

# Dynamic World Design

By Alun Hancock, June 2013

## *Introduction:*

This document is intended to be a guideline to assist with the pre-planning and early design of a Dynamic Multi-Player Game World such as a MUD, PBM, PBEM, MMORPG or Rogue-like engine.

## *Multi User Dungeon (MUD)*

These are typically text-based adventure games where users interact by directly typing their commands using traditional words and text. They are frequently played via complicated Telnet services with limited interface capabilities. Operators and moderators frequently make changes to the game world to manually perform most dynamic elements to support the current chapters of their story-telling.

## *Play by Mail / Email (PBM / PBEM)*

This style of gaming originated with postal gaming systems where turn cards are sent to the games master who processes the results and sends back a printed reply. Real-time servers, automated emails and online processing systems have allowed the PBM games to evolve from basic pen & paper replies into graphically rich games with hand-moderated content allowing for specialised gameplay to be tailored for each player. The Games Master will occasionally make changes to the game world as a direct result of player actions.

## *Massive Multi-player Online Role Playing Game (MMORPG)*

These are the studio produced, mass-market games based mostly on generic adventures with every player performing the same quests in the same location for the same NPC. They contain the best visually and auditory experiences but lack the depth of rich story-telling and flexibility of a hand-moderated world. Any appearance of dynamic elements is either entirely artificial and only occurs as a direct results of the player progressing through a pre-written storyline.

## *Rogue-like*

Take the worst elements of all of the above and put them together – text based graphics and input methods, turn-based movement and events, typically single-player oriented storylines – and somehow logic defies explanation and you get a much more compelling game style than any of the above. Go and play Angband, NetHack, Ancient Domains of Mystery, Dwarf Fortress or DoomRL. While Dwarf Fortress is an exception, these games have no dynamic elements that occur away from the player's direct interaction. It is not possible to simply sit back and watch the world unfold.

# Dynamic World Design

By Alun Hancock, June 2013

## Contents

|   |    |
|---|----|
| Chapter 1: Fundamental principles of a Dynamic World.....           | 3  |
| Definitions and Purpose of a truly dynamic world .....              | 3  |
| Practical theory of a dynamic world .....                           | 4  |
| Individuals, Settlements, Societies and Conflicts.....              | 5  |
| Chapter 2: Good Guys vs Bad Guys.....                               | 7  |
| Who are the Bad guys? .....   | 7  |
| Who are the Good guys? .....  | 8  |
| Bad-Guy Lairs and Raiders .....                                     | 9  |
| Chapter 3: How the World is made and populated.....                 | 10 |
| In the Beginning, there was the map.....                            | 10 |
| Evolution of the terrain by flood, earthquake and devastation ..... | 12 |
| Building Templates, Masters, Apprentices and Workers.....           | 13 |
| Age, Death, Disease, Corpses, Graveyards and Necromancers .....     | 14 |
| Chapter 4: Alternative World Layers .....                           | 15 |
| The Pyramid World .....   | 15 |
| The Great Underworld .....  | 16 |
| The Boat People .....   | 17 |
| The Sky World .....   | 17 |
| Chapter 5: Specific Data Structures .....                           | 18 |
| Individuals .....   | 18 |
| Buildings.....  | 18 |
| Settlements .....   | 18 |
| Societies .....   | 18 |
| Alliances .....   | 19 |
| Lairs .....   | 19 |
| Raiders .....   | 19 |
| Chapter 6: Summary and Technical Footnotes.....                     | 20 |

## Chapter 1: Fundamental principles of a Dynamic World

### Definitions and Purpose of a truly dynamic world

The term “Dynamic World” is misused in the gaming community as much as the words “unlimited”, “infinite” and “literally”. Let us begin by defining a proper meaning to the term Dynamic World in the context of an open-ended multi-player game.

In most games the world is static until the players complete a pre-written quest that affects a change. For example a ruined castle will stay ruined from the player’s perspective until that player completes the quest to rebuild the castle. For the player that has completed the quest it will then show as a completed castle with NPC services that only the individual player can use, but to everyone else it will still be an empty ruin. This is a static world. Only a single player’s viewpoint has changed and unless further changes are scripted into the quest chain then the viewpoint does not change back. Even if the player doesn’t visit the castle ever again it will remain a castle and never fall back into ruin, unless of course it is Schrödinger’s Castle.

In a partially Dynamic world, the castle would fall into ruin from inaction. In a fully Dynamic world the castle could continue to run via the actions of the NPC population who maintain the castle. Outside events such as invasion forces may cause critical damage to the castle system that it falls to ruin, but as long as the builders have resources to repair the walls and the farmers provide the food to keep everyone fed then the castle should remain intact.

Note that the soldiers might leave because nobody is there to pay them and the builders and farmers may be enslaved by the invading hordes but the castle will remain intact.

Dynamic worlds work because of the automation of the systemic elements. The bad guys will gradually get stronger and invade. If they start to win too much then they split into factions and end up fighting each other, consuming the excess resources and reducing their strength again. The good guys will keep growing, encroaching on hostile land, getting attacked, and drawing resources away from the safe-havens to maintain their territory. The world powers will swing back and forth, always fluctuating, never becoming extinct. The entire system is based on the balance of a pendulum, as something grows too far, its strength is reduced by the greater number of obstacles and it incurs greater losses. As the obstacles grow in power, they find the resources available to use are no longer being generated as they decline. Everything balances out in the end. The further you sway to one side, the greater the pull back to normal.

