Touch Drive Pro v1.0 Manual

Yusuf AKDAG

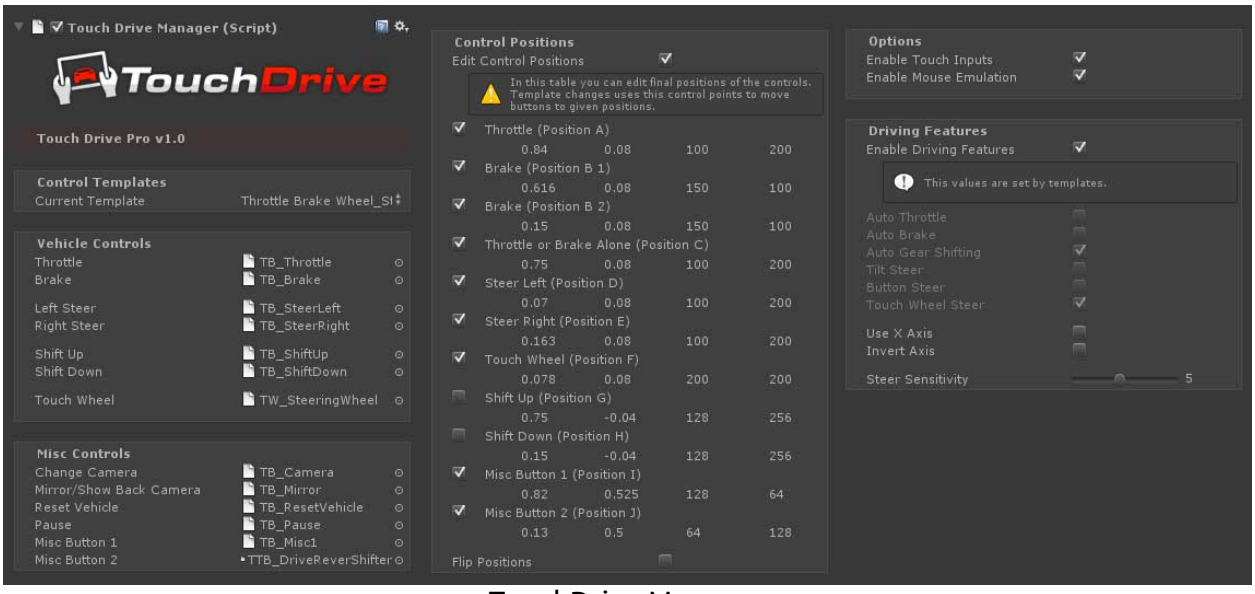
10/10/2013

TouchDrive

TouchDrive is an input component set for mobile devices to control your vehicles. This components TouchDriveManager, TouchButton, TouchToggleButton and TouchWheel

TouchDriveManager

TouchDriveManager is main component of the TouchDrive that handles and allocates inputs from devices to configured buttons. Main feature of TouchDriveManager is control templates. These templates allow you change input type instantly in game.



TouchDrive Manager

Control Templates

Control templates are created for fast switching of the controllers based user choice or game requirement. In example, in a circuit game, you may allow throttle and brake buttons but in drag race you only want to display shifters. With TouchDriveManager is easy as calling SwitchTemplate(WhichTemplateYouWantTo) method.
Template list and feature table;

Template	Active Inputs	Note
ThrottleTilt_Steer	Throttle	Good usage for tilt steer and autobrake
Throttle Button_Steer	Throttle Left Steer Right Steer	
Throttle Wheel_Steer	Throttle Touch Wheel	
Brake Tilt_Steer	Brake	Good usage for tilt steer and

		autothrottle
Brake Button_Steer	Brake Left Steer Right Steer	
Brake Wheel_Steer	Brake Touch Wheel	
Throttle Brake1Tilt_Steer	Throttle Brake	
Throttle Brake2Tilt_Steer	Throttle Brake	
Throttle BrakeButton_Steer	Throttle Brake Steer Left Steer Right	
Throttle BrakeWheel_Steer	Throttle Brake Wheel Steer	
Shifter Only	Shifter Up Shifter Down	Created for drag races
Hide All	Hides all controls	Good for pause moments.

There are two sets of controls in TouchDrive;

Vehicle Controls

Vehicle Controls are template managed controls. Their positions and visibility can be changed on demand. These controls are; Throttle, Brake, Steers, Shifters and Touch Wheel.

Misc Controls

Misc controls are other controls then vehicle controls like Pause, Reset Vehicle, Change Camera and Show Back View/Mirror View, NOS and Fire weapon

Options

Enable Touch Inputs: Enable or disable touch inputs from touch controls. This option created for pause mode.

Enable Mouse Emulation: This option enables mouse click as touch inputs.

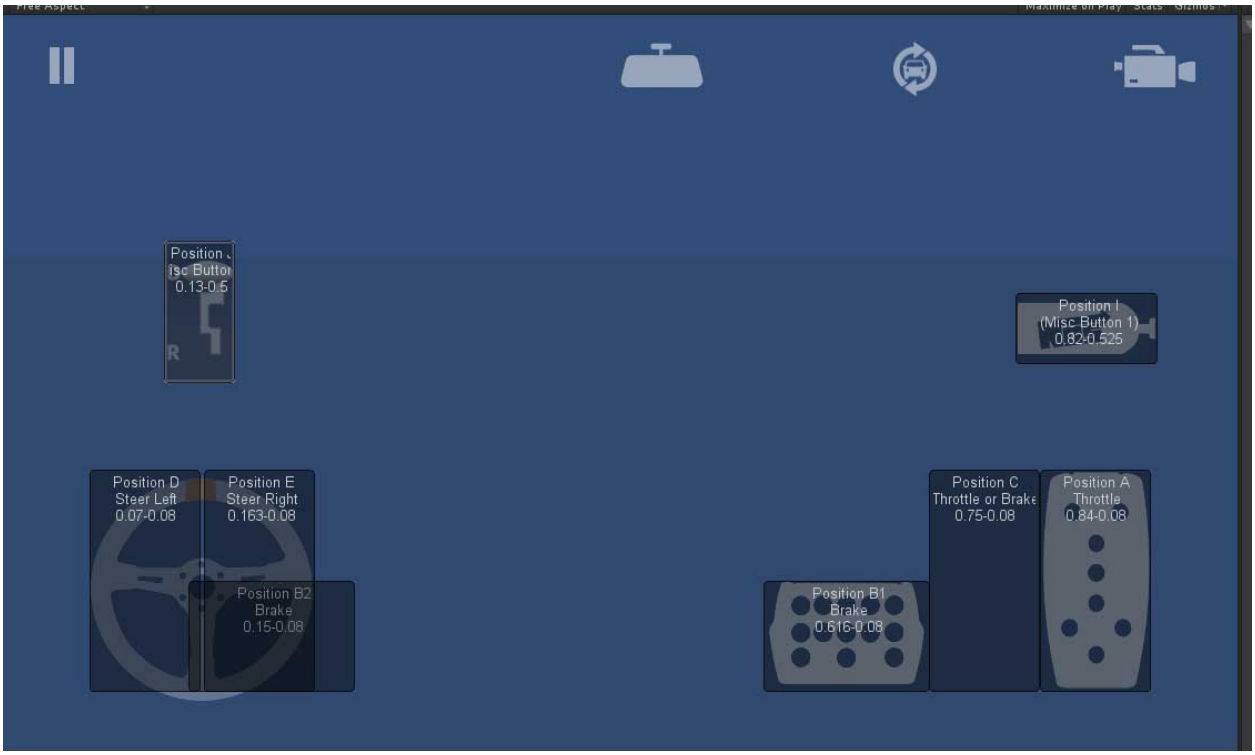
Driving Features

Driving features are created to help you to create more control on your vehicle based touch controls. It does not edit your alter your game code actually. If you want to use this option, you need to implement this feature yourself.

(Note : When you use TouchDrivePro with RGKCar C2 Mobile Controllers, this options automatically implemented, you don't need to write this codes again.)

Control Positions

Control positions created for define configuring of the control positions in race. Template switching mechanism uses this control positions for placing controls on scene. As default this control positions created for iPhone5 resolution on landscape view (1136x640).



Some of control positions in edit mode

Position Data Table

You can show/hide control position helpers individually by clicking checkboxes for each position name. This will allow you clear look the position itself.

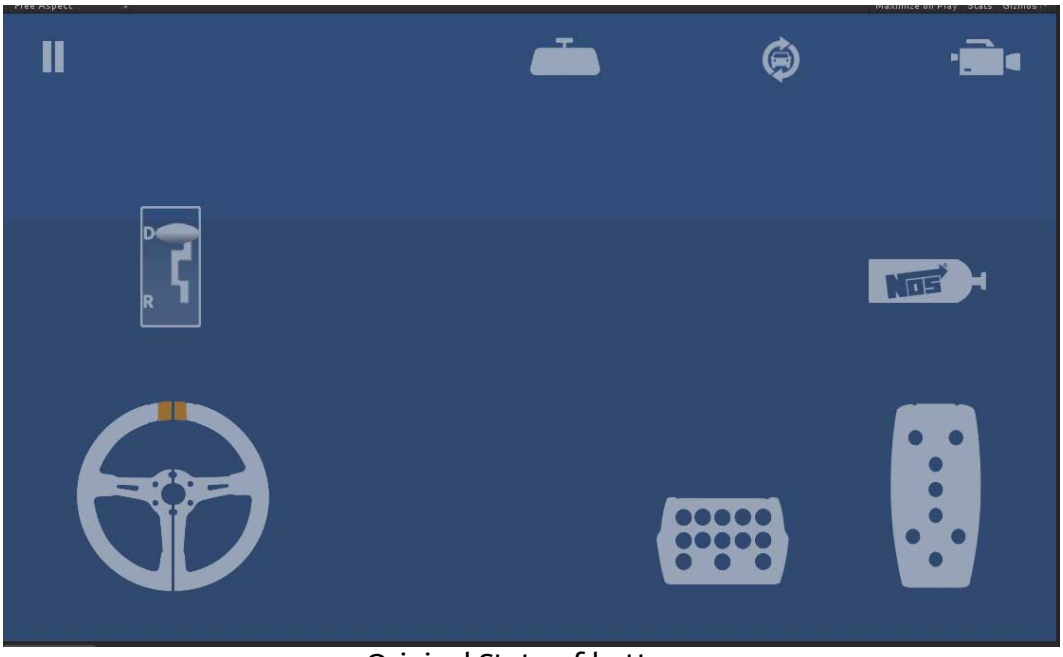
In this table, first column is X position, second column is Y position, third is button width and fourth is button height. This values used for calculation new position on flipped state.

How to Setup Position Table
Position table will final position of your input controls. For better results you need to fill these values correctly. First of all, you need to place your controls on screen non flipped state. Do not use GuiTexture's "pixel inset" properties. All positioning must be done on transform properties. TouchDriveManager uses this values to flip calculations.
After your positioning done, copy each controls position values to Control Positions table. This table fields show on image above.

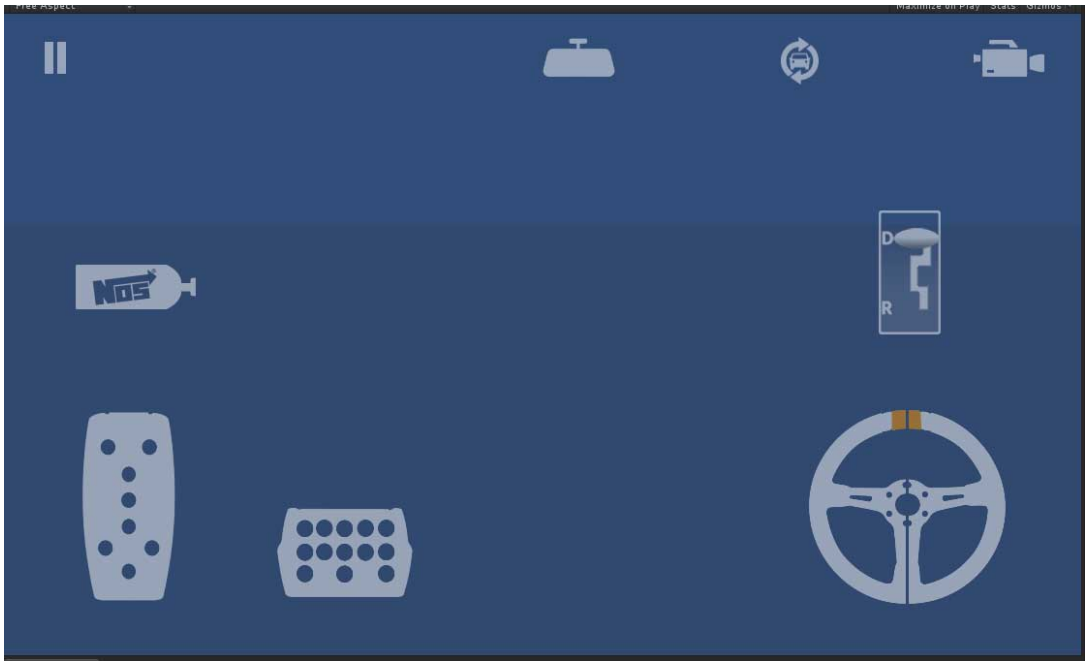
While working with position table, you can show/hide current position area with clicking checkbox of the position name.

Flip Positions

You can flip button vehicle and misc. buttons positions by this option. This option helpful for left handed player. When flip checked this calculates made by TouchDrive Manager based control positions.



Original State of buttons



Flipped state of buttons

TouchDrive Controls

TouchDrive provides 2 different buttons and a steering wheel for your game.

TouchButton

TouchButton can be used for any kind touch feedback from user. This is component requires basic GuiTexture configuration.

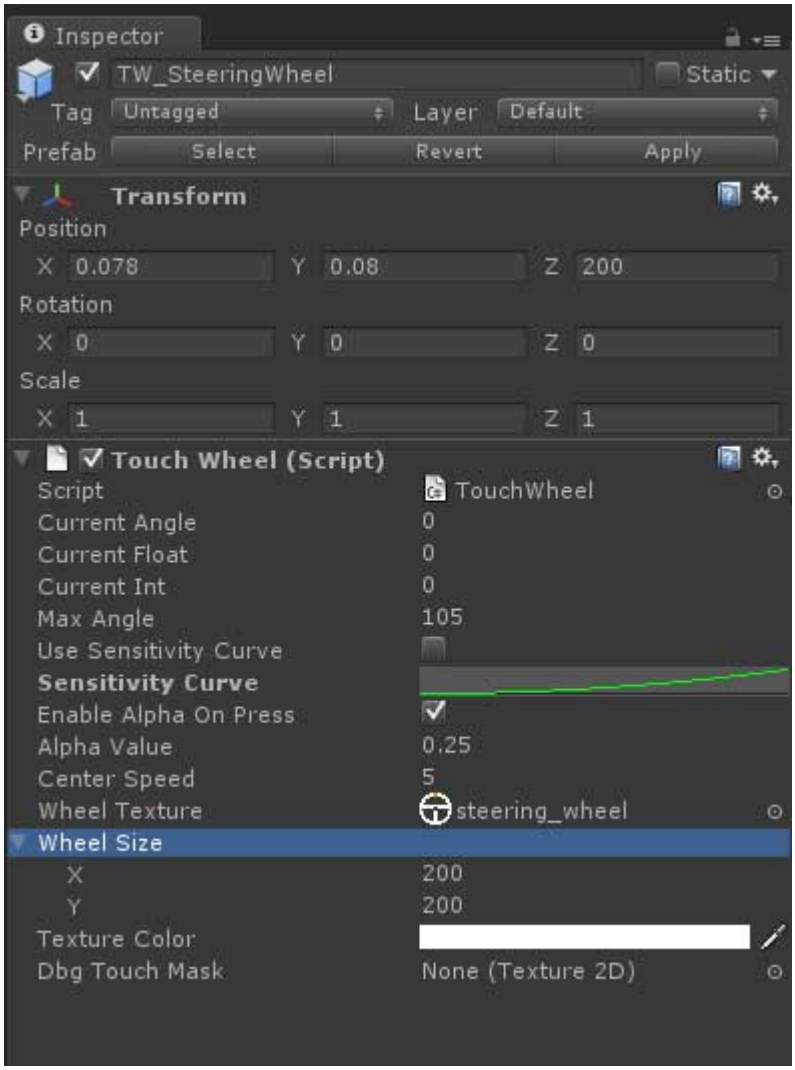
Properties;
IsPressed : returns if player touching this button.
Enable Alpha On Press: Enable/Disable alpha change on press.
Alpha Value: Alpha value when button not touched. 0 is full transparent, 1 no transparency.
One Shot: Enables IsPressed value short time then waits for remove touch from button for next fire. Its good to use when you don't control how long it pressed and requires very short press scenarios like Shifter Up, Shifter Down or change camera.
TouchData : Debug information for current touch.

TouchToggleButton

TouchToggleButton is extended version of TouchButton. Additionally it provides toggle state and changeable texture properties. Its works like on off buttons or switches.

Properties;
IsPressed : returns if player touching this button.
IsToggled : returns if toggled. Requires touch again for release toggle.
Enable Alpha On Press: Enable/Disable alpha change on press.
Alpha Value : Alpha value when button not touched. 0 is full transparent, 1 no transparency.
Released Texture: Texture for released state. (You can set GuiTexture image)
Toggled Texture: Texture for toggle state.
TouchData : Debug information for current touch.

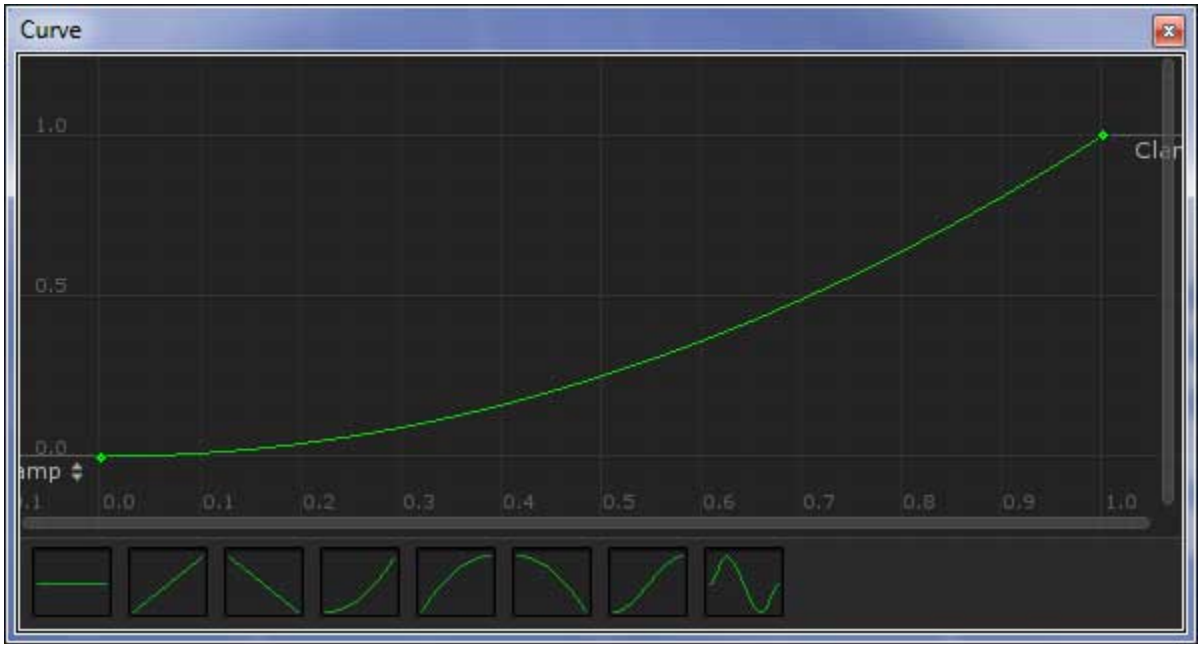
TouchWheel



TouchWheel simulates steering wheel input by user touch.

Properties;
Current Angle: Angle of wheel;
Current Float: float data of based wheel turn (float -1 to 1)
Current Int: integer data of based wheel turn (integer -1,0,1)
Max Angle: Maximum rotation angle of wheel.
Use Sensitivity Curve: This option allows you define how sensitive of the wheel. When not used float value of wheel will return as (1/MaxAngle)*CurrentAngle value. Like when Max Angle=105 and CurrentAngle=20 , CurrentFloat value will be 0.19 in this case. But when Smoothing Curve used, based your curve configuration low angles will be lower values and this will help you steer more accurately while driving.

Sensitivity Curve: A curve data for steer smoothing. As example ;



Enable Alpha On Press: Enable/Disable alpha change on press.
Alpha Value : Alpha value when button not touched. 0 is full transparent, 1 no transparency.
Center Speed: How fast wheel return to center when released.
Wheel Texture : Texture file of wheel
Wheel Size : Size of the wheel on screen as pixels.

Adding TouchDrive to Your Scene

TouchDrive comes with preconfigured and ready to use prefab for your game. To add this prefab your scene, please navigate to "_TouchDrivePro" folders under project panel and drag "_TouchDrivePro-Manager" prefab to your hierarchy panel. It's done.

Accessing the TouchInputs

Reading button and wheel data requires small scripting knowledge but it is actually easy.

Basically, instead finding and using each button reference in scene, you only need to use TouchDriveManager's button reference. Then use desired buttons field and its touch property.

You can do it that with three ways;

```
public class YourCarController : MonoBehaviour
{
    //Create a field for accesing TouchDrive.
    public TouchDriveManager TouchDriveManager;

    void Update()
    {
      //Usage 1 : Throttle button itself from TouchDriveManager's Throttle
Property
      if (TouchDriveManager.Throttle.IsPressed)
          {
             //Do Your Throttle Stuff Here
          }
      //Usage 2 : TochItems list that contains all buttons assigned.
      if (TouchDriveManager.TouchItems[0].IsPressed)
          {
             //Do Your Throttle Stuff Here
          }
      //Usage 3 : GetTouchItem(TouchItemName) method if you don't want to
deal list.
        if(TouchDriveManager.GetTouchItem(TouchDriveManager.TouchItemName.Stee
rLeft).IsPressed)
        {
             //Do Your Steer Stuff Here
        }
    }
}
```

You can find detailed vehicle controller sample at Demo folder.

Using TouchDrivePro with RGK Project

TouchDrivePro can be used with RGK C2 Mobile Controllers together. But for enabling this option you need to edit two scripts for integration. Please follow this steps ;

1- Import TouchDrivePro package to RGK Project
2- If exists, please disable or remote _RGKTouchDriveManager (TouchDrive controls bundled with RGK). Please ensure you have only one type of TouchDrive manager exits in your scene.
3- Uncomment line 24 in TouchDriveManagerBase.cs

```
//using RacingGameKit.TouchDrive;
```

4- Change iTDM interface decleration with iRGKTDM at line 27 in TouchDriveManagerBase.cs

```
//Actual line
public class TouchDriveManagerBase : MonoBehaviour, iTDM
//Required Change
public class TouchDriveManagerBase : MonoBehaviour, iRGKTDM
```

5- Change all iTouchItem interface with iRGKTouchItem in TouchDriveManager.cs lines 32, 34 and 42 (You should change 4 entries)

```
//Actual line
private List<iTouchItem> _TouchItems = new List<iTouchItem>(13);
//Required Change
private List<iRGKTouchItem> _TouchItems = new List<iRGKTouchItem>(13);
```

6- Uncomment line 15 in TouchItemBase.cs

```
//using RacingGameKit.TouchDrive;
```

7- Change iTouchItem interface with iRGKTouchItem in TouchItemBase.cs line 21

```
//Actual line
public class TouchItemBase : MonoBehaviour, iTouchItem
//Required Change
public class TouchItemBase : MonoBehaviour, iRGKTouchItem
```

8- Save the documents and let the unity recompile.
9- Please be sure “pausemenu” component on camera have active TouchDriveManager on scene.