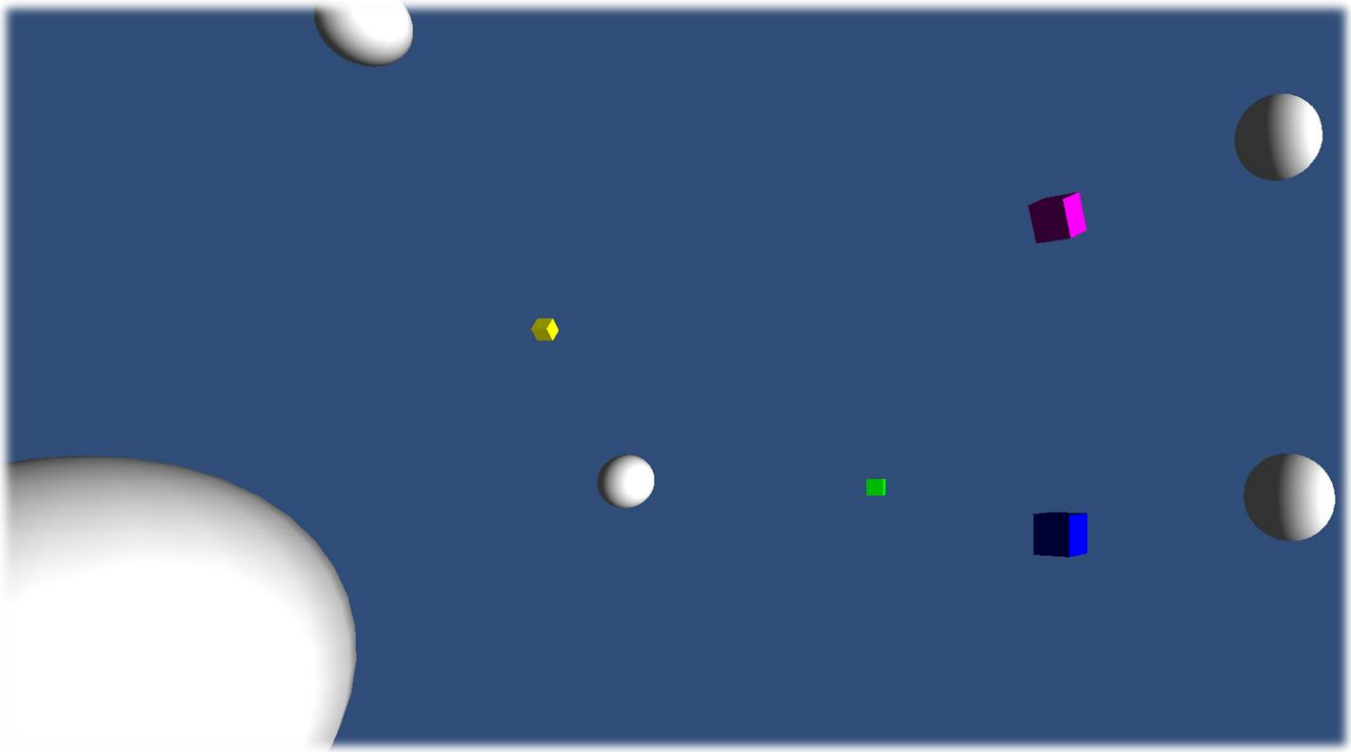
TradeSys

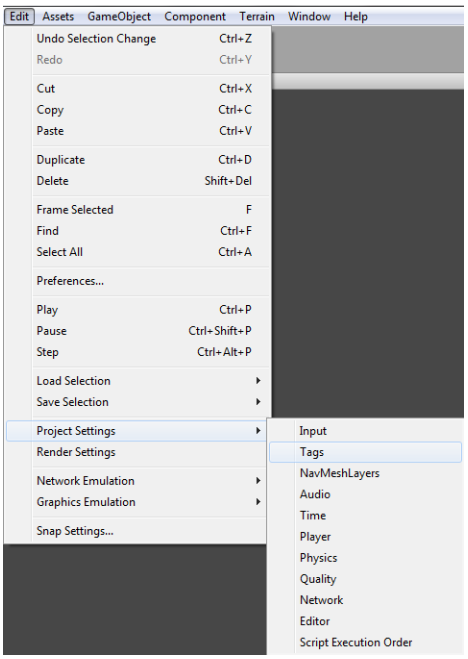


A trading and manufacturing system

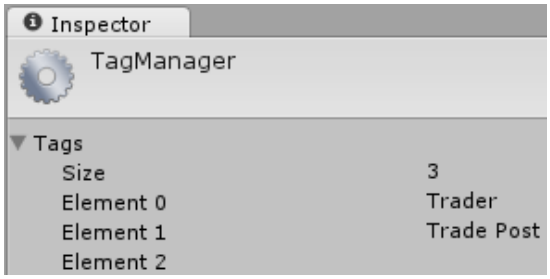
Version 1.0

Tradesys requires some setup before you can start sending out your traders, but fortunately this is very simple.

