Tech Tree Tool

API Documentation
http://www.differentmethods.com/docs/TechTree/html/files.html

Video Intro

http://www.youtube.com/watch?v=i5SxZsYW890

Video Tutorial

http://www.youtube.com/watch?v=NrK8FL-H5bc

Support

support@differentmethods.com

General Usage Notes

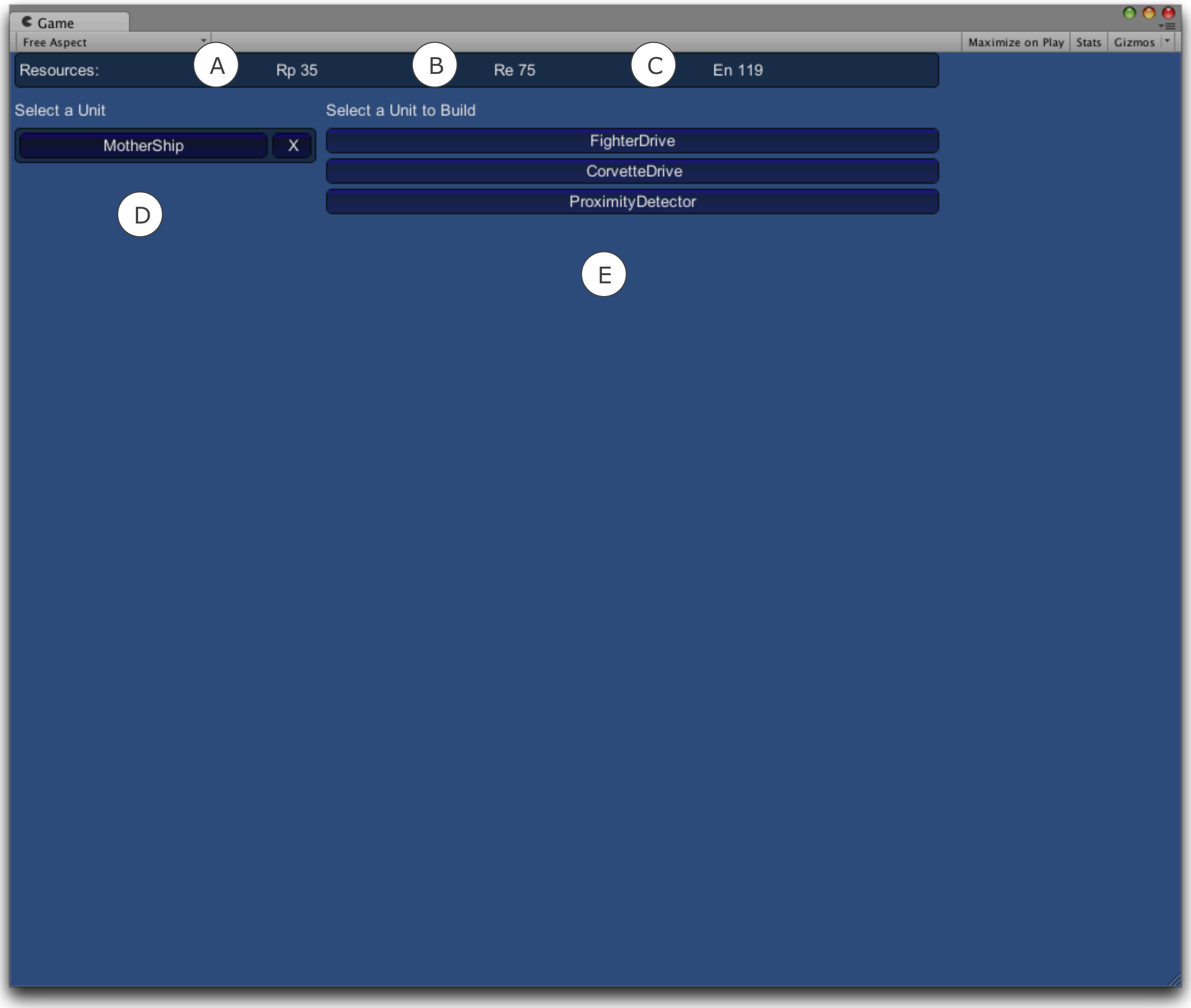
Colors are used to identify resources and groups. Make sure you set these to a distinct color when you create them.

