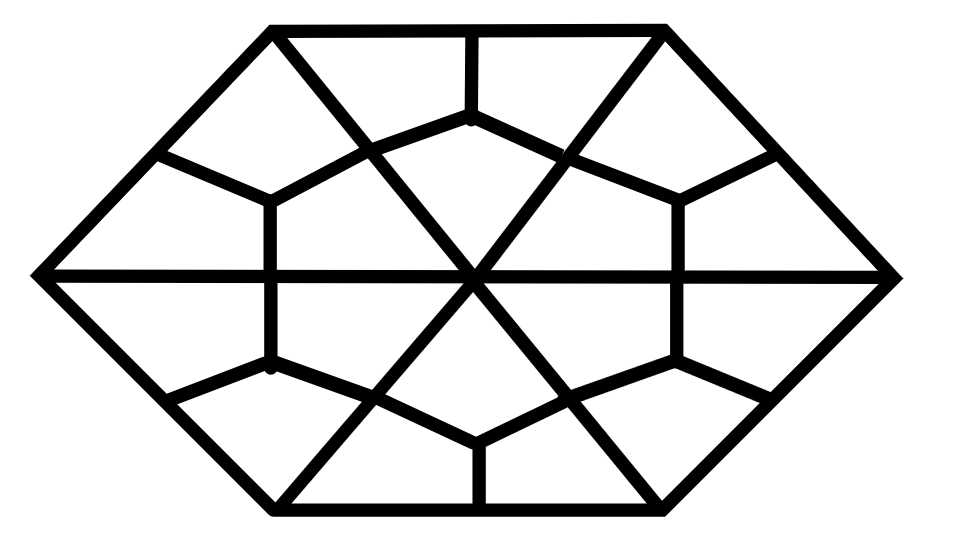
**Map Generator**

All the game is written in pure C# with no game engine (in this web version), so we got to depict somehow the “physical space”. I find that the best way to depict physical spaces in a 2D figure is a hexagon, but we just got squares (an index in some array memory), so I had to depict a series of 2D figures to go from a square to end in a hexagon, but it needs a somewhat big scale to achieve it, so that I got to draw a trapezoid into a bidimensional array of 16x16 slots so to speak, there we got the canvas to draw the trapezoid. Their legs are easy, columns and rows are just enough, the problem was to draw the hypotenuse, so I use pixel-art to draw that diagonal, and simulate it to select slots, or index, to load the world objects, within the drawing obviously, and what was left outside those slots just become unusable, like blocked.

Then with the trapezoids, with two different figures but with two possible positions for one of two trapezoid figures, and four different possible position for the remaining. The six final trapezoid figure combinations are useful to shape an equilateral triangle, so let’s note that, the six trapezoids serve to form an equilateral triangle, and what then? Easy, there you got the two positioned equilateral triangle to form the hexagon! Don't believe me? Check out the picture below. 

**Region: a hexagonal (and conformed by)**

**Straight Zones:**

* Pointed Triangule Area
* Western Trapezoid Area
* Eastern Trapezoid Area

**Inverted Zones:**

* Pointed Triangule Area
* Western Trapezoid Area
* Eastern Trapezoid Area

Parcel: see below to be aware the shape of the Area figures

Over the Area matrix, each "\*" is an empty item, so it does not exist in the physic game (it is only necessary to use the squared matrix shape), and the Parcels are depicted as the "0" items, each "0" are the matrix index where a Parcel object is hosted.