Setting up TradeSys:



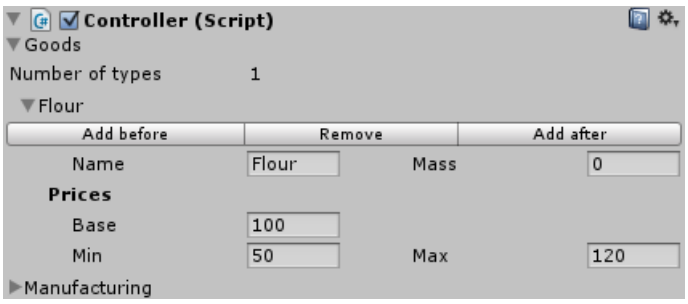
1. New tags need to be added to your game. The tags need to be called ‘Trader’ and ‘Trade Post’. Tag adding can be found by Edit > Project Settings > Tags. Here, there is a dropdown of Tags, and the new tags are added here.

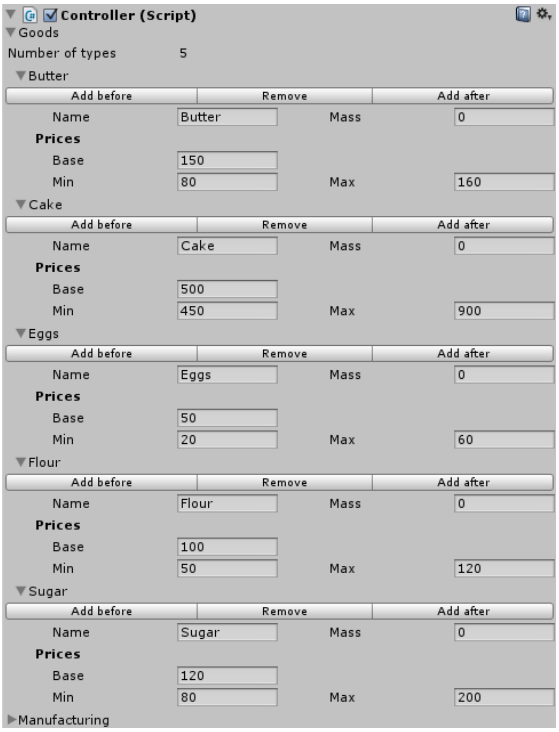


2. An empty GameObject needs to be created and called ‘Controller’. Add the controller script to this object. This will handle all of the possible trades that you set up.
3. Now it is time to setup the Trade Post. Either add a new GameObject (this time not empty, otherwise you won’t be able to see it!), or on a previously created object that you want to be your Trade Post, set the tag to Trade Post. Add the TradePost script to this object; repeat this step for all of your posts.
4. The Traders need to be sorted now. Once again, either create a new object, or use a previously added object. Change the tag to Trader and make sure that the trader is inside one of your Trade Posts (this won’t affect the game, it just looks better). Add the Trader script, and change the Target Post variable to the Trade Post where they currently are. This is so the controller knows which post to send the trader to. The stop time value is how long the trader has to stop at each post for before it is allowed to leave, it helps make it slightly more realistic,. The speed multiplier is part of demo code, which would be removed to work with your own AI. Repeat this step for each of your traders.

Adding goods to trade:

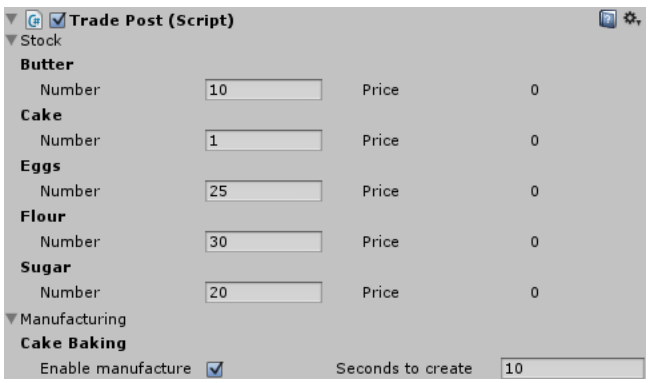
1. On the controller object, under goods, press the ‘Add’ button. This will create a new item where you can set its name. The mass is there as although it has not been scripted, you may want the trader to slow down depending on how much it is carrying, but it can be left at 0. The prices also need setting; there are three different values that require setting. The base price is the price used to work out the price at each trade post, and the min and max will limit the prices to within the range specified.