Every Resource, Group and Blueprint is identified by a unique string ID. You must make sure the ID is unique and clear. It is used in your code to fetch blueprints from the model. It is helpful to name the ID after the item it belongs to.

Green buttons are additive operations, Eg to add a connection between blueprints. Red buttons are destructive operations, Eg to remove blueprints / groups. Yellow buttons show more information, or toggle visibility on and off.

The Test Levels

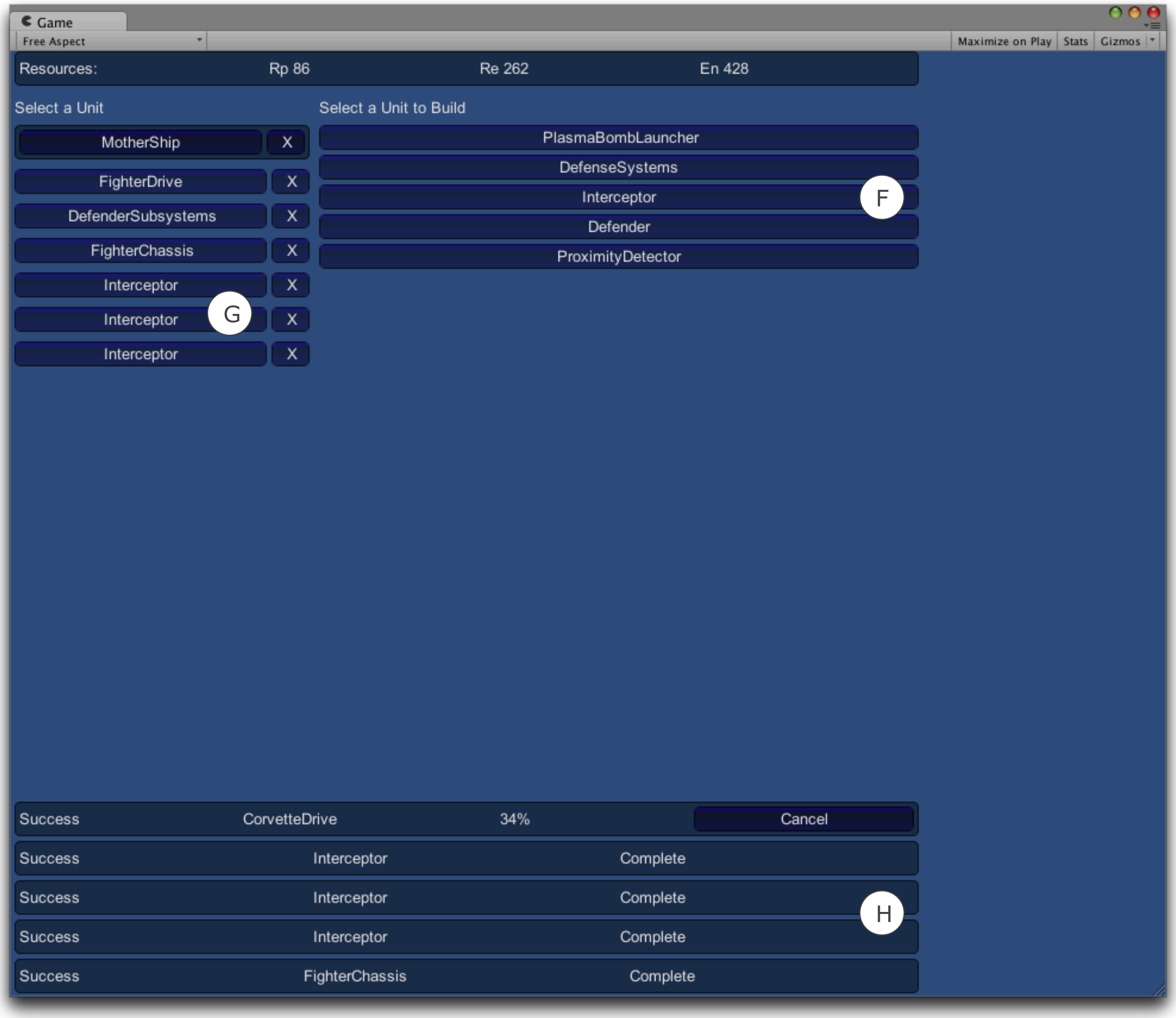
The Test level included has a generic game interface which you can use to test your tech tree models. This interface is the most basic implementation needed to use a tech tree model in a game.