The purpose of a dynamic world is for events and changes to take place, logically, without any players being present in the game. Quests are not generated from a set script or by selection from a random list. They are created by the progression of natural events in the game world.

The style and flavour of a dynamic world is not fixed to the Tolkien genre of fantasy gaming and can be applied to any technological level of fantasy or science fiction realms. The world map doesn’t even have to be typical Earth-bound terrain and can be made from Planets and Galaxies at the highest scale of space-faring adventure, down to patches of rock and dirt for an amoebic adventure.

## Practical theory of a dynamic world

The basic theoretical blueprint for a Dynamic World is based upon jobs and societies. Each individual within the world is assigned a job type and a set of locations or radius where this job is performed. This job contributes to the society where the individual lives. I'll start an example with the basics of any society, which is usually food.

A settlement is created with a base population of unskilled workers and is assigned its own Society. The settlement then assigns a plot of land and the basic farming skill to an unskilled worker. The resources for building the farm are removed from the society's stockpile and the farmer moved from the general population and placed into the farm location.

There are some background logistics that take place here that are invisible for now. The security radius of the settlement and the suitability of the terrain will be used to decide where the farm is located. Farming is considered a basic skill that can be assigned to unskilled workers, so you don't need to have a teacher to learn basic farming. As a farmer learns, either from a teacher or from experience, their skill will increase to improve the productivity of the farm and unlock new farming products. These will all be discussed later.

Without any involvement from the players, the farmer will perform his assigned tasks according to the rules of his profession.

- If there is a barren field within radius and seeds are available then crop fields are planted.
- If a crop field is ripe for harvesting then crops are harvested and the field becomes barren.
- Seeds are stockpiled at the farm for later use.
- Excess seeds and harvested goods are delivered to the Society.
- If there are hostile entities in the area the Farmer will post a Quest with the Society to remove the hostiles.
- If the farm is damaged the Farmer will post a Quest with the Society to request assistance with the repairs.
- If the farm is destroyed the Farmer will return to the Society's general population, retaining his Farming skill for later use.

There are several continuances of these actions that do not require player-intervention. If hostiles are reported then any available soldiers will be despatched by the Society to deal with the threat. If the farmer is also skilled as a builder then he can gradually repair the farm himself. If the farmer is killed then the farm will fall to disrepair as there is nobody present to post a request for repair to the society.

All quests should be made available to the players with the Society only taking intervention measures when the players fail to respond in time. There are Quest options that will not be dealt with automatically by the system. If the farm is invaded by bandits then the farmer may be kidnapped and enslaved. This will result in a Quest to rescue the farmer. The local security forces will not deal with this. Only dedicated "Adventurers" would undertake such a mission and typically this role is reserved for the players.

## Individuals, Settlements, Societies and Conflicts

The basics of map design and the ability of a dynamic world to provide a continuation of events relies on the careful balance and placement of Good Guys vs Bad Guys. In a dynamic world these are handled by the placement of Conflicts around the map. A Conflict is a Bad-Guy Lair that will work against any nearby Settlements and other Conflicts. Settlements are the Good-Guys, but owing to their allegiance to a Society they may find themselves as the Bad-Guys in some scenarios.

The NPC population provides the individuals needed to maintain that population and provide the protagonists required to create an interesting world. These are the farmers, builders, merchants, lumberjacks, butchers, millers, bakers, tanners, armourers etc. Each individual belongs to a settlement, which is a central home from which they work. Individuals will post Quests within their settlement to acquire the goods they need to maintain their work.

Farmers will request seeds for planting, wood to fence the borders of their land, and builders to help repair damaged fences. Builders will require tools and use wood and stone to construct new buildings. Millers will request grain and water to make into flour, which is requested by bakers to make bread. These requests are treated as Quests and will affect the price of items purchased at the stores, and the missions made available to players from the settlement.

The settlement is made of two different types of building. External buildings are the farms, mills, lumberyards and quarries that are located outside of the settlements physical boundary. These external buildings must exist within the settlements security radius otherwise they become independent settlements of their own. Internal buildings are the basic government buildings such as a Village Centre or Town Hall where Quests are offered; manufacturing buildings such as bakers and butchers; and training halls such as the barracks, archery range and guild halls. The settlement's defensive fortifications and military population will determine its security radius. This is the distance of control it has over the surrounding area where individuals who work externally to the settlement are able to quickly communicate and interact with each other. From a military viewpoint, if an outlying work camp is attacked then a warning can reach the settlement and soldiers are able to leave the settlement to protect the camp quickly enough to intercept the attack. From a production viewpoint, the security radius defines how far out a worker can be to place requests for local goods before it requires a dedicated supply wagon and merchant dealer to make the delivery.

Every individual and settlement (not buildings) belongs to a Society. This is the Governing State, Organisation or Guild that rules over that entity. This is important when a settlement is hurt enough that its security radius is reduced and outlying camps become disconnected from the settlement. When the radius increases again and the camp returns to the settlement's control, it does so peacefully as it is part of the same Society and is not considered to be a hostile takeover. Buildings are not aligned to a Society. They are just hollow shells that perform a function. If the individual who works there is removed and a member of another society arrives and takes over, then the building and its new owner are no longer controlled by the parent settlement that created the building. This allows outlying camps to be invaded and a hostile take-over initiated. This creates a Conflict Event.

