Real-World Object Oriented design example

In my class I learned about all the beautiful things that an object oriented programming can do, but I remain confused about why object programming is better than others? I can write code in Java and C++ like I can write code in C. I always pondered upon the common features of an object oriented programming like inheritance, polymorphism and encapsulation etc. But I could not figure out the real benefit of all of these features until I design some of the projects using object oriented methodology and I am sharing one of these projects with you.

Here I developed a software product for one of my client couple of years ago and I will share the client's requirements, the approach that I used for software development and object oriented design of the software.

I started developing this product for one of my client a couples of years ago. Client ask me something like this "Umair I want to develop a desktop software where I can view satellite images". I said you can do this with Google Earth software where you can browse satellite images quite easily. My client said "Google have their own images for which they have the publishing rights therefore we cannot use them but we have our own satellite images for which we have bought the publishing rights". I said o.k then you can use any other open source software which can work with any satellite imagery. Client said "Well, we also need some custom features"

I replied "o.k but I have no experience of developing any such software so why don't we discuss the requirements in detail" and the client responded positively.

At this point I know basic information about the project but I know one thing very clearly that a critical and important feature of the software is going to be "**to browse the satellite imagery**".

1. The Beginning

I started working on this highly important feature (browse the satellite imagery) and started research on the satellite images and cartography.

Next thing that I did was to write a simple program to just view a single image on the screen.This take me around 3 weeks to do. Because I have to implement certain projection algorithms and understand how the satellite images work and what are some distinct characteristics of a satellite image. This step is called the proof of concept. I know that if can just render a single satellite image in my software then I can design a software which can render more than one image.

Up-to this stage I have discuss with my client at least two times and finalize some of the top and critical requirements of the project. At the end of the project I realize that these initial requirements were only 20 percent of the total requirements that I have implemented during the lifecycle of the complete project but these 20 percent requirements were the most

essential and important requirements. 80 percent requirements will be addressed later in the project but initially I will develop software to fulfill these requirements only.

Therefore the key idea while developing a software is to identify the key requirements or critical requirements and then develop the software to fulfill these requirements as soon as possible.

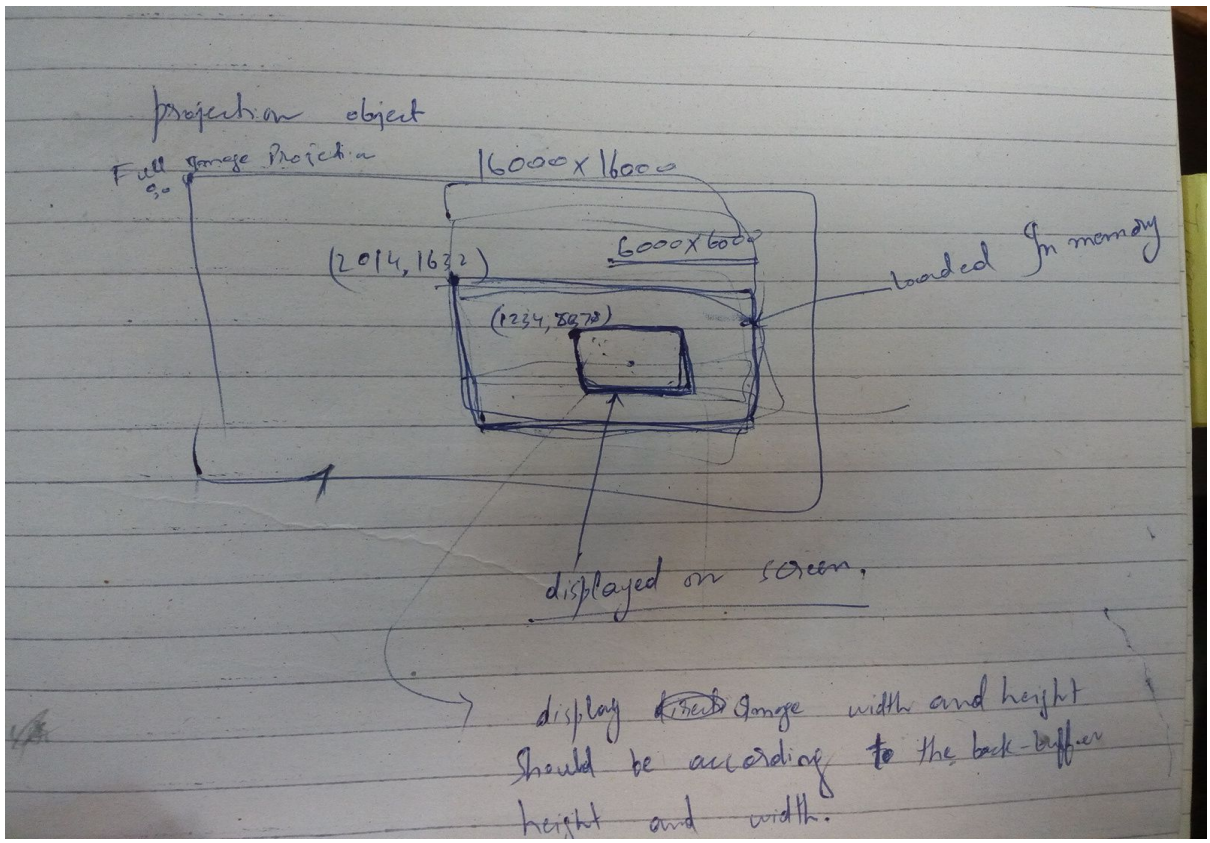
1.1 How much time is taken upto this stage?