(A) (B) (C) These are the resources you have which are used to build things. Each blueprint can have a different cost per resource. In the demo, these are set to auto generate.

(D) This column lists things that have been built from a blueprint . (Eg units)

(E) This column lists things (blueprints) that can be built with the currently selected unit. Only factory type units can build things.



Resources: Rp 86 Re 262 En 428

Select a Unit

MotherShip	X
FighterDrive	X
DefenderSubsystems	X
FighterChassis	X
Interceptor	X
Interceptor	X
Interceptor	X

Select a Unit to Build

PlasmaBombLauncher
DefenseSystems
Interceptor
Defender
ProximityDetector

Success	CorvetteDrive	34%	Cancel
Success	Interceptor	Complete	
Success	Interceptor	Complete	
Success	Interceptor	Complete	
Success	FighterChassis	Complete	

F	As you build more units from blueprints, more blueprints become unlocked and available to build.
G	Some units can be built multiple times. Units can also be destroyed using the X button.
H	This is the build queue. It shows percentage complete, and allows you to cancel a build. If you cancel a build, the resources are refunded.

Using the Editor

1. Choose Window -> TechTree Manager. A new window will open. I recommend that you maximise or dock this window, as the tree can become large!

2. Click the object field at the top right of the window, and choose the HomeworldTechTree from the Assets. The window should now load the initial Resources screen.

3. To create your own TechTree, right click in the project window and choose Assets->Create->TechTree Model. You will want to put the new asset in it's own folder, as during editing, many more assets will be created alongside the main asset.