There are three types of Conflict in a dynamic world: A Conflict Lair that spawns the entities needed to create an event; a Conflict Party that is the entity which causes the event; and the Conflict Event itself. I will explain the Lair and Party entities later.

A “Friendly” Conflict occurs (or to be exact: an event does not occur) when something within the security radius of a settlement belongs to a different society. If that society is part of an alliance then the issue is typically ignored, or resolved as part of the alliance pact. For example, any work camp that falls within the radius of an allied society is transferred to the control of that society. This allows allies to give resources or help each other to grow and development their infrastructure. A neutral pact agreement would simply be to leave the camp under the control of the creating society and ignore its presence within the security zone. These pacts would only change due to player or Games Master intervention. Under the basic flow of the dynamic world, all societies will seek to eject any conflict points from their security radius. It is important to plan the default pacts between societies carefully. If a settlement in the high mountains were to establish a lumber camp outside of its security radius, then that settlement will become its own society. To ensure that trade and defence can be established with their own camp, the parent settlement society will create an automatic alliance pact with the new camp society. Any quests generated by the camp can then be fulfilled by the settlement, and vice versa.

A “Hostile” Conflict occurs when a hostile society has resources within the security radius of another society’s settlement. The settlement will issue Quests to deal with the conflict, and if there are no players to deal with the issue then the settlement will seek to dispatch soldiers to remove the conflict. Things get interesting when security zones from hostile settlements overlap. As troops from one side enter the security radius of the other further conflicts are created, leading to the escalation of troop movements and hostile actions. To stabilize this escalation spiral, each settlement should keep a fixed defensive force in reserve to avoid over-escalating and also to ensure protection against sudden extinction. If all of the troops suddenly leave the settlement undefended to go off and invade then the dynamic world is going to break as the settlements bounce ownership between invaders and counter-invaders.

Conflicts do not just exist from Good-Guy settlements having a go at each other. The Bad-Guys will form raiding parties to wander the world and attack anything that belongs to another society. Bad-Guys will always be hostile to other societies. They do not form natural alliances as part of the dynamic world’s events and they do not build settlements. Only player or Games Master controlled events will cause a Bad-Guy society to ally with another society or to build a settlement of their own. As part of the initial world generation, and possibly when a ruin is conquered by invaders, the GM may allow a Bad-Guy society to build a single settlement structure, such as a bandit hideout or fortified tower. This will be entirely dependent on the natural or racial ability of the society members. It would be inappropriate for a wolf-den to build a fortified outpost, or for bandits to build a farm, windmill and bakery. That defeats the purpose of being Bad-Guys and gives the NPC a more important task than causing the conflicts that drive the dynamic aspect of the world.

## Chapter 2: Good Guys vs Bad Guys

### Who are the Bad guys?

The bad exists to cause conflicts all across the world map. They are spawned in sufficient numbers and regularity as to provide a challenge but not overwhelm the players and good guys of the world.

The Bad-Guys also don't have to be particularly Evil in alignment. Peaceful creatures can make an appearance as members of their own Bad-Guy society that trigger a default conflict reaction to be ignored or treated as friendly parties, unless certain conditions are met. A wandering herd of sheep would not be attacked by the Good-Guys unless the settlement is short of food. If they are attacked then the intention is to capture / enslave the sheep. A skilled shepherd would have the ability to convert the sheep to join his society, reducing the amount of security (farm hands) needed to maintain control and prevent the hostile sheep from escaping or damaging the farm.

Unfriendly animals such as bears and wolves, bandits, monsters, aliens and robots, world-eating space beasts and tiny germs can all play the part of a Bad-Guy based on the genre of your world. The dynamic events will play out the same provided you can give the Good-Guys a reason to expand and the Bad-Guys a place to wander.

If the Good-Guys are not around, the Bad-Guys will turn on each other. This is the reason why Bad-Guys should not form alliances with other societies. The alignment or attitude of the Bad-Guys will decide if they directly assault other society members and if they retain control of captured buildings and settlements.

A wolf pack will move to attack a farm and kill the farmers, but they won't keep control of the farm. Instead they will keep wandering the world afterwards.

Bandits will also move to attack the farm, but instead of leaving it derelict they will attempt to keep control. As long as the bandits keep the farm they will continue to grow in power and eventually release further raiding parties from their new stronghold. To stop the bandits from growing too strong and eventually dominating the world the new bandit stronghold will have its own society. As the bandits expand, they eventually end up fighting each other and restoring balance to the world population.

Despite being Conflict triggers, Sheep will not attack any other entity. If they walk into a farm location, they will ignore the farm and continue wandering until they are slaughtered by wolves. Poor Sheep.

## Who are the Good guys?

The good guys are the civilizations and empires of the world. They build, grow, research, learn, teach and supply. There is no assumption that they are Good in alignment or that the player is able to interact peacefully with all of the Good-Guy societies. It's entirely feasible to have an Evil aligned city-state that will attack players and other outsiders on sight. These Good-Guys are there to pose an intentional and deliberate threat as part of the story line, or to provide the basis of a player-vs-player element.

Every individual Good-Guys NPC will belong to a settlement and society. The NPC carries out its assigned duties based on some basic rules. Gatherers of raw materials will live externally to the settlement and have a choice between maintaining their buildings, seeding new resources, gathering ready materials and handling the transfer of goods with the parent settlement.