Since it is the very start of the project this requirement gathering and proof of concept takes around 5 weeks to complete. Most of the efforts were invested in the proof of concept programming.

1.2 What else to do at start

At the start of the project the most important is to understand the problem, highlighting the critical features of the software and doing some programming experiments to determine that you can actually solve the problem. In addition to that any non-functional requirement that your client requires such as performance and/or reliability should be fully understand. In my application client needed the data offline and want to load the huge images into working memory when the application starts.

To understand the problem I use drawing the problem on a rough paper. Let me share a rough mental model to understand the problem better which I created into my working journal:



You can see nothing is fancy here. I just draw a rough visual model on my journal to help myself understand the problem. From my experience I can tell that drawing such diagrams is very helpful in solving any problem. Even such a rough diagram can help you tremendously to reduce the complexity of the problem as you put your mental model out of your mind and onto the paper. This will help your cognitive ability to work more efficiently as most of the information is available easily from this rough paper model.

1.3 The Approach

My approach will be to work in iteration and in each iteration I will develop some user requirements. Length of each iteration will be 3 weeks to 6 weeks. At the end of each iteration I developed enough to demonstrate to the end user and take inputs and suggestion. There are many approaches and process models to develop a software but let's stick to my simple approach.

I do not recommend investing 5 weeks at the start of each iterations. This is acceptable only at the start of the project. For each Iteration one should only invest from 2 hours to 2 days only.

2. Requirements Gathering

From all the discussions with the client I understand the requirements, solve some critical programming issues and draw rough visual model of the problem and solution.

One can write a detail requirements document at this stage but I didn't do that. I write requirements in a very simplest form and it consist of one page only and it represent only 20 percent of all requirements. Remaining 80 percent requirements were decided and implemented later in the project. Next is the user requirements which I have written for the first iteration.

2.1 User Requirements

There are a lot of methods to capture images of our planet and then translate them to maps. The scientific field that deal with this problem is called the cartography.

Earth is elliptical in shape and to represent this elliptical object into two dimensional rectangular object (paper map or image) we have to apply process called the projection. There are many projection methods but our client's requirements is to use mercator projection.

Satellite images of the earth are available at client's database and in mercator projection format. Satellite images are huge images at different zoom levels. To store them and use them in computer these huge satellite images are stored in small chunks called tiles. These tiles are of size 512 X 512.

Maps are stored in tiles format with respect to the projection used (Mercator projection in our case). User needs to browse the satellite imagery at different zoom levels on the computer screen and show some custom overlays on the images such as name of a particular feature on the map or highlight a particular area on the map.

One cannot load all the images at once into the memory as there are 10s of gigabytes of images at a single zoom level. There are 19 zoom levels. Therefore images displayed on the screen will be according to the screen resolution of the user.

As user move the screen the tiles related to that area will be loaded into memory and displayed on the screen. If user change it's zoom level then again tiles at the selected zoom level will be loaded into memory and displayed on the screen.