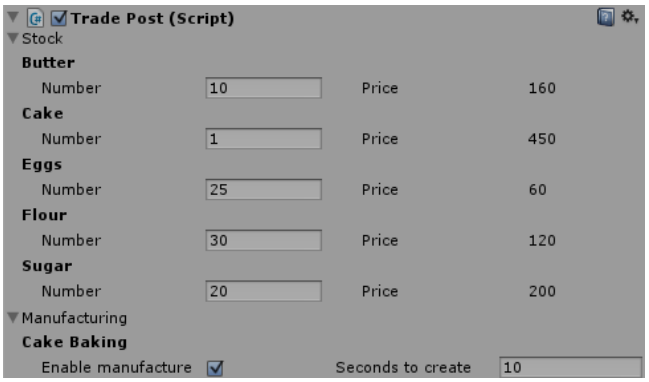


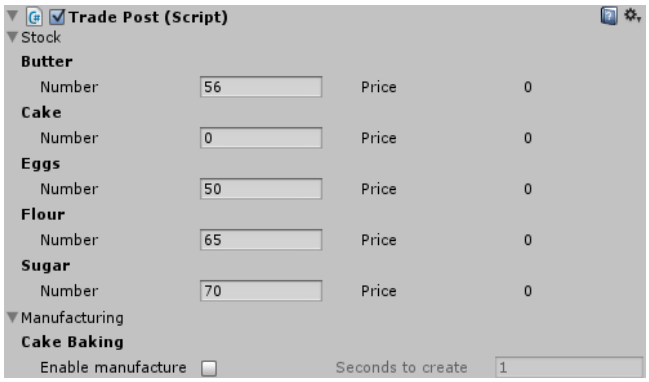


2. Each item can be removed using the ‘Remove button’, and new items can be added before or after, so it is possible to keep your items in alphabetical order.
3. Manufacturing processes are also set up in the controller, and a new process is made by pressing the ‘Add’ button. Here, you can set its name, remove it, or set the items in the needing and making sets. In both these groups, it will only need or make items that you have specified in the goods section.

4. Now that you have added the items in your game and the different manufacturing processes, it is now time to sort out the trade posts. Under the stock group, it will show all of the items that you have set in the controller and all you have to do is set how many are found at the post at the beginning of the game. The price cannot be set because the controller sorts this out for you, and will change throughout the game!
5. Under the manufacturing group, if you want the post to follow one of the processes, then check the box. This will enable you to set how long it waits before the process is repeated. It can only manufacture items when it has the required number of each item set in the controller. Steps 8 and 9 need to be repeated for each of your Trade Posts.







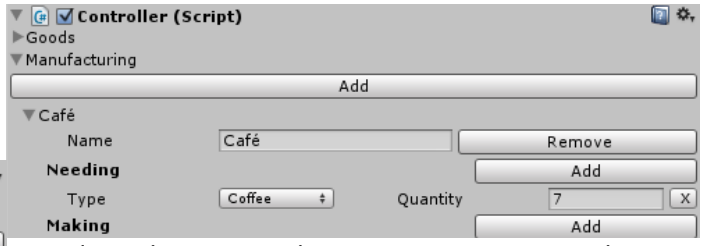
On the left, the numbers for two trade posts sorted. On the right, the prices of each item at the stations that TradeSys sorts.

You have now successfully set up TradeSys!

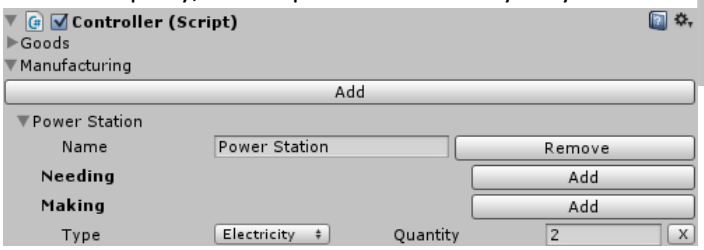
Information: In the trader, the Update() is demo code, added to show trader movement. This would be replaced with your own AI with the target post as the final target destination. When it reaches the destination, it needs to call the AtLocation() method.

Tips:

- To add a new Trade Post, just create the new object from your script, and call NewPost(times) on the new object. This will set everything up, and will generate a random number for each stock item. Times is an int[] which is required because if the number is > 0, the process will be set to true and will have the time that is specified. The length of the int[] needs to have as many elements as the number of manufacturing processes.
- To add a new Trader, create the new object through the script, call NewTrader(post), where the post is the Trade Post GameObject that the trader starts at.
- The manufacturing processes do not need to have items in needing or making; for example, a café would use coffee, but not produce anything because it is used inside. As a result, only the needing section would be used.



- Equally, a solar power station may only produce electricity and not require inputs, so only electricity on the making would be needed.



- A more industrialised trade post would have quicker manufacturing processes, so reduce the time to create. Less industrialised places would take longer, so would have a higher value.