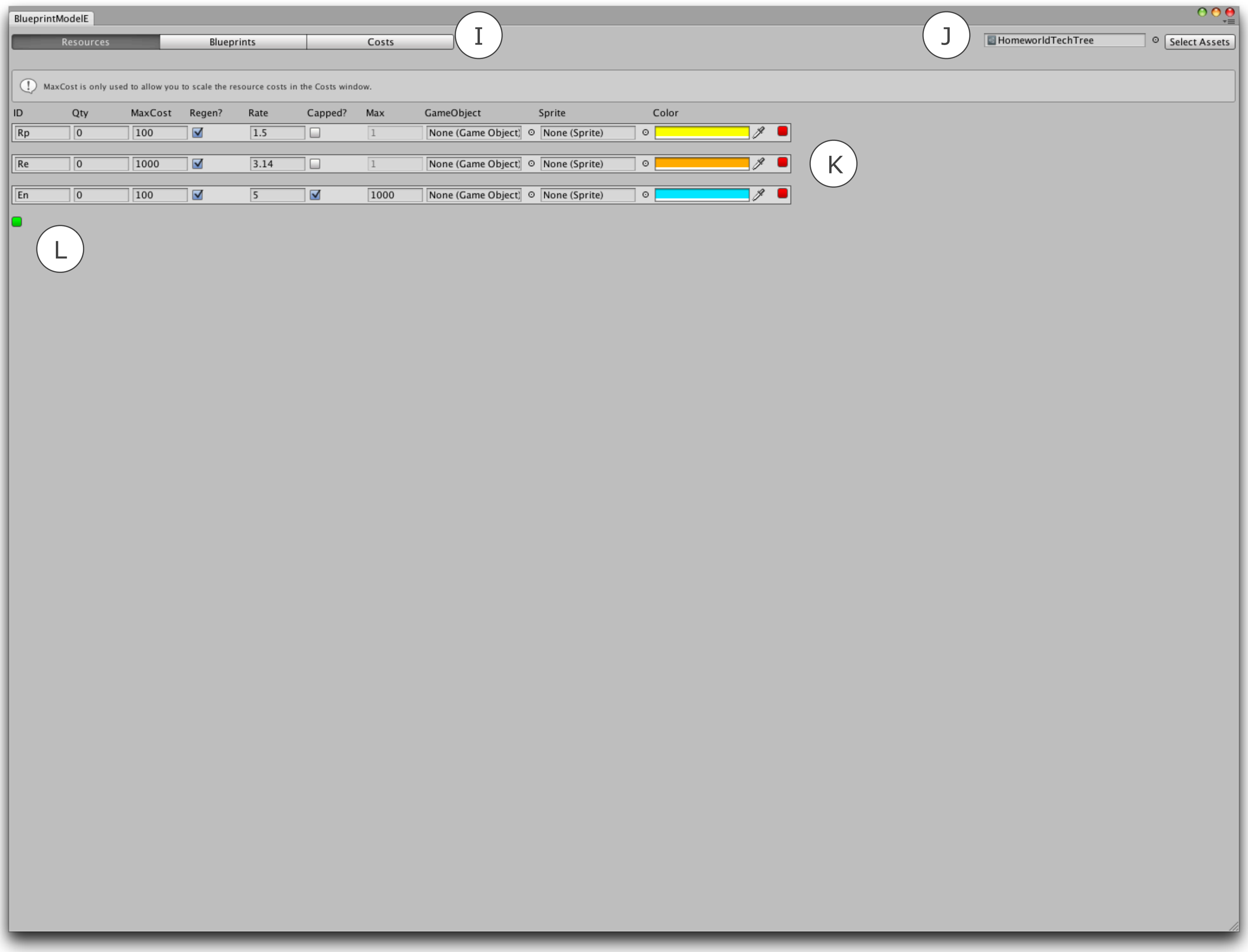
Resources

These are the variables that the player must manage in the game, and use to create units, research technology etc. They could include "Research Points", Credits, Energy, Wood, Gold, Stone Etc.

Resources have an initial quantity (qty) and a max possible blueprint cost (MaxCost). The MaxCost is the maximum value that any blueprint might require to be built. This value is only used to provide a helpful scale in the Costs editor.

The Regen flag is used to make a resource automatically increase its quantity at game time. The amount it increases is controlled by the Rate value.

If a resource is Capped, then the most quantity of that resource ever available will be the the Max value.



MaxCost is only used to allow you to scale the resource costs in the Costs window.

ID	Qty	MaxCost	Regen?	Rate	Capped?	Max	GameObject	Sprite	Color
Rp	0	100	✓	1.5	☐	1	None (Game Object)	None (Sprite)	🟨
Re	0	1000	✓	3.14	☐	1	None (Game Object)	None (Sprite)	🟧
En	0	100	✓	5	✓	1000	None (Game Object)	None (Sprite)	🟦

I These tabs switch between the different editor modes.

J This field lets you choose which asset to edit.

K In general, the red buttons remove items.

L In general, green buttons will add an item.

Blueprints

These are the things that can be built, researched or constructed in the game. They could include soldiers, factories or spaceships. Most blueprints have prerequisite blueprints, which must be built by the player during the game.