3. Iterative Development

At this stage I have understand the user requirements at highest level, identify top or critical requirements of the client, done some simple programming tests and write the requirements description in a simple form.

Next task is to design an iteration around these requirements to develop the software. Developing software in iteration is a most common and successful software development approach. There are many widely used software process model(another name for the approach to software development) which are based on iterative development such as rational unified process model, SCRUM and XP. My simple iterative development approach has adopted concepts from these process models.

The most difficult work in iterative development is defining the requirements to develop in each iteration and size of the iteration. One tip here is that select those requirements which are critical to your clients in the early iterations and leave less important requirements for iterations later in project.

At the end of each iteration a working model of the software must be delivered or a demo should be given to the end user. This is to ensure that you are on track by taking the customer input at the end of each iteration.

As far as size of the iteration is concerned one should consider the size of iteration from 2 weeks to 6 weeks maximum. This is because no significant work can be done in less than 2 weeks. Also, iterations greater than 6 weeks will be too risky if whatever you have developed is not required. For example when the customer look at the demo after the iteration and says 'NO' all you can do is follow this principle "Customer is always right!" and discard/change the work that you have done in that iteration.

Upto now I have only discussed the process (iterative development) that I have used in this project and discuss the very first stages of any software development process (requirement gathering, prioritizing and analysis). Next stage in the process is related to software development methodology.

4. Object Oriented Design

Object oriented is the methodology that I have used for developing the software. There are numerous benefits of using object oriented methodology but most important is the capability to translate the real world object into the programming world. One can easily model any real world entity into the programming world object using object oriented methodology. There are other methodologies such as procedural and functional. For example Visual Basic 6 was a procedural programming language.

When I was in college I developed a simple database application for a medical pharmacy store and I used visual basic 6. To develop a software with moderate complexity using Visual Basic 6 was such a pain. Because each time I change in one feature of the software other software feature get disturbed. Correcting the problem in one part of the software was causing problems in other parts of the software.

Once I start using object oriented methodology to develop a software then software development started to look easy. One can model the real world and understand how any entity work in the real world and easily translate that into the programming world with the help object oriented methodology.

4.1 Domain Model

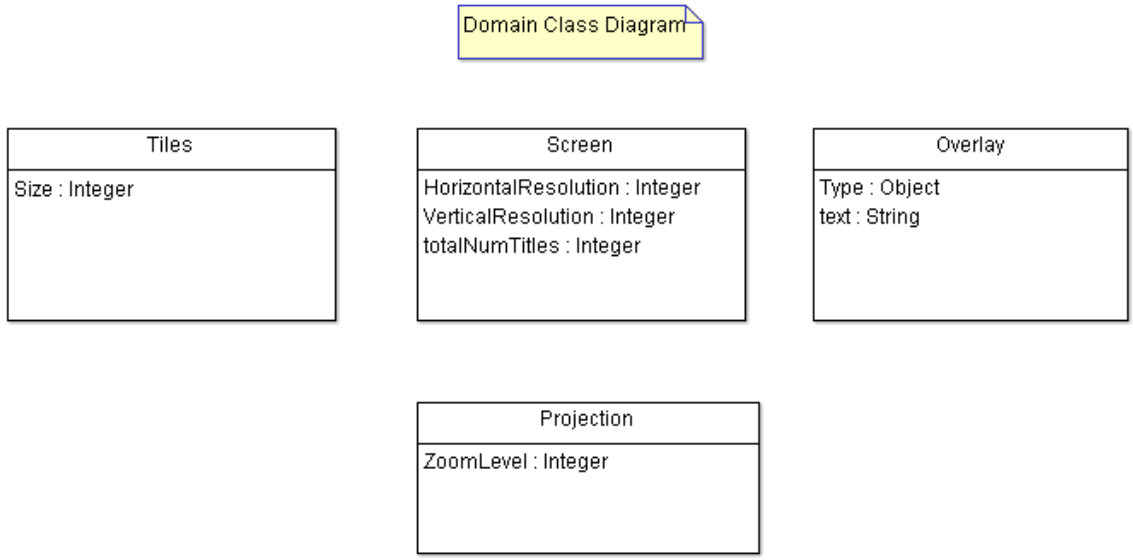
First step in object oriented methodology is to develop a domain model of the intended software. This domain model represent the real world objects with attributes. In domain model we try to identify the entities of the real world or the domain which can be represented as class. Attributes of an entity and/or association between the entities are also identified in the domain model.