The parent settlement does not have to be a village, town or city. It could be a central work camp, such as a lumber yard or storage area. The external buildings will be the individual cutting sites for the lumberjacks to work and the internal buildings will be a basic storage area and merchant yard. The work camp settlement then manages merchant caravans to transport goods back to the allied city society, in exchange for tools, workers and guards to protect the site. If the external buildings are threatened then a Quest is generated with the parent settlement, which is the lumber yard. If the yard does not contain any troops to send out in defence, then the Quest is passed along via the society alliance to the city, who will decide how to respond via the alliance agreements. Since the lumber yard is an important camp for the city they will have a defence agreement and will send out troops to help in the defence. As the lumber yard is likely to be far outside of the city's security radius then troops will take time to get there and could be intercepted on route.

Settlements have a priority list of goods that need to be acquired and buildings that need to be constructed. Food, Water and Defence will take priority over other requirements until they reach a suitable stockpile level. Tools and basic resources are then acquired, followed by more advanced items and services. The settlement uses its alliances to trade with other settlements. Any surplus of items above the stockpile level will generate a Quest to sell the excess items, and the items needed to construct or supply the next building or project will generate a Quest to buy the needed items. A settlement will attempt to build an external building within its security radius if it does not have enough stock and there is no available building of the correct type already within its radius. Empty buildings will be targeting for occupation by a skilled worker and the settlement will attempt to fulfil any quests it has internally before relying on trade alliances.

As an example of the Settlement's construction, quest and trade models, suppose a village is low on food and has no farms of its own, but there is an abandoned, ruined farm within the security radius. The settlement will send a farmer to the building to occupy it. This will generate several quests with the settlement. A builder is needed for repairs; Wood is needed for repairs; and Seeds are needed for planting. The village sends out the wood, but has no seeds or builders. As there is no way to generate seeds the quest is opened for players and allied settlements to provide a solution. The village has a hammer tool and an unskilled worker so is able to create a new builder and sent them out to repair the building. Once repaired, the builder will return to the settlement.



## Bad-Guy Lairs and Raiders

Now that we have completed the basic principles of the dynamic worlds interaction we need to start getting into some game-specific technical details. I will use the Tolkien genre to describe such things as Goblins and Wolves as the Bad-Guys, and Heroes and Elves for the Good-Guys, but be warned that just because Elves are Good-Guys this doesn't mean they won't kill player on site if you approach their city. Just like the woods of Lothlorien, the good-guys might kill strangers on site just for being in their territory. Being a Good-Guy just means a settlement. It does not imply any alignment or alliance. Fortunately that's all I need to say on the Good-Guys as their homes are mostly decided by the Games Master and Player influence. The Bad-Guys are more simplistic.

Bad-Guys are born in Lairs. These are underground or off-map structures whose purpose is to create Bad-Guy raiding parties to wander the surface map and provide Conflicts. The Lair can be a dungeon that the player can enter to loot and pillage as part of their adventures. This will reduce the threat and possible halt the production of the Lair.

Assume the world map is split into a grid with around 10x10 squares per sector. Every sector will have a Lair. The Lair will be themed to suit the world map. It could be goblins or bears in the mountains, bandits or wolves in the forest, and lions or tribesmen amongst the plains. Each lair will have a growth rate, a population cap, a fraction value and a group value. These creatures will live and grow inside their Lair until the population reaches its cap and then they split into groups. The fraction value decides how many leave the Lair, and the group value decides how many groups are created from those that leave.

It is important to remember that once a group has left it's lair they cannot re-enter. No society, Good or Bad, can enter a Lair. Only players can do this if allowed by the game mechanics where Lairs are considered part of the Dungeon framework. Raiding parties will belong to the Lair's society. Since each party belongs to the same society they will avoid attacking each other as they leave the area. If a raiding party captures a building then the building acts like a lair. It will generate extra bandits each game cycle until the population cap is reached and then further raiding parties will be released. This allows a lair to spread its influence via ruined buildings and abandoned player locations, however raiding groups cannot enter other lairs, so the expansion will be constantly threatened by other lairs closer to the new buildings.

The Raider parties will have specific orders depending on their content. As mentioned earlier a raid group that consists of peaceful sheep is not going to move towards a target and attack it, but a bandit group will. Raid groups follow three basic rule:

1. Do you target and move to attack nearby NPC groups from another Society?
2. Do you target and move to attack buildings from another Society?
3. Do you defend conquered buildings or abandon them and continue wandering?

In a more advanced system you can add more rules, such as targeting player groups and giving attack priority to buildings over NPCs. Wandering spiders are unlikely to attack a windmill, but they will attack farm or lumber camp and they will occupy a conquered quarry site.

## Chapter 3: How the World is made and populated

### In the Beginning, there was the map

...and on that map there was terrain, and Conflict Lairs, and ancient sites of interest. Before the start of the game several thousand game cycles need to be processed to simulate the growth of lairs, raiders wandering the map and battles taking place. This will prepare the world for the Good-Guys and Players to come along and be heroes.

Before any of that, there needs to be a map. The traditional method of creating a world map is to build a very large grid. This is the overland map. Once the terrain is in place then Lairs need to be placed by a careful algorithm to ensure they are not clustered too close together in one area of the map while leaving other areas bare. Splitting the map into sectors of 10x10 squares and adding a Lair within that sector allows for a good distribution without making it easy for the players to immediately work out where a lair is likely to be placed. You can also fluctuate the placement of the Lair so that it can appear outside of its allocated 10x10 grid, or skip some sectors completely while having multiple lairs on other sectors, in order to make lairs even harder to predict.

