

# Allt om spelet

## Innehåll

<b>The One True Principle: Simplicity</b>	<b>4</b>
I programmering . . . . .	4
The order problem . . . . .	4
Require vs Use . . . . .	5
Sidoeffekter . . . . .	5
Global state . . . . .	5
Att inte använda interfaces/protocols . . . . .	5
Subsystem som callar varandra direkt . . . . .	5
DRY-principen . . . . .	6
Kameran och världen . . . . .	6
I speldesign och annat . . . . .	6
Vim-plugins . . . . .	6
HoN-kartor . . . . .	6
Invisibility . . . . .	7
Armor types . . . . .	7
Mutalisks . . . . .	7
Reducers . . . . .	7
Vim-plugins . . . . .	7
Skor och skosnören . . . . .	7
<b>Protocols</b>	<b>8</b>
<b>RFB-metoden</b>	<b>9</b>

<b>Hammock-Driven Development</b>	<b>9</b>
<b>Genvägar och specialfall</b>	<b>10</b>
<b>Speldesign och spelidéer</b>	<b>10</b>
Gör kärnan rolig . . . . .	10
Strävan och beroende . . . . .	10
Reward-feedback-loop . . . . .	11
Feedback . . . . .	11
Theme Park vs. Sandbox . . . . .	12
I Am The Boss . . . . .	13
Tibia . . . . .	14
Story . . . . .	14
FFA eller lagbaserad PvP . . . . .	14
Världen geografiskt . . . . .	15
PvE . . . . .	15
Items och spelets ekonomi . . . . .	16
Sänkor . . . . .	16
Random stats som i Diablo . . . . .	17
Token Economies . . . . .	17
PvP rewards . . . . .	18
Straff för att dö . . . . .	18
Combat mechanics . . . . .	19
Alla måste ha tillgång till allt . . . . .	20
Instanser (av dungeons) . . . . .	20
Levels och expning . . . . .	20
Alternate Advancement . . . . .	21
Crafting . . . . .	21
Vapen- och spellskills . . . . .	22
Klasser . . . . .	22
Roller – Tank, Healer, Dps . . . . .	22
Specs . . . . .	23

Factions . . . . .	24
Quests . . . . .	24
Auction House . . . . .	24
Random idéer och tankar . . . . .	24
Superitems . . . . .	24
Resistanser . . . . .	24
Några nice saker med Tibia . . . . .	25
<b>Implementation – frågor och lösningar</b>	<b>25</b>
Lista med features ordnade efter viktighet . . . . .	25
Klass 1 . . . . .	25
Klass 2 . . . . .	25
Klass 3 . . . . .	25
Klass 4 . . . . .	25
Klass 5 . . . . .	26
Klass 6 . . . . .	26
Mobs-tabell . . . . .	26
Drops . . . . .	26
Pathfinding . . . . .	26
Sidoeffekter, state, och deltas . . . . .	26
Spells . . . . .	27
TCP vs. UDP, o.s.v. . . . .	27
<b>Referenser</b>	<b>27</b>

## The One True Principle: Simplicity

Simplicity, som definierat av Rich Hickey i [Simple Made Easy](#) (och [Simplicity Matters](#)) är den ultimata designprincipen; den enda ledstjärna man behöver. Den funkar i all typ av systemdesign, t.ex. programmering eller speldesign.

### I programmering

För att hjälpa en med att göra enkla saker bör man tänka på “what, who, how, when, where, why”, och separera dessa aspekter. T.ex. ska inte “what”, som man definierar i ett interface/protocol ha någonting med “how” att göra.

Man kan också testa sin design genom att tänka sig hur det skulle vara att unit testa olika delar, och även hur det skulle vara att skriva olika delar i andra språk, eller köra dem på en annan dator eller i en annan tråd. Kan man inte tänka sig det så är saker complectade med varandra.

Följande är exempel (specialfall) på principen. De är ofta exempel på brott mot principen som vi har begått tidigare, och som vi bör undvika i framtiden.

Jag vill också understryka att man ska sträva efter simplicity på alla nivåer: inom en funktion, mellan funktioner, mellan moduler, mellan subsystem, mellan program, o.s.v.

### The order problem

“The order problem” är när ordningen spelar roll, t.ex. ordningen av statements i ett program eller ordningen av argument till en funktion. Problemet med ordningen av argument till en funktion eller liknande löser man genom att använda maps/keyword arguments i stället för listor/positional arguments. D.v.s. i stället för att göra

```
(defn draw [object color size position] ...)
```

så gör man

```
(defn draw [object {:keys [color size position]}] ...)
```

Då behöver man inte komma ihåg ordningen av argumenten, och det går bättre att lägga till nya argument till funktionen. Man behöver inte alltid använda keyword arguments dock, men man bör tänka efter när man skriver en funktion.

## Require vs Use

När man programmerar Clojure bör man aldrig använda `use`, utan bara `require`. När man använder `use` så complectar man namespaces. Vad är det för fara med att använda `use` när inga namn i namespacesn krockar? Låt oss säga att man har ett namespace A som `use`:ar B. De har inga namn gemensamma. Allt är OK. Men senare så tillkommer saker i A och/eller B som gör att de nu finns krockar. Är det lätt att ändra från `use` till `require`? Nej! Nu måste man ändra alla calls i A till saker i B från `(some-fn ...)` till `(B/some-fn ...)`, vilket är jobbigt. En annan fördel med `require` är att när man läser koden i A, och man ser en funktion som man vill veta var den är definierad, så ser man det direkt om den är namespace qualified.

## Sidoeffekter

Förut skrev vi funktioner som t.ex. som `attack-character` som beräknade skada, uppdaterade state och skickade nätverksmeddelanden, allt på en gång. Funktioner med sidoeffekter complectar effekter med beräkningar och funktioner med state. Det är problematiskt om man ska testa koden, det är svårare att läsa och förstå, det blir rörigare, o.s.v. Dessutom så