A domain should not describe GUI or any other software related entity unless you are designing for a software where these elements are itself the domain. For example, if you are designing for a rendering application then ‘window’ and ‘canvas’ can be your domain entities. But if you are designing for e-commerce website then you cannot take ‘window’ as an entity.

One extremely helpful tip while designing a domain model is to read the user requirements and highlights the nouns/names of entities and then put them on a paper or any modeling tool that you are using.

You can stop at this stage and develop a domain model for yourself from the description of the user requirements above. No fancy diagrams, you will be good if you just come up with the names of some of the classes that you can think will be useful and could fulfill the user’s requirements.

Here is my domain class design copied directly from my journal which I redraw in a UML tool for the purpose of this report.



This is the very first and important step of the object oriented design. This is where the real world and programming world comes together. This is an important feature of object oriented methodology that you can easily translate a real world into the programming world.

As you can see that this a very simple domain model. Only 4 classes are identified in this iteration with some attributes. There are no associations(you can draw for your domain model if you want to) and very few classes. But at the end of the project the complete class design (consisting of 26 classes) contained these 4 classes with very minor modification.

In your domain class design you may have chosen different names or a complete different domain class design. This is fine at this stage because you were not in my place who was designing the domain model after 5 weeks of learning therefore your model maybe completely different than mine.

I chose ‘screen’ class to demonstrate the screen through which the user is interacting and another reason is that in this application every pixel on the screen represent some information (such as latitude and longitude). ‘Screen’ also defined that how much data should be loaded into the memory.

Projection class represents mercator projection. It contains the information about all the tiles at a particular zoom level.

‘Tile’ class represent an individual tile and a ‘Screen’ class can contain more than one tile. ‘Overlay’ class define the type of overlays and at what point/area they will render onto the screen.

When people ask us to design in object oriented way we thought about polymorphism, inheritance and many such object oriented terms but in the design above there is no inheritance or polymorphism just raw classes with some attributes. This is all you have to do while designing domain class model.

Next step is to design an interaction design.

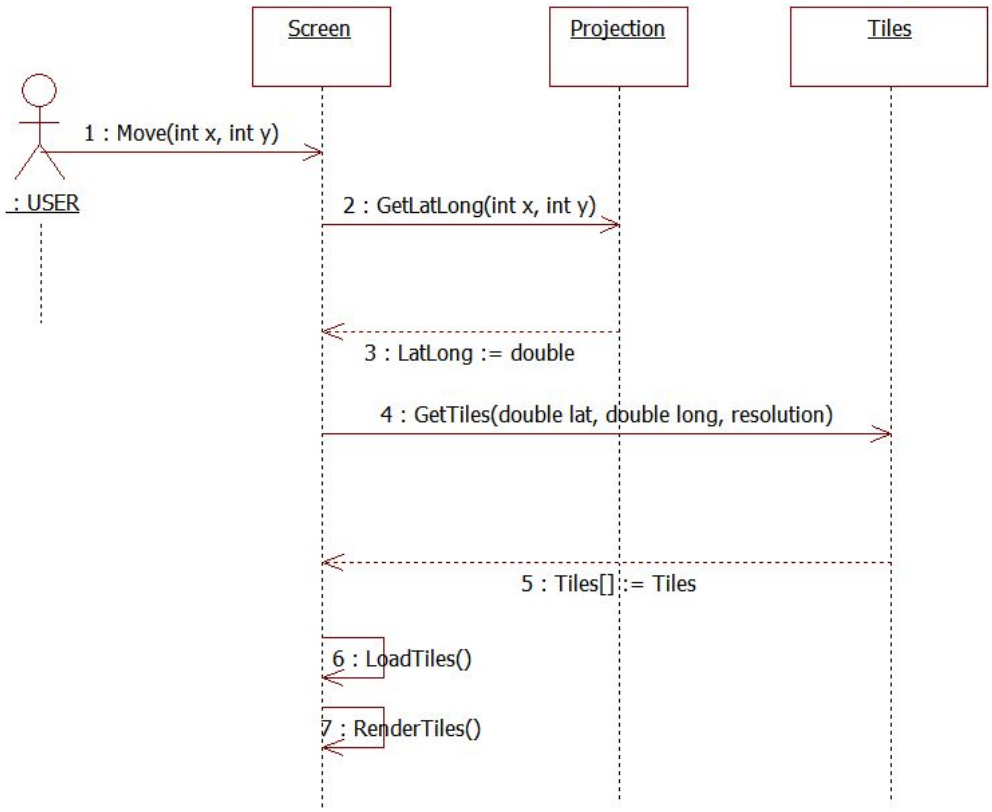
4.2 Interaction Design

Till now in this first iteration I have decided all the possible classes or entities that will fulfill my first iteration requirements. But classes itself does not run a software. The most important part of any class is its behavior or in simpler words the functions that it contains.

Therefore to model the behavior of the classes identified in the domain model design we use interaction design. An interaction design shows the objects of classes and the message they pass to each other (in simpler word function calls to each other) to fulfill user's requirement.

In all activities of object oriented design the interaction design is the most important part. Through this design we can design how the objects of classes will interact with each other.

To design an interaction diagram for my project I analyzed what were the requirements of the user and the very first requirement is to browse the satellite images. To browse a satellite image a user has to move the screen.Following diagram shows the message that will pass between objects of different classes to fulfill this user requirement.:



The number near the message arrow shows the order of the function calls. The message also shows the signature of the function such as parameters and return types.

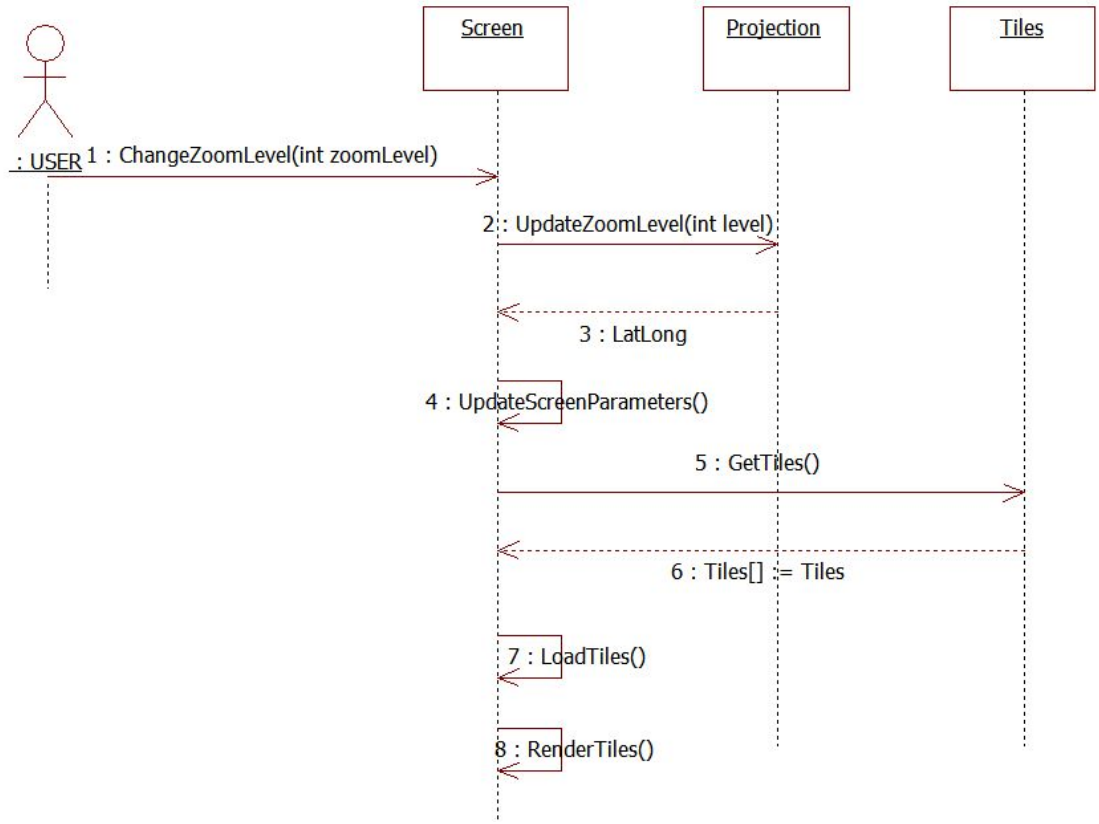
When the user move the screen then the information related to the changed screen needs to be updated. Therefore first message is sent to the ‘Projection’ object which will update the latitude and longitude values according to the current screen position.

Then with respect to updated latitude and longitude position application needs to update the tiles and for that it calls upon the ‘Tiles’ object to give the new tiles for that position. In return the ‘Tiles’ class return the related tiles. Note that here ‘Tiles’ object does not loaded the tiles it just gives the information about the tiles such as resolution and location of the tile.

In the end the ‘Screen’ object load the tiles and then render these tiles on the user’s computer screen.

Another action that a user can perform while browsing the satellite images is to change the zoom level of the images.

I modeled this behavior using interaction diagram which is shown below:



This is very similar to the first interaction model except few new function.

In this iteration I only drew these two interaction diagram and I did it manually on my journal. For this report only I redraw these interaction diagram in a tool to make it readable to

a large audience as my handwriting is readable by me and by my close friends and colleagues.

One tip while designing an interaction diagram is that treat the objects like real world entities. Such as when you browse anything on your computer your screen is changing and the contents are coming according to their location.

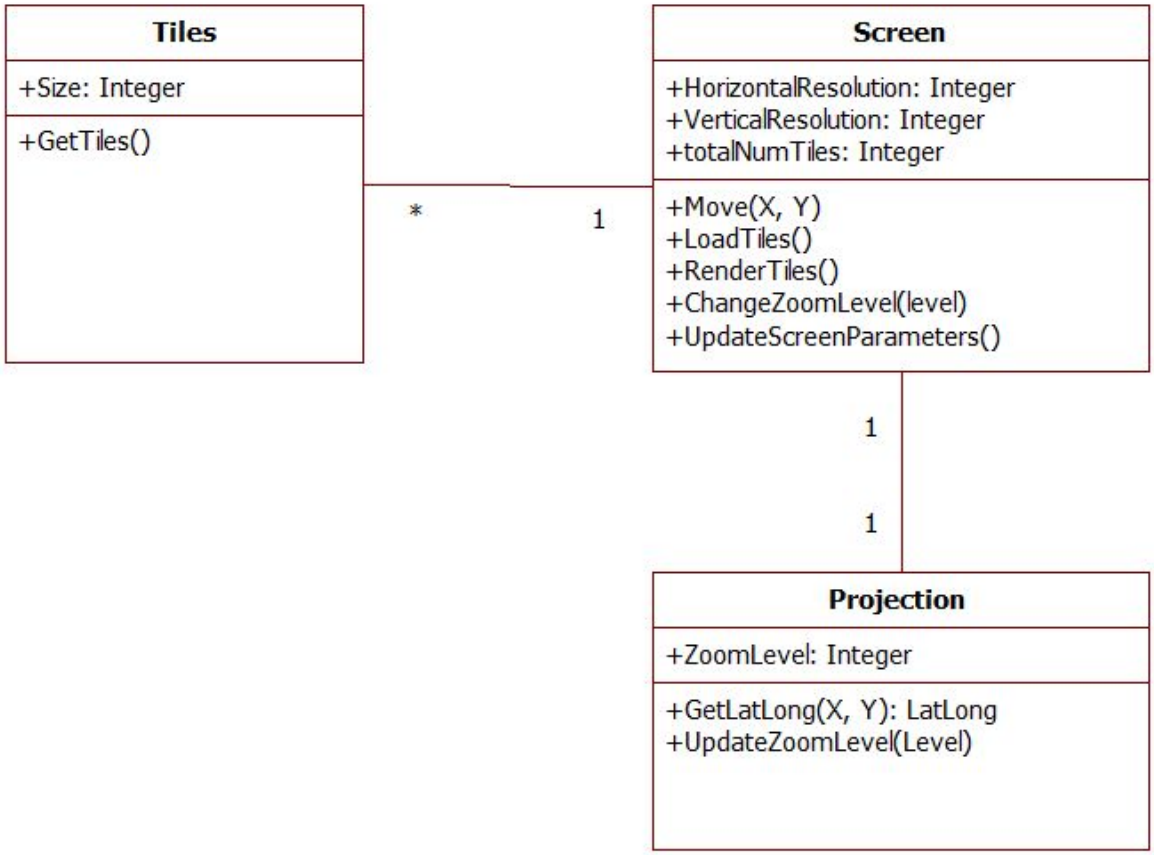
After interaction design I can easily design the next artifact and that is implementation design and that will be the last design artifact for this iteration. After that I will directly go into the programming which consist of more than 90 percent of the total effort for this iteration.

4.3 Implementation Design

As it is emphasized earlier that interaction design is the most important activity that you will do when designing using object oriented methodology. This is because the decision made during interaction design will impact in implementation design and in the programming.

Now by looking at the previous step one can easily design an implementation model. All you need to do is to look at the messages which are passed between the objects.

In the first interaction diagram the external user or system passes a message to the ‘Screen’ object. Therefore the screen should fulfill the responsibility to implement the ‘Move’ method. In simpler words a ‘Screen’ class should have the function name ‘Move’. Similarly one can easily assign the responsibility to each class using the interaction design and this is shown in the following model which is called the ‘Design Class Diagram’ or the implementation design:



Note the difference between this design class diagram and the domain model. This class diagram is more closer to the programming world as it shows the actual methods which we will implement during construction or programming.

This design class diagram shows all the messages as methods as they are shown in the interaction design. Here each class has been assigned a responsibility which will be programmed in the next phase which is the construction phase.

This is all the work that I completed for designing the objects for the first iteration of my project. There were minor changes at the end of the first iteration such as I added more responsibilities to each classes during implementation and added new attributes.

One can also use design patterns to assign responsibilities and structure the associations and classes. But I haven't applied any design pattern in the first iteration. Although I have applied some object oriented design patterns and principles in later iterations and I will share them in separate reports and posts.

5. Programming, Testing ,Demo and then Iteration 2

Let me share some programming snippet with you to show you that how I write a program using the design above. Consider the first interaction implementation design and first interaction diagram, my ‘Screen’ look like this:

public class Screen

```
{

        Projection currentScreenProjection;
        Tiles[] currentScreenTiles;
        int  horizontalResolution;
        int  verticalResolution;
        int  totalTiles;
        public void Move(int x, int y){
                //Implementation
        ……...
        }
        …
        ...
}

public Class Tiles
{
        int size;
        public Tiles[] GetTiles(double latitude, double longitude, int resolution)
        {
        …..
        }
}
```

And similarly other classes were implemented. I added many local methods to these classes and added other responsibilities during programming. I implemented projection algorithm, filing and rendering method as well.

90 percent of the work during this iteration is programming and testing. At the end of iteration I gave a running demo to the end user and record the user input and suggestions and plan these suggestions for the next iterations.

A customer initial requirements can contain more than one iteration but each iteration should end on a demo to the user and user input should be incorporated as soon as possible.

In second iteration I again discussed with the client and record new requirements and then again start with the domain model then interaction design and finally implementation design. As the design grow bigger than before I incorporate design pattern to make the project more manageable.

6. Final Words

There were total 6 iterations and at the end of the final iteration my design class diagram contained more than 26 classes. My designing time never exceeded more than one day in any iteration. After each construction or programming phase or at the end of each iteration I update the my design according to the understanding gained during programming.

Was there any benefit of using object oriented methodology to design this software? Simple answer is yes. When my client asked me to make a change or update the software

then it was a piece of cake for me. I just know where I have to change and that change did not break the code unexpectedly (80 percent of the time). If someone asks me to add a new feature then I just know where I have to write the code and how I should write the code.

With object oriented design I can code faster (but it took pain to initially design using more than one class) and understand the design better. Extend the software better and even use part of this software in other projects.

People think that object oriented design is about UML or just diagrams. No it's a big mistake if you just thought about just learning diagramming conventions or UML designing tools. The key objective of object oriented design is to convert the real world problem into a programming problem. For that these three tools : a) Domain model, b) interaction design and c) implementation design and this simple process which I have demonstrated in this report is absolute basic and I have used these basics throughout my career.

There are additional important things such as object oriented design principles and design patterns but the focus of this report is on converting the real world into programming.I will discuss design patterns and design principles in other reports and posts and if you have any question you can ask them at this email: umair@objectorienteddesign.org.