Now this is how the Area matrix is depicted:

**Area Matrix**

**Straight Pointed Triangule**

[\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*]

[\*][\*][\*][\*][\*][\*][\*][0][0][\*][\*][\*][\*][\*][\*][\*]

[\*][\*][\*][\*][\*][\*][0][0][0][0][\*][\*][\*][\*][\*][\*]

[\*][\*][\*][\*][\*][\*][0][0][0][0][\*][\*][\*][\*][\*][\*]

[\*][\*][\*][\*][\*][0][0][0][0][0][0][\*][\*][\*][\*][\*]

[\*][\*][\*][\*][\*][0][0][0][0][0][0][\*][\*][\*][\*][\*]

[\*][\*][\*][\*][0][0][0][0][0][0][0][0][\*][\*][\*][\*]

[\*][\*][\*][\*][0][0][0][0][0][0][0][0][\*][\*][\*][\*]

[\*][\*][\*][0][0][0][0][0][0][0][0][0][0][\*][\*][\*]

[\*][\*][\*][0][0][0][0][0][0][0][0][0][0][\*][\*][\*]

[\*][\*][0][0][0][0][0][0][0][0][0][0][0][0][\*][\*]

[\*][0][0][0][0][0][0][0][0][0][0][0][0][0][0][\*]

[0][0][0][0][0][0][0][0][0][0][0][0][0][0][0][0]

[\*][0][0][0][0][0][0][0][0][0][0][0][0][0][0][\*]

[\*][\*][\*][\*][0][0][0][0][0][0][0][0][\*][\*][\*][\*]

[\*][\*][\*][\*][\*][\*][\*][0][0][\*][\*][\*][\*][\*][\*][\*]

**Western Straight Trapezoid**

[\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*]

[\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*]

[\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*]

[\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*]

[\*][\*][\*][\*][\*][\*][\*][0][0][\*][\*][\*][\*][\*][\*][\*]

[\*][\*][\*][\*][\*][\*][0][0][0][0][0][0][\*][\*][\*][\*]

[\*][\*][\*][\*][\*][0][0][0][0][0][0][0][0][0][0][\*]

[\*][\*][\*][\*][\*][0][0][0][0][0][0][0][0][0][0][0]

[\*][\*][\*][\*][0][0][0][0][0][0][0][0][0][0][0][0]

[\*][\*][\*][\*][0][0][0][0][0][0][0][0][0][0][0][0]

[\*][\*][\*][0][0][0][0][0][0][0][0][0][0][0][0][0]

[\*][\*][\*][0][0][0][0][0][0][0][0][0][0][0][0][0]

[\*][\*][0][0][0][0][0][0][0][0][0][0][0][0][0][0]

[\*][\*][0][0][0][0][0][0][0][0][0][0][0][0][0][0]

[\*][0][0][0][0][0][0][0][0][0][0][0][0][0][0][0]

[0][0][0][0][0][0][0][0][0][0][0][0][0][0][0][0]

**Eastern Straight Trapezoid**

[\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*]

[\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*]

[\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*]

[\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*]

[\*][\*][\*][\*][\*][\*][\*][0][0][\*][\*][\*][\*][\*][\*][\*]

[\*][\*][\*][\*][0][0][0][0][0][0][\*][\*][\*][\*][\*][\*]

[\*][0][0][0][0][0][0][0][0][0][0][\*][\*][\*][\*][\*]

[0][0][0][0][0][0][0][0][0][0][0][\*][\*][\*][\*][\*]

[0][0][0][0][0][0][0][0][0][0][0][0][\*][\*][\*][\*]

[0][0][0][0][0][0][0][0][0][0][0][0][\*][\*][\*][\*]

[0][0][0][0][0][0][0][0][0][0][0][0][0][\*][\*][\*]

[0][0][0][0][0][0][0][0][0][0][0][0][0][\*][\*][\*]

[0][0][0][0][0][0][0][0][0][0][0][0][0][0][\*][\*]

[0][0][0][0][0][0][0][0][0][0][0][0][0][0][\*][\*]

[0][0][0][0][0][0][0][0][0][0][0][0][0][0][0][\*]

[0][0][0][0][0][0][0][0][0][0][0][0][0][0][0][0]

**Inverted Pointed Triangle**

[\*][\*][\*][\*][\*][\*][\*][0][0][\*][\*][\*][\*][\*][\*][\*]

[\*][\*][\*][\*][0][0][0][0][0][0][0][0][\*][\*][\*][\*]