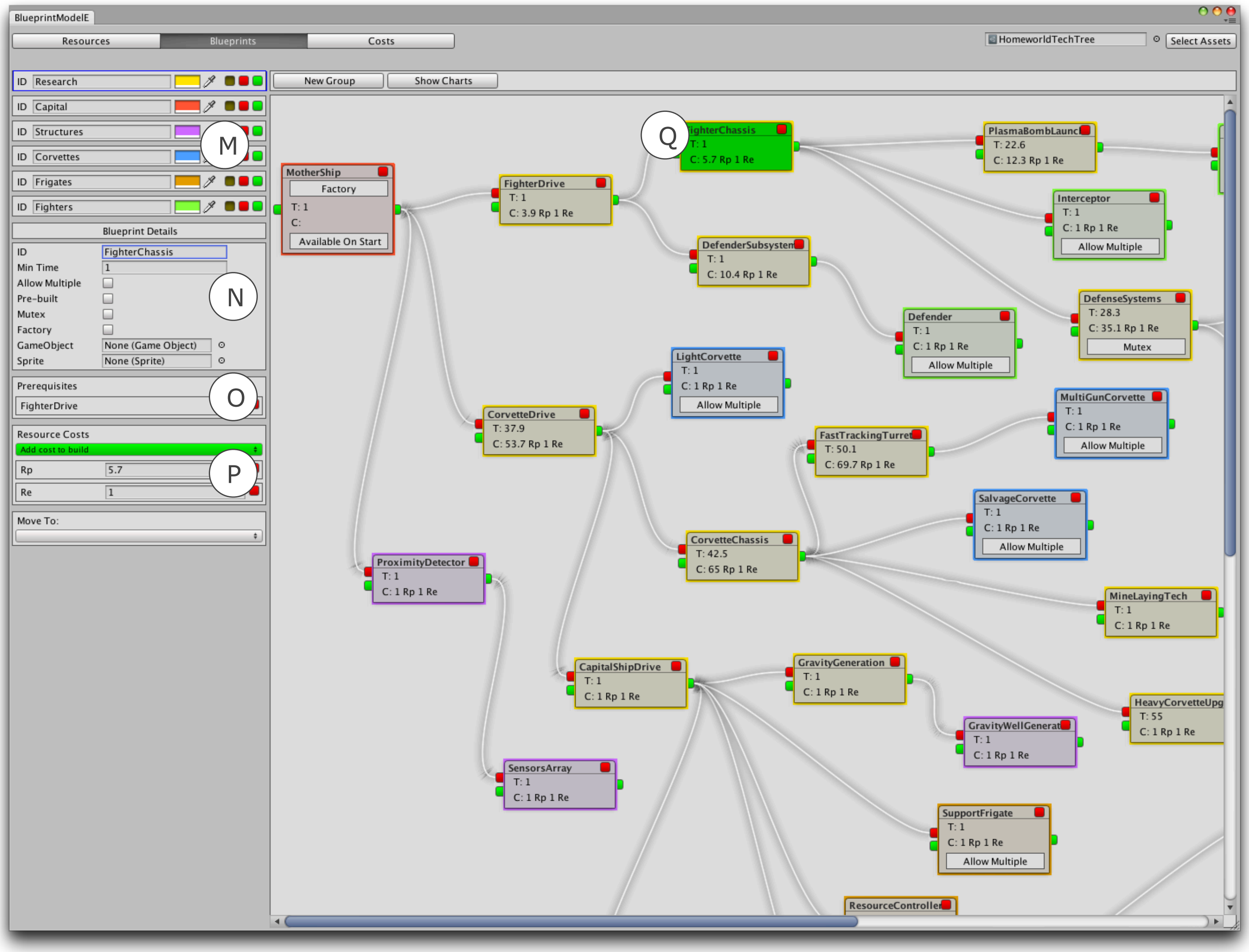
Some blueprints have a "Mutex", which means only one of the dependent blueprints can be built by the player. This can force them to choose between two branches of technology, and helps adds strategic depth to the game.

Some blueprints are a Factory. This means they are used to build other blueprints. (Only if they have been built themselves!) Most games only have 1-3 factory blueprints. For example, you can have a Factory for building air force units, and another for building land units, and perhaps one for building structures, or researching new technology.

You will always need at least one factory blueprint already built for the player at the start of the game. From this initial factory, the player can start building other blueprints during the game. You enable a factory to be in this state by checking the "Pre-built" flag for the blueprint.

Usually research type blueprints are only built once, however unit or structure type blueprints can be built multiple times. You will need to set the "Allow Multiple" flag on the blueprint to enable this.

All blueprints have a GameObject and a Sprite field. You use this to attach game assets to a blueprint. For example, you might attach a soldier model to the soldier blueprint. You can then Instantiate this model at runtime when the soldier blueprint is built.



ID Research
ID Capital
ID Structures
ID Corvettes
ID Frigates
ID Fighters

Blueprint Details

ID	FighterChassis
Min Time	1
Allow Multiple	☐
Pre-built	☐
Mutex	☐
Factory	☐
GameObject	None (Game Object)
Sprite	None (Sprite)

Prerequisites

FighterDrive

Resource Costs

Add cost to build

Rp	5.7
Re	1

Move To:

M These are the blueprint groups. It makes sense to separate a research group from a structures group, for example.

N These are the properties of the currently selected blueprint (Q).

O These are the prerequisites of the currently selected blueprint.

P These are the costs per resource required to build the currently selected blueprint.

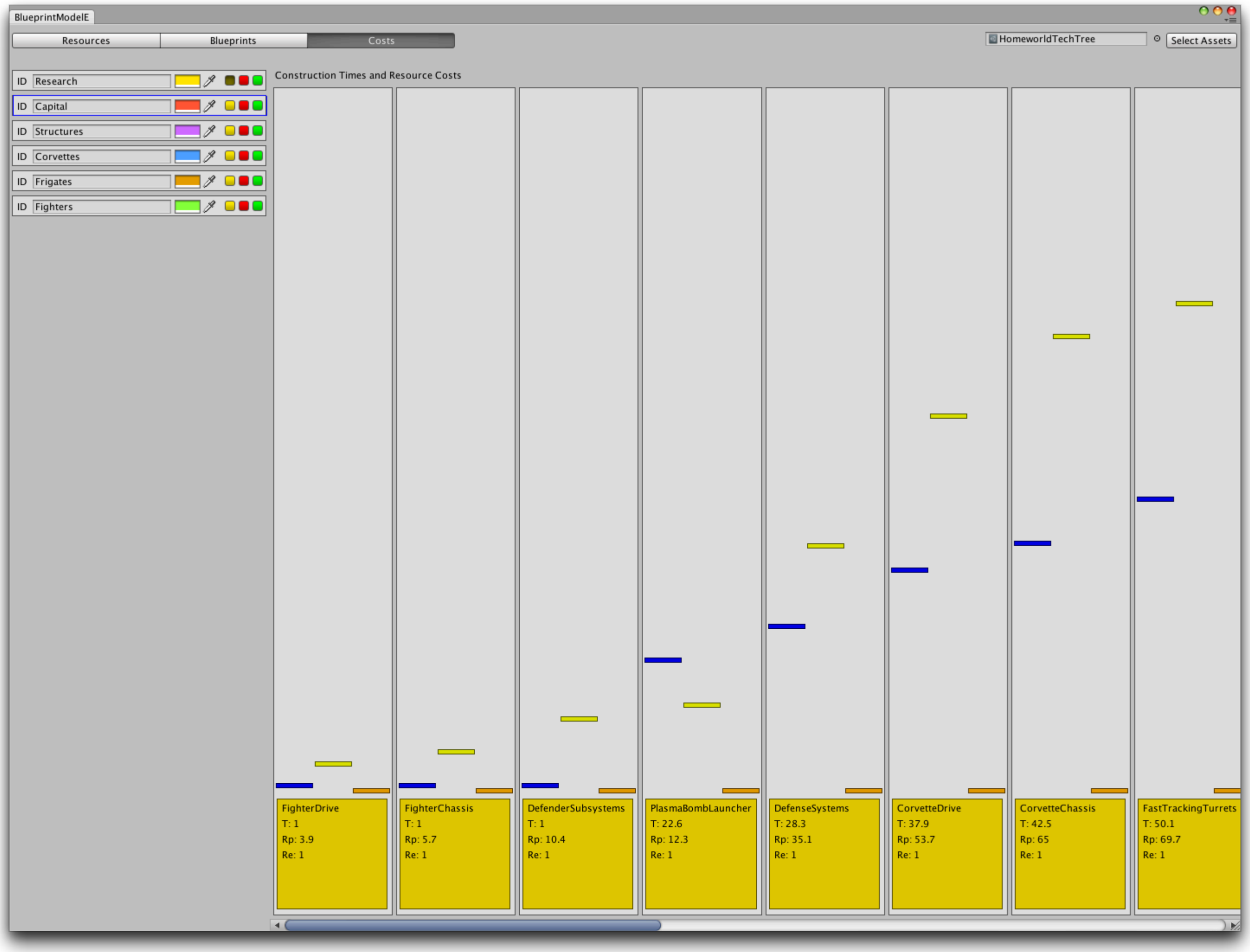
Q The currently selected blueprint is highlighted in green.

Costs

When a blueprint is built, it can consume resources. It also has a Min Time value, which is the minimum time it would take to build the blueprint, if enough resources were available.

If a resource is consumed, and there is no more of that resource available, the build process will pause until more of that resource becomes available.

Time and Resource Costs can be modified in the Blueprint inspector, or in the Costs window using slider controls.



Construction Times and Resource Costs

FighterDrive	FighterChassis	DefenderSubsystems	PlasmaBombLauncher	DefenseSystems	CorvetteDrive	CorvetteChassis	FastTrackingTurrets
T: 1	T: 1	T: 1	T: 22.6	T: 28.3	T: 37.9	T: 42.5	T: 50.1
Rp: 3.9	Rp: 5.7	Rp: 10.4	Rp: 12.3	Rp: 35.1	Rp: 53.7	Rp: 65	Rp: 69.7
Re: 1	Re: 1	Re: 1	Re: 1	Re: 1	Re: 1	Re: 1	Re: 1

API Overview

Once you have built a tech tree, you can use it in a game with the BlueprintModelController component. Assign this component to a GameObject, then assign the model you created with the Tech Tree Editor to the BMC field.

You access the Tech Tree information using this component, with one instance of the component per player.

When the BMC starts, it will check all blueprints and instantiate any that are marked to be pre-built. If the blueprint does not have a gameobject assigned, it will create an empty gameobject. It then adds a TechTreeUnit component to the new object, and if the blueprint is a factory, it will also add a TechTreeFactory component to the new object. Finally, an "OnSpawn" message is sent to the new gameobject, with a single parameter which is the factory that created the unit. NB: The factory parameter will be null for all prebuilt units.

TechTreeFactory components have a public array of blueprints, which you can iterate through and use in your game code. The .CanBeBuilt property can be used to see if the blueprint can be built into a unit. To start a build process, the component has a Build(string ID) method which will return a BlueprintBuildRequest object. You can use this object to track the status of the build. When the build is completed, the unit will be instantiated and the "OnSpawn" message will be sent to the new object.

The Blueprint BuildRequest object is used to track the state and progress of a blueprint that is being built. It has a status property, which is used to determine if the request was successful or not. It has a blueprint property, which is the blueprint which is being built. You can also use the percentComplete field to monitor progress of the build. The BuildRequest object also has a Cancel() method, which terminates the build and returns the resources.