Lairs should be themed based upon the surrounding terrain. Crocodiles should not appear in the frozen mountains and bears should not appear in the desert. To avoid creatures from one terrain climate from occupying an opposing climate the movement of Raiders should be restricted to prevent them entering unnatural terrain. A swamp crocodile will not enter desert or mountain terrain, but it will venture into the forest and occupy an abandoned watermill, where it will then remain protected by the shelter and produce baby crocs on a regular basis.

Beneath this world map is usually an abstract map of individual locations in the dungeon layer. In theory a player should be able to dig a tunnel from one dungeon to another but this is rarely allowed and is prohibited by the game mechanics, as Dungeons are unique map entities that are not connected to anything but a single sector on the world map. Settlements, Dungeons, Buildings and Lairs are assigned to a square on the overland map that takes players to a unique, unconnected layer for that location. No matter how far a town expands, it never leaves that single square on the overland map. A massive city of millions of people takes up the same space as a tiny village of a dozen people. Any dungeons beneath that city can also expand to a massive size, without ever leaving the borders of its overland square and connecting to a neighbouring dungeon.

For non-dynamic worlds this rather bland method allows for highly detailed, graphically customised locations to be built for each unique feature. The idea that a feature is just a sub-region of the bigger square also gives flexibility for a dynamic world and a crafted world. If you want to create an expansive underground realm that connects to the upper world then you can create a second world map for the underground and use Lairs as the connection between them. You can do the same for a “sky world” map, or a magical nether-realm that runs parallel with the physical world. We solve the problem of connecting the abstract dungeon and lair locations by using gateway rooms. This is an exit room to leave the dungeon that doesn’t assume the exit is to the overland square where the dungeon is located. Exit gateways can connect to a room in another dungeon, or to a different square on another world map, such as the underground world or a spiritual plane.

The Dwarven Realms occupy a mountain range on the overland map. There is a settlement on each side of the mountains, with unoccupied lairs beneath them. Access to the dwarven realm from the overland requires players to travel through the settlements and lairs. The lairs are simple two-room gateways between the maps. One room exits to the overland world, and the other room exists to the underground world. The settlement prevents wandering Raiders from occupying the lairs.

Some map elements and lair gateways can be dynamic entities. Some portals into and out of lairs may only work during a celestial event. There may be floating islands or clouds that move around the map, carrying their settlements along with the tides and winds. In a space adventure the planets will revolve around their stars, and the stars revolve around the galactic central point. This could perform the basis for a simulated weather system.

Above the normal world map is a weather layer. Any square above a body of water has a chance to become clouds. Map cells have a counter for every cloud that occupies that square. Wind blows the clouds across the map, with each square being pushed a random distance. As the clouds build up in each square they become heavy cloud, rain cloud and storm clouds. The squares underneath get wet and may suffer damage depending on the density of the storm. Some squares will have a natural affinity to cloud. For example, a desert square has a good chance to decrease the cloud level for every adjacent cell. The opposite can apply to mountain squares that force rain to fall even from light cloud. To simulate dynamic seasons the chance of clouds forming will increase in the winter and decrease in the summer. This rainfall can then affect the terrain design of the map. Areas of drought will dry up the land so that grassland becomes desert, trees and crops will grow slowly unless irrigated from nearby rivers. Too much rain will turn plains into swamp if the water cannot find a nearby river for drainage. Fields will be flooded and the crops may be lost.

During the winter, these rains should become snow. This will make mountain squares impassable and wilderness travel more dangerous. Resting outside at night will not recover much in the way of energy and stamina. Raid groups of wild animals will attack more often, cause by a change in their society's default reactions. In summer a wolf pack would only target groups within 2 squares, but in winter this extends to 4 squares.

## Evolution of the terrain by flood, earthquake and devastation

The game world needs to change as part of the dynamic structure. Some events will occur regularly and with some predictability. Heavy rains will fall in spring and autumn and snow will fall in the winter. Summer will be dry with little chance of rainfall.

Areas around a river will flood after heavy rain and then drain back to grassland when the water recedes. This will destroy any fields planted and damage any farms on that terrain. Heavy snow will block sectors next to mountains and push ice glaciers further along their path. These things can be quickly and easily handled by a game engine by using simple comparison rules based on a square's content and the neighbouring squares.

More difficult and traumatic events should be governed by the Games Master. Invading hordes may set forest fires that burn away great sections of woodland, causing smoke and dust to blow across the land. This dust would be handled in the same way as rainfall, causing minor damage to buildings in its path. Other major events that can radically change a landscape are meteor craters, eroding rivers and land-slides.

Meteor craters are easy to simulate but have dangerous consequences to the maps design. After locating a designated Impact point and Impact size, the meteor strikes that target square and inflicts major damage to the occupants of that square. The impact size then determines the radius from the target point that is affected and all items within this radius are damaged, and the terrain is flattened. A low damage impact would simply knock down the trees and possibly start a forest or bush fire, but a major impact would flatten mountains and hills, block rivers and cause a crater of new hills to form around the edges of the impact radius. The amount of river sectors that are disrupted would cause the water passing into the area to flood across the new crater floor, possibly forming a lake or swamp land.

Over time rivers will gradually change course by erosion, flood damage, land-slide obstructions and engineering. These should be very slow, gradual changes that rarely affect the game world. This is one of the hardest systems to dynamically simulate as the river cannot be analysed as individual squares on the world map and a regional analysis of the river and the surrounding terrain needs to be performed to maintain consistent activity over a very long period of time. The changes to create loops, breaks and orphaned sections of river must not then reverse a change at a later date. Erosion does not go backward 'up' a rivers path. When dealing with splits and joins a river must flow in a linear path without looping back on itself. Islands will form where a river splits and merges but on both channels the river continues to flow 'downward'. The river does not flow down one side, loop around the island and then flow back up the other side and the downward 'exit' should never dry out. Erosion may cause the river to move away on one side of an island which cuts off and eventually drains the channel on the other side so that the island joins with the mainland, but the 'exit' path beyond the island must remain intact, even if this means moving the exit to meet the new channel.

Land-slides are fairly simple to simulate and should occur in the vicinity of mountains, hills and rivers. There are two types on land-side event: Small scale will damage or destroy buildings and fields but will not change the overall terrain of the world map squares. Large slides will deform the terrain and permanently alter the world map, possibly affecting the course of rivers and floods.

## Building Templates, Masters, Apprentices and Workers

Each building will have a template design to designate its population, required supplies and stockpile of produced goods.

Buildings work from one of two methods: Generation or Conversion. A generator building creates new materials from nothing. Farms will produce seeds and libraries will produce research without the need to any input resources. A convertor building will change an item into another item. A barracks will convert a militiaman into a soldier, and a bakery will convert flour into bread. The conversion process is rarely one-to-one and multiple resources are required to upgrade goods. The barracks will consume wooden swords, leather armour and straw dummies in the training process, and the bakery needs flour and water to produce bread.

Each building will have a population of at least 2 people to allow for a Master and an Assistant. In many cases the population will be much higher, such as a vineyard would have a population of 10 to allow for many gatherers. In this case there will be one master vintner, an apprentice, and 8 unskilled workers. At the end of each work cycle the template will decide how Experience Points are divided between the master and the assistants. Any unskilled worker who gains sufficient experience has a chance of learning the trade skills from the master, and will become a skilled worker. When a Quest is generated in the parent settlement or a nearby allied society for a skilled worker then the Vineyard may expel a spare skilled worker in order to complete the quest and recruit a new unskilled worker to replace them. Most buildings will only need one master and one assistant to run at efficient productivity. Additional workers will provide additional goods but at a much lower rate than the master. For example a corn field can produce 100 stacks of corn. A Master Farmer alone will product 75% of the goods. Having an Assistant will provide the additional 25%, and each worker will contribute an additional 5% up to a maximum of 150% capacity.

Inbound goods are the spare tools, wooden planks or stone bricks needed for repairs and the seeds, ores and source materials needed to perform the buildings purpose. When the stockpile falls below the minimum levels defined in the template then a quest is generated to acquire more stock.

Outbound goods are the finished materials that need to be transferred back to the parent settlement or are placed on sale directly. Distant external buildings and work camps that are outside of the parent settlement's security radius will generate a quest for a supply wagon to come and transfer the goods back to the settlement.

## Age, Death, Disease, Corpses, Graveyards and Necromancers.

In a dynamic world people start as babies and take time to grow into children. Children can go to school to gain experience points but these are not spent on skills until they become juveniles. Juveniles can go to college and learn a specific skill from a teacher using their experience points, or they can go into the unskilled worker pool and be put to use in any industry where their unspent point can be used more randomly. Eventually, juveniles will become adults, middle aged and then old people until they die.

People can die younger from attack on the settlement, disease, murder or accident within a settlement. Dead people don't just disappear. They leave corpses behind. In the wilderness these bodies are burnt after battle, but in a building or settlement they need to be removed. Some buildings such as crematoriums and graveyards will dispose of the bodies safely but if too many corpses are left for too long then disease will spread through the settlement and more people will die.

Workers at a crematorium will use fuel to dispose of bodies but this is a fast way to remove the dead. Once the corpse is burnt it is gone and its details removed from the game world. Graveyards have a fixed number of plots in which they can bury the dead and each plot needs time before it can be reused. While cheaper to run than a crematorium the graveyard solution is much slower and runs the risk of Undead being raised by a Necromancer. This may not be a bad thing if the necromancer is friendly towards the society and has the skill to keep his summoned minions under control but there is always the chance that a necromancer will lose control and the settlement will be attacked by undead from inside their defensive walls. For this reason many settlements will only allow graveyards as external buildings, but the lack of an internal graveyard means that necromancers are unable to practice or teach their skills.

Any rogue undead that are unleashed within the city walls are not affected by the settlements defensive structures and cannot be attacked by siege weapons. It takes troops on the ground to battle them in open warfare, and any civilians in the area will be at high risk of attack during the battles.

## Chapter 4: Alternative World Layers

### The Pyramid World

What if each layer of the world was not simply a cloud map, overland and underground? What if the world had a hundred layers to simulate the shape of a pyramid. Traditional farms could only exist on the outer edges of the walls and rain would fall more often on one side of the world than another. The lower levels would be more valuable as they have a bigger area on the outer edge than higher levels. The greatest treasures would exist on the lowest levels, towards the centre of the map – the location furthest away from the safe settlements controlled by the societies.

In this world, landslides caused by mining inside the pyramid would not only affect the terrain of the affected squares on a single layer, but could cause a chain reaction to the layers above and below.

Flooding at a top level could cause a new river to flow through the around layer and then winding down through each layer until a new exit can be found. Rooms/Squares will flood every game cycle as the river seeks to find a new exit.

The society alliance model would be affected by a resource shortage, causing allies to eventually turn on each other in a desperate bid for food and water during the changing seasons. Supply caravans would make up the majority of the population as they seek to move greater amounts of goods from the outlying work camps towards the parent settlements, triggering Conflict alerts every time they pass too close to other society's resources.

The ecological model would also affect the dynamics of the world. With such limited outside resources the internal ecology of the pyramid would be used to grow alternative resources. Mushroom farms would provide the food and fibres needed to replace the traditional farms and cotton mills of the overland world. Fishing would still be possible but nowhere near the scale of an ocean-side settlement. Soaking fibres and dredging dirt from riverbeds to make clay and brick would replace the lumber mills and quarries of overland. The dynamics of the settlement's growth and harvesting will dramatically change between and outside settlement and an inside settlement. The security radius that provides instant quest solutions to external buildings can now be vertical between layers.

The pyramid can also be inverted, with the widest section at the top of the thinnest point at the bottom. Farming the top layer now provides a large number of conflict points as settlements and buildings start to overlap each other's security zones. Battles for control of the entrance points to lower levels would be frequent and allegiances would constantly sway without the players' involvement. It's even possible that player-interaction would stabilise the world dynamics as they resolve quests and conflicts to reduce the amount of changes made by the dynamic engine.

## The Great Underworld

The traditional surface-based world has three layers: the clouds, the terrain and the underground. Why limit the underground to a single layer when it can be far deeper and complex than simply a darker version of the over-ground layer?

In the Great Underworld map there would be multiple layers of underground worlds, each darker and more dangerous than the one above. As mentioned in the pyramid layers the ecology of the underground would be vastly different to the overland due to the lack of farming and lumber that is available. Narrow tunnels and twisting passages would increase the distances for supply wagons to travel to reach their destinations, increasing the storage and stockpiling needs of the deeper settlements. Quests would remain active for longer and the NPC population will expand slower. Bandits in control of vital choke-points in the tunnels that are outside of the security radius would only be cleared by player intervention and the constant theft of supplies passing that area will cause raiding parties to be released more often.

Unique terrain effects such as flowing lava can pass through a map, burning anything in its path and leaving behind rock and gemstone resources for exploitation. The settlement dynamics will make these mineral deposits a priority over food supply as a gem cutter building would be worth a dozen farms in the overland.



## The Boat People

What if the entire world was flooded and the only settlements are giant floating ships that move along with the tides? Towns and Cities built on these ships would constantly travel around the map, moving the “boat” terrain with them. External buildings would eventually fall behind the security radius and become work camps. The dynamics now requires that a settlement deliberately abandon a building once it falls beyond the reach of the parent settlement. The raiders will surely take over quickly and start to turn out their raiding parties to follow the boat-cities movement over the seas.

## The Sky World

In the same style as above except the flooded world is the underground layer and everybody lives in floating castles in the sky. Farms and lumber are placed in the dungeon layer and goods can only be transported to the sky world via Sky-Bridge gateways.

## Chapter 5: Specific Data Structures

### Individuals

An individual's data is stored in three tables. The People table contains the primary data on the individual, the Skills table contain a list of skills known to that individual and the Inventory table contains a list of the tools, armour and weapons that the individual has. Several dynamic factors can be decided based on these tables. If an individual is diseased and they have a cure then they will consume this item. If they are injured and they carry bandaged they will attempt to heal themselves. If they have skill in medicine then they will have a greater chance of improved healing ability.

- PEOPLE: Person\_ID, Person\_Name, XP, Age, Health, Society\_ID, Location
- SKILLS: Person\_ID, Skill\_ID, Skill\_Level
- INVENTORY: Person\_ID, Item\_ID, Quantity

### Buildings

Building data is stored in three tables. In the event that a building can be upgraded then the Template ID is changed to reflect the new building. This assumes that all buildings are available for construction and the only construction materials are Wood and Stone.

- BUILDINGS: Building\_ID, Template\_ID, Settlement\_ID, Society\_ID, Internal/External, Health
- CONTENTS: Building\_ID, Item\_ID, Quantity
- TEMPLATE: Template\_ID, Building\_Name, Required\_Wood, Required\_Stone, Health, Defence\_Bonus, Max\_Masters, Max\_Assistants, Max\_Workers, Master\_XP, Assistant\_XP, Worker\_XP, Upgrade\_Template\_ID

You can expand the Template to make buildings unavailable, or require specific races or technology to complete. You can also remove the Wood/Stone requirements and put the building resources, percentage built and other game-specific details into a related table.

### Settlements

Settlements generally have very little information unique to them as everything is determined by the individuals and buildings that make up the settlement. Even the inventory, guards and specific defences are stored in the Building information tables, with general defences from terrain bonuses calculated real-time to allow for terrain changes to take place in combat situations.

- SETTLEMENTS: Settlement\_ID, Settlement\_Name, Society\_ID, Location, Morale

### Societies

With the society data we record even less. The Alignment determines the default reactions to other societies that offer or accept quests and conflicts that encroach within the security radius.

- SOCIETIES: Society\_ID, Society\_Name, Alignment

Once a new society has been encountered the Alignment will determine the default Alliance that is created to that society.

## Alliances

The Alliance tables are a very simple combination of the relation between a society and its target.

ALLIANCES: Society\_ID, Target\_Society\_ID, Attitude

The attitude ranges from Military Allies, Friendly, Ignore, Repulse to Invade and Destroy. With an invade attitude the society will attempt to capture and hold buildings, with destroy any buildings will be razed to the ground if captured.

## Lairs

The Lair records the current population and the growth rate of its occupants. When the Population cap is reached a number of Raider Groups are formed. The Group\_Population then leaves the lair and divide themselves between the raider groups.

- LAIR: Lair\_ID, Location, Occupant\_Race\_ID, Population, Population\_Cap, Society\_ID
- TEMPLATE: Template\_ID, Occupant\_Race\_ID, Growth\_Rate, Population\_Per\_Room, Raider\_Groups, Group\_Population

The template is used to record the different types of population a lair can hold. For example a Dragon's lair would create new population based upon the Occupant\_Race\_ID of a dragon creature class. This class includes baby, young, juvenile, adult, greater and ancient dragons. Any of these can be spawned into the lair. The population cap of an individual lair will depend upon the number of rooms, or the size, of that particular lair. A single room Dragon's lair will be capped at 2 dragons with Raider\_Groups and Group Population both set to 1. This means that a single dragon will occupy the lair and further dragons will exit as Raiders. In a large complex of 20 rooms there will be up to 20 dragons before raiders are created. If this were a bandit camp then the population\_per\_room would be around 12, with 3 raider\_groups being released comprised of 8 population each. Care must be taken with larger group sizes to avoid emptying a lair completely. If there are not enough rooms to allow for a large population then the raiders should not exceed 50% of the total lair population.

## Raiders

Raiders are the Bad-Guy parties that wander the world map and cause Conflicts events. Each group is made up of individual members, and the members are based on a template. The template governs the range of a creature's statistics, such as maximum health, damage range, attack skill. The ranges in the template are used to calculate the exact figures for the individual member.

- RAID\_GROUP: Raid\_Group\_ID, Society\_ID, Race\_ID, Location, Alignment
- RAID\_MEMBER: Raid\_Group\_ID, Creature\_ID, Health, Damage, Attack, Defence...
- CREATURE\_TEMPLATE: Creature\_ID, Race\_ID, Creature\_Name, Health, Damage, Attack, Defence...

For example, the Creature Template for a Wolf would be: Race\_ID = 99 (Wolves);

- Creature\_ID = 1, Creature\_Name = "Wolf Cub", Health = 5 to 10, Damage = 2 to 5
- Creature\_ID = 2, Creature\_Name = "Wolf Prowler", Health = 8 to 12, Damage = 4 to 10
- Creature\_ID = 3, Creature\_Name = "Wolf Alpha Male", Health = 15 to 20, Damage = 5 to 15

## Chapter 6: Summary and Technical Footnotes

The dynamic world is maintained by constantly applying changes and responses to the game world. Weather affects the terrain and production; Lairs produce Raiders who in turn cause Conflict Events; Alliances are struck to maintain the supply of goods and Time causes Ageing, Death and Disease. Each dynamic event is part of a long chain of action and reaction that tells a hidden story within the game world.

- Old Farmers who are experts in their fields (literally) will eventually die and be replaced by lesser-skilled assistants.
- Food production will decrease causing trade deals to fluctuate.
- More land will be devoted to farming drawing more attention from raiders.
- Raiders will trigger conflict events drawing troops away from settlements.
- Some troops will die in the battle, others will become more experienced and gain promotions.
- Heroes and Generals will be formed from the veterans.
- New recruits will be needed to replace the fallen.
- Iron and Steel will be needed to forge new swords and armour.
- Iron mines need more food to cope with the increase in demand.
- Food... brings us back to the old farmer. Gone.

Running these dynamic events will consume large amounts of calculation power and processing time. The size of your game world, the frequency of changes and the level of dynamic aspects that you implements will require careful balancing of time management on a real-time world, or a period of downtime to perform the end-of-day updates in a turn based world.

In a turn-based world it makes sense to apply all of the dynamic changes at the same time, either at the start or end of a day. This can also be performed as part of the data backup process.

In a real-time world the dynamic updates should not be applied at the same time. On my own test application I have the Lairs growing their population at 3am, Raiders move around the map at 6am, buildings gather resources at 12 noon, the Weather is re-calculated at 6pm and Settlements will resolve internal Quests at 9pm. Backups would happen at midnight. This allows for Conflict Quests to remain open to players from 3am to 9pm before the settlement attempts to send out its own soldiers. These times give players the maximum amount of time during the day to take action against any Conflicts that arise and weather events are reduced in importance.

Depending on the speed of your server and the number of connected players you may wish to implement a flexible approach rather than fixed times. By allowing the Games Master to be able to quickly and easily move the times that dynamic events take place you can avoid instances of high-player activity impacting or being impacted by the calculation of dynamic events. The events don't all have to happen on a one-to-one ratio. You could decide to recalculate the weather every-other day, or have building production twice a day, or manually trigger additional Lair growth should players go on a combined rampage and slaughter everything in sight.