[\*][0][0][0][0][0][0][0][0][0][0][0][0][0][0][\*]

[0][0][0][0][0][0][0][0][0][0][0][0][0][0][0][0]

[\*][0][0][0][0][0][0][0][0][0][0][0][0][0][0][\*]

[\*][\*][0][0][0][0][0][0][0][0][0][0][0][0][\*][\*]

[\*][\*][\*][0][0][0][0][0][0][0][0][0][0][\*][\*][\*]

[\*][\*][\*][0][0][0][0][0][0][0][0][0][0][\*][\*][\*]

[\*][\*][\*][\*][0][0][0][0][0][0][0][0][\*][\*][\*][\*]

[\*][\*][\*][\*][0][0][0][0][0][0][0][0][\*][\*][\*][\*]

[\*][\*][\*][\*][\*][0][0][0][0][0][0][\*][\*][\*][\*][\*]

[\*][\*][\*][\*][\*][0][0][0][0][0][0][\*][\*][\*][\*][\*]

[\*][\*][\*][\*][\*][\*][0][0][0][0][\*][\*][\*][\*][\*][\*]

[\*][\*][\*][\*][\*][\*][0][0][0][0][\*][\*][\*][\*][\*][\*]

[\*][\*][\*][\*][\*][\*][\*][0][0][\*][\*][\*][\*][\*][\*][\*]

[\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*]

**Western Inverted Trapezoid**

[0][0][0][0][0][0][0][0][0][0][0][0][0][0][0][0]

[\*][0][0][0][0][0][0][0][0][0][0][0][0][0][0][0]

[\*][\*][0][0][0][0][0][0][0][0][0][0][0][0][0][0]

[\*][\*][0][0][0][0][0][0][0][0][0][0][0][0][0][0]

[\*][\*][\*][0][0][0][0][0][0][0][0][0][0][0][0][0]

[\*][\*][\*][0][0][0][0][0][0][0][0][0][0][0][0][0]

[\*][\*][\*][\*][0][0][0][0][0][0][0][0][0][0][0][0]

[\*][\*][\*][\*][0][0][0][0][0][0][0][0][0][0][0][0]

[\*][\*][\*][\*][\*][0][0][0][0][0][0][0][0][0][0][0]

[\*][\*][\*][\*][\*][0][0][0][0][0][0][0][0][0][0][\*]

[\*][\*][\*][\*][\*][\*][0][0][0][0][0][0][\*][\*][\*][\*]

[\*][\*][\*][\*][\*][\*][\*][0][0][\*][\*][\*][\*][\*][\*][\*]

[\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*]

[\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*]

[\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*]

[\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*]

**Eastern Inverted Trapezoid**

[0][0][0][0][0][0][0][0][0][0][0][0][0][0][0][0]

[0][0][0][0][0][0][0][0][0][0][0][0][0][0][0][\*]

[0][0][0][0][0][0][0][0][0][0][0][0][0][0][\*][\*]

[0][0][0][0][0][0][0][0][0][0][0][0][0][0][\*][\*]

[0][0][0][0][0][0][0][0][0][0][0][0][0][\*][\*][\*]

[0][0][0][0][0][0][0][0][0][0][0][0][0][\*][\*][\*]

[0][0][0][0][0][0][0][0][0][0][0][0][\*][\*][\*][\*]

[0][0][0][0][0][0][0][0][0][0][0][0][\*][\*][\*][\*]

[0][0][0][0][0][0][0][0][0][0][0][\*][\*][\*][\*][\*]

[\*][0][0][0][0][0][0][0][0][0][0][\*][\*][\*][\*][\*]

[\*][\*][\*][\*][0][0][0][0][0][0][\*][\*][\*][\*][\*][\*]

[\*][\*][\*][\*][\*][\*][\*][0][0][\*][\*][\*][\*][\*][\*][\*]

[\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*]

[\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*]

[\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*]

[\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*][\*]

**Environment Features**

There're several environment types, forming the Biomes

**Land Types Distribution**

**Production Chain**

***Now everything's set up, how does it start?***

Once map generated and population settle in one Parcel position, the “citizens” will set up their Necessity prioritization set. There'll be necessities that may not be possible or feasible, so at minimum there’re three that must be Satisfy at expense of Health points if they aren’t so, you must procure Satisfy them if you want at least a little of stabilization. These three are of course minimal Feeding (Water and Food), Cover and Shelter, and are the most basic stage of satisfaction, because a Necessity has several levels of satisfaction according to their quality in consumption, especially in quality not quantity. Once the necessities set, the citizen proceeds to do the ‘cheapest’ way to satisfies them –if you don’t set anything particular or specific action– this is the way a Citizen will behave, without meaning the save in efforts will be always the maximum if it’s not first priority, because he will expend more ‘energy’ or Capacity in that necessity if it’s feasible and affordable to do so, whether he can he’ll do it.

**INecessity interface**

A Necessity is a special and particular goal shared by citizens, everyone will strive for them, and they’re set automatically and unintentionally, and always that a citizen could Satisfy them without engage its future consumption levels he will proceed to do it; this last take it with a grain of salt, because that ‘responsible’ behavior depends on its consumption profile, or its Education and chiefly on its Frivolity levels. The key to understand necessities is their mathematical fundamentals.

Whatever the Necessity was, a Citizen begins its Turn, or time, with 0% (of that Grade) of Necessity satisfied, if this Necessity is selected to be worked then the Citizen Capacity is spent until fill the percentage wished, it’s a percentage goal (the Expectation) that is set alongside its Priority; the Capacity could set a high level of Satiety, if the Capacity and Priority do so, based on the so-called Expectation number. The necessities grades are set up and then the Capacity is consumed until that Necessity is satisfied, or at least that will try, if the intent failed will reevaluate whether it’s feasible to keep trying to Satisfy that Necessity (because it’s super important), or whether it will be left; in any case if at the end of the Turn the Expectation goal isn’t reach, the gap between Expectation and current Capacity spent effectively will be the so-called Frustration points (one of the subtle social factors). This is how the game sets the Polis’s purchasing power.

Beyond all above here’s a simplify table to look all the necessities and their levels:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Level / Necessity | 1° Level or Grade | 2° Level or Grade | 3° Level or Grade | 4° Level or Grade | 5° Level or Grade | 6° Level or Grade | 7° Level or Grade |
| Feeding | Water | Food | Nutrition | Plenty | Variety\* | Luxury | Plastic |
| Garment | Cover | Clothes | Outfit | Plenty | Variety\* | Luxury | Plastic |
| Dwelling | Shelter | Homely | Extension | Plenty | Luxury | Monumental | Plastic |
| Security | Defense | Politics | Stability | Security | Reliance | Insurance | Martial |
| Knowledge | Faith | Reasoning | Science | Complexity | Virtue | Wisdom | Peace |
| Energy | Heat | Kinetic | Powerful | Plenty | Steam | Electricity | Fuel |
| Social | Clanship | Recreation | Moral | Instruction | Branch\*\* | Formal\*\*\* | Peace |
| Health | First Aid | Medical° | Particular° | Surgery | Hygiene | Prevent° | Plastic |

\*: for spacing reasons, we choose Variety instead the real hard code world: Assortment

\*\*: in this case the word is: Syncretism

\*\*\*: and here is: Institutionality

°: for spacing reasons, just add “Care” next to the one shown in the table, to know about the hard code

**Priority and Grade properties**

Should every Necessity be satisfied? we already know they don’t. They are sorted by Priority, a float number that goes around the 1, so if it’s above 1 then it must be satisfied, if below it’s not need to, but it will be whether the prioritized ones are satisfied and the Capacity left still being above 30%. This is a dynamic number as you probably deduced, but they come from a static one set it according to its nature.

In the Grade case, normally one Citizen not go to scale to the next level if the entire set of Necessities Grade level are already satisfied, if they don’t then he will focus in the remaining ones. But there are exceptions, the first and most evident is the Feeding, you saw above that the first two “grades” are like the first one of the rests of “necessities”, these two are treated like one single Grade. Then Social is also a particular case because the first three “grades” are able to be fulfill if the previous grades aren’t satisfied in the rests, after these three levels the cost to Satisfy them becomes more expensive.

But for the rest basically their dynamics are regulated by the starting Priority value, Knowledge for example is a Necessity placed in the last of the “priorities”, so, it’s tough to be developed so to speak, and also the rests of “necessities” can be developed one extra Grade above this. Health is also another costly Necessity, but this keeps the same Grade level up rule that everyone; so, in the same way, Security it’s almost the opposite case, it goes faster than normal but awaits the rest to follows the next level. The remaining, Feeding, Garment, Dwelling and Energy go in a very similar velocity, but Dwelling and Energy are typically more expensive.

**Capacity and Cost properties**

But how’s all this of expensive or “cheaper” Necessity? Well, everyone has a Cost property that decrease the Capacity effect on the Necessity satisfaction, is a float number always below or equal to 1 and above 0 (much more in fact), that will be used to be multiplied by the Capacity percentage application that will be used.

But wait, where that number came from and what’s the real effect of that Capacity percentage used? Is it something that just is applied and everybody happy? If you’re asking that question we are on the right track. First the Cost is the result of the several condition so it’s a variable not a constant, what it’s constant is the starting value, and the Cost it’s used also for calculating its feasibility, giving Citizens to foresee not to do risky tasks.

Regarding the Capacity ability to fulfill “necessities”, of course it is not lineal. The Capacity spent will go to the “spaces” where it must to go –the closest one– to reach the resource, or the Product correspondent to the Activity Inventory needed, do the task and get what he can, then he consumes those achieved “products” (of course most of the cases, especially when the Market is settled, he'll do whatever he can do to obtain more value from his capabilities). As I told above, the gap between resulting Satiety from the Capacity application, to the initial Expectation (remember, the goal set up before apply the Capacity, or action to reach that goal), is an increase level to the Frustration.

Here's when it becomes a little tricky. If that Necessity isn’t satisfied to the minimum required percentage, and no one with a similar Priority level is lacking to be Satisfy, and also the Capacity is above 30% –unless the Necessity is Urgent (this is the above, the first levels of Feeding, Garment and Dwelling)–, then the remaining Capacity will be used again, in the same Turn.

**Fatigue State**

We just say that the descripted behavior above is when the Capacity is up to 30%, because when the Citizen lacked Satisfy at minimum basic necessities level, or even when the Player force the population to do it, the Citizens that use their 30% of Capacity left, they enter in the so-called Fatigue State, where every point used penalize the starting points of the next Turn, and for each extra point used an accumulator that multiply the effect is set up, while the total points used in the Fatigue State will be subtracting the Health property. This use makes the Citizen more vulnerable to Sickness, so, keep it in mind.

**Known Spaces property**

Now we got to go upper to the Polis object, remember this is the object that contains all your society –divided by “places”, this is, always, any selected level of Space– and it makes easier to overview your population anywhere you gaze. The “citizens” must to know where are the resources, what is available, but to simplify things a lot, everybody will know what everyone is going to watch.

Now that is everything that is in the same Parcel where the Citizen is located and the Parcels around, the 8 of them. But when the caravan comes, so to speak about the starting point of the game, one Citizen will be placed in one Land, everybody all around will form something likely similar to a circular shape, and they will know what they can, and research starting from there.

Continuing with the previous theme, the Capacity will calculate based on this information, so the starting point is somewhat confused unless they got closing resources to start laboring. If they don't find anything useful to them, they must to do another try following the rule of Expectation goal, so they could accumulate a lot of Frustration points at the start, making a little Shock of it.

**DoWork method**

So, after pay, as it were, the Cost from take the action for Satisfy a Necessity, the remaining Capacity is spent in do the job, whatever it is, to grasp a Product, whatever it is, and here comes a set of modifiers variables for the Productivity to calculate the final amount taken from that job. All this is done into the Citizen class method called DoWork, it takes the Product as parameter, with those properties we get the needed data.

At the beginning you obviously had nothing more than you working Tools, that enhance your Productivity, but you had no Inventory, so you must make it. With no working Tools the effort done is at 100%, barehanded, the Tools usually gives floats above 1 to multiply, just like Cost but opposite, but also could give you the power, multiplying exponentially the Productivity. This last number is the final result after take the modified Capacity to apply in the base Product to work in Inventory, this of course is, for those that aren’t paying attention, the set of “products” used to produce the final Product, the latter which will be placed into Stock property; so, Productivity is the Stock/Inventory ratio.

**Availability property**

We could say that we get a unit for our Inventory per some amount of Capacity value applied, that divisor is the Availability property of the Product object class, but its value is for barehanded job, how we explained above, the final Capacity applied has a Cost (Transportation, Arduousness, isBarehandedTask are the main properties involved in this final value) – when if your Citizen does barehanded job, he wouldn’t do false isBarehandedTask tasks – and has Tools if so, and then the amount taken for Inventory.

Then he must reset the same intended Capacity for the same Activity, but use the current correspondant Tools if so, and reset the Cost conditions because, in this time, the condition to transform, for example in first stages activities, IRawProduct product to IManufacturedProduct product, and so on, there’s always a transformation from one class to another one, and each Product has its own Availability value.

**The Market**

***But... how do the workers exchange their Products in order to create the Product Chain?***

In fact, in the same way that they offer their products to final consumers, through the Market function injected to each Area, and also could be affected in each Parcel if it were the case, but the difference is minimal between Areas, the ones between Parcel are generally negligible. Here's where all the goods and services are supplied and demanded, so we’re going to see much of these two concepts, Supply and Demand, Bid Price and Ask Price, and their tug of war so to speak.

**Demand property**

Based on the “necessities” selected, they are naturally attached to a particular Product, so, the Demand first sort the needed “products” set (including those you have to produce), and taking their “priorities” values the Citizen will be more able to give something of its own property, those are the less “demanded” set of its Capital property, those are the last ones inside of this first arranged demanded product list.

So, we got the less demanded products against the most required, well, this is the famous Demand line, the point where those who want that product more will take the lower Price of exchange –this is particularly complex at the beginning when there’s no currency set up, because it gets several referential prices as it were– of those who wanted less but find an according seller. And those who want it with a lower price will be finding sellers in the next Turn, and will be those who define the next Price of exchange with those sellers who don’t make the exchange because they want it to sell the same Product to a higher Price.

**Supply property**

Same process with Supply property within the same Citizen. But how does the people want to offer their Stock? In what measure? Because this varies across cycles, so to speak, in the first “turns” the offer tends to save “stocks”, just because few people can supply few necessities with a short productive chain, but just when they accumulate more of that they can produce easier, they become to push bid to the market producing deflation, this begins to release hand of work, but let’s not go any further. First let’s solve the first stage, where a Citizen get, for example from some IRawProduct Product, so he just works and obtain a IManufacturedProduct Product that he can supply.

With the products produced aren’t enough to satisfy all the minimum necessities required, so he will not trust over the other workers, he procures work and give out the 70% of his Capacity to satisfy all of them, but the circumstances, and the resources locations, will lead the society to divide their work and value everything according to his provision and genuine Demand, descripted above. Because he will not sacrifice Fatigue points to do something that other people offer for something that he already has too much (unless, as I told, that you as Player force them to do specific things).

So, he will offer the Stock that has and value it with the Capital available and if he lacks something then he will Supply their Stock –or his Capital in the worse cases– at Ask Price, just as before, when the same Citizen, but in the opposite operation, the Products he wants to consume, and at the same time he doesn’t have in his Stock or Capital, he will Demand it at Bid Price.

***Great! Now that we got the entire Market structure, is that all?***

Well, now that we got the Market function, we go to see what happens with the saves that naturally the people are going to be having through the working weeks and months, unless they suffer an external Shock, but when that process happens the people begins to specialize, and the people begin to save Stock and keep the other products unconsumed, so they begin to accumulate Capital

**Capital property**

The Capital is nothing more than the “products” the Citizen bought but not consume, after all, he wants to save not only that of what he produces, the Stock, he wants to save the rest and that is the rest of the Product kinds, then the quantities are set by the ratios that managed among its necessity priorities.

The accumulation starts and a Currency will be preferred, this subject will be explained soon later, and that is the consequence of the massive accumulation of that particular Product. When this process is complete the Capital obtained is easier to develop some primitive level of IFinanceSector “products” to powered the impulse of economy, basically lending the Capital that will not be used, this process is basically the end of all Capital process, lending or borrowing money to create new jobs that are more distant to the consume layer, widening the productive chain length, creating work that will create Tools to do faster the primal work, boosting the final consume output.

The Financial Sector is somewhat magical, it consists mainly in two activities, lend and borrow, anybody that has extra Capital that will not consume will want to lend his money no matter how much, but only those who provide the service of hold, guard but mainly lend all the part of the money that has been lend by the real savers, so we have two layers of money path, the last is the Banker and the first are “traders”, they are “traders” because they buy and sell financial assets, but “bankers” manage that money, literally they borrow that assets saved by traders, and lend a safe part of it to the last borrowers, that are basically the “investors”, but Investor is not a INiche object inherited class, is one of the three IClass object inherited class, that’s because the way to manage its business, but the Capital ends in the “investors” hands, and it’s because of this that Capital is always in use in the productive chain creating new sources of jobs and widening its complexity.

**IClass interface**

This interface declares different ways or structures to make money, one is with your labor force, the most basic way to earn a living, the production chain description all above is basically that Class named Employee. Investors, instead of “employees”, borrow money or take from saves (this is practically unusually) to create and manage a business, so they work substantially different. And the last ones are the “owners” that charge a fixed rent (fluctuant as every Price) over a Building (this object includes open field as human property).

Practically “employees” are explained, everybody starts as an employee if they start with nothing more than their bare hands. Now we got to understand how does the Investor and Owner business works, and of these two, “owners” have the simplest part to explain, they offer a property or Product from IRealStateSector interface, the Rent is the fixed income they gained. A Citizen can have several “classes” at the time, so they can have a Rent while work in another areas. The Rent is also, of course, the value or current Price of a leasing Product, and fluctuates accordingly.

**Investor IClass class**

Investors practically takes money to buy things in order to create a Business. Business are sets of products that together can produce a final one. So, they take several products as “fixed costs”, once bought they just wear with a WearTime property, that decrease its value at a set time; and then they hire the “variable costs”, the “salaries” are properties from the INiche interface classes, also variable costs are the products considered the inputs, this will give them the marginal profit, and they will try to reach the breaking point, whether they can invest.

They have a CommercialBalance property as well, to measure their “assets” and their “debts”, this is important to deduce their Solvency and Liquidity, properties that are required as parameters for “traders” to invest, but mainly for MoneyFlow calculations, this mechanism –that will be describe later– influence strongly the RealFlow, mechanism that measures the Market flow, or goods and services provision. A shortage in the Liquidity causes a Liquidity Crisis, so terrible as Shock Crisis.

**Business class**

A Business is conformed by “fixed costs” and “variable costs”, as we said, but both can change along the time of course, the difference lies in the response on the final Output Quantity, Fixed Cost does not change if the Investor produces more or less, Variable Cost does, this is why it’s related to marginal profits or marginal costs. Then a Business have a Goal, this is the last Necessity behind the Business, set by the production chain or ISocietySector interface shared with other Business colleagues, but Goal is depicted by the Output, Goal is for final consume info, and Output is the actual Product made.

I depict all Business in the Business Table excel document, please consult there to see every Business with all their requirements.