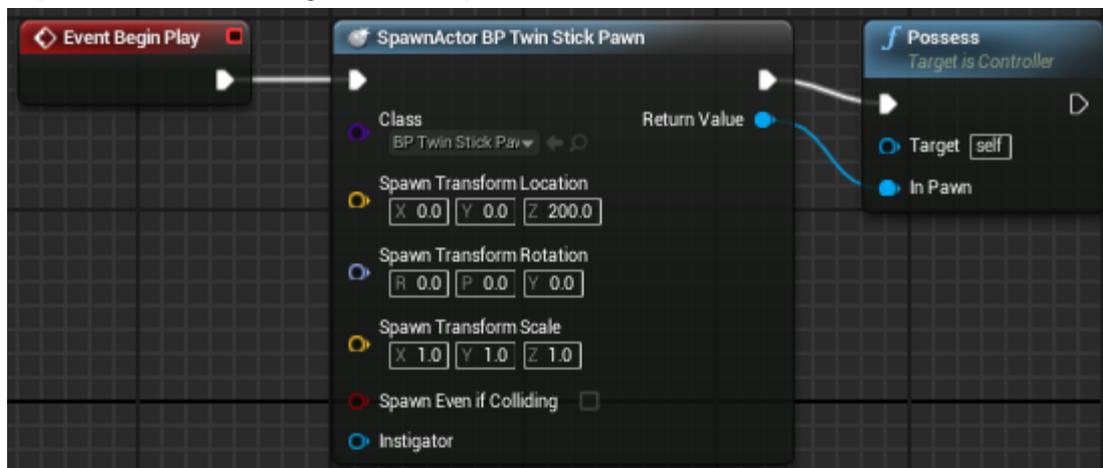
Now we have all the bits, we need to start writing the player behaviour. I haven't written up the blueprint as a step by step process so follow as long as best you can. I've provided a screenshot of the completed event graph for reference.

<http://i.imgur.com/sOrRCsP.jpg>

## PlayerController

We have provided a PlayerController blueprint but the EventGraph for this blueprint needs to be implemented.

1. Open BP\_PlayerController from the Content Browser
2. Implement the following Event Graph:



3.

4. And set the Class to the Pawn blueprint we created earlier (the name I suggested was BP\_PlayerPawn).
5. Save and close.

This ensures that our player pawn is possessed by a character controller. This is required for player input to be sent to our pawn.

## Projectiles pew pew

We have provided a blueprint for projectiles, but the blueprint behaviour is missing.

1. Open BP\_Bullet in the Content Browser
2. Follow the tutorial for implementing the event graph.
3. Use this image of the complete graph for reference: <http://i.imgur.com/qd0tqve.png>

At this stage you should be able to drive the ship around and fire projectiles!

## Updating the Game Mode

The game mode blueprint specifies the default pawn, player controller, player state and game state classes. Now that we've created a player pawn we can fill this in.

1. Open BP\_TutorialGameMode blueprint
2. Set the Default Pawn Class to BP\_PlayerPawn (or whatever you've named it)
3. Check that the default Player Controller Class is BP\_PlayerController
4. Check that the default Game State Class is BP\_TutorialGameState

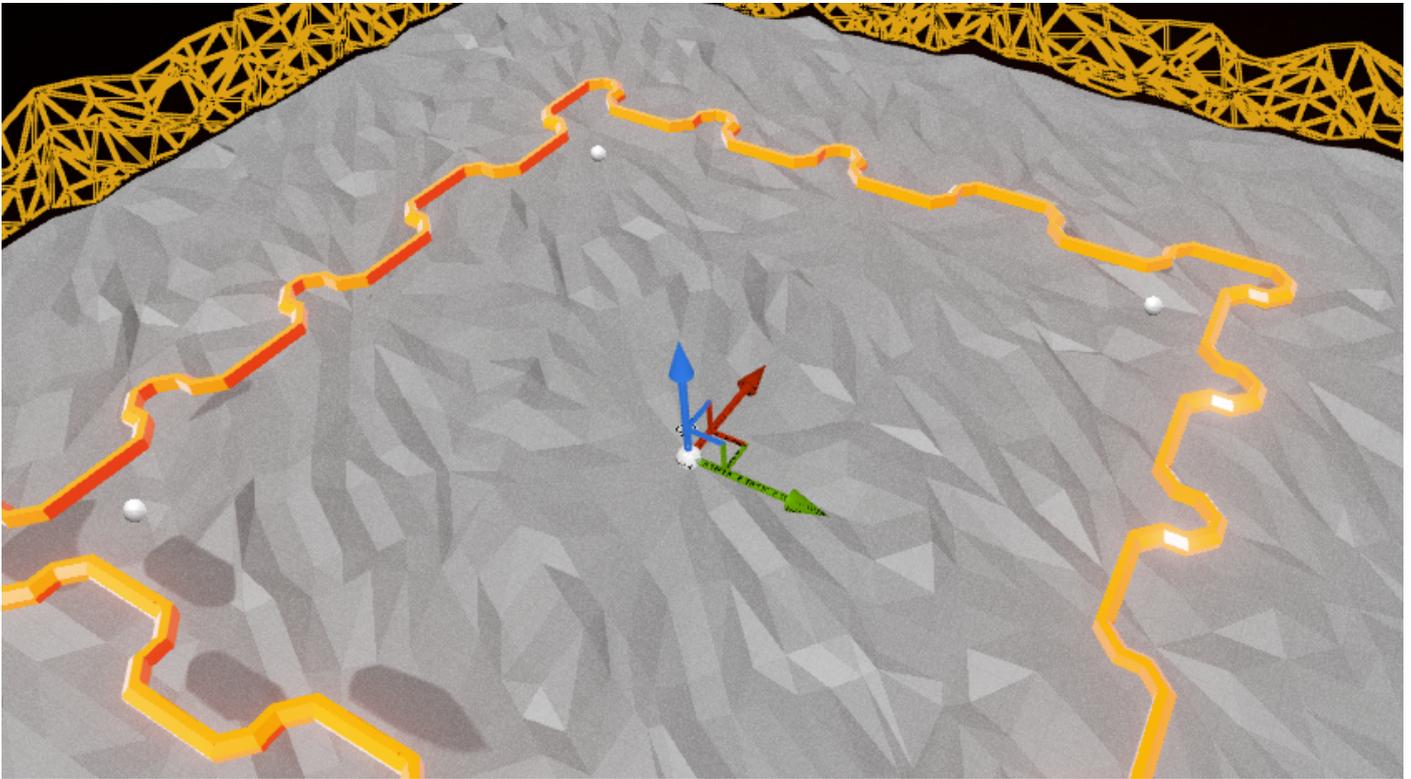
## Editing the level

The level provided isn't quite complete. To complete the arena made of *Shape\_Trim* objects use these steps:

1. Select one of the trim objects (left click) for copying.
2. CTRL+C, CTRL+V.
3. Press W to switch the widget to translate mode, then click and drag the arrows to move the object around.
4. Press E to rotate an object.

You'll also find a blueprint named BP\_WavyFloor which you can add to your level

1. Drag and drop the BP\_WavyFloor from the Content Browser onto TutorialMap 3D viewport
2. In the Details panel with your object BP\_WavyFloor selected in the World Outliner panel, set the following properties:
  - a. Location(0, 0, -150),
  - b. Rotation(0, 0, 0),
  - c. Scale(50, 50, 50)



## Building baddies

We've provided a BP\_Energy but the blueprint event graph is blank!

1. Open BP\_Energy from the Content Browser
2. Follow the tutorial for implementing the Event Graph
3. Use this image of the complete graph for reference: <http://i.imgur.com/jghnhm2.png>

## Spawning baddies

We've provided an BP\_EnergySpawner but someone forgot to finish the darn thing. Here's what the Event Graph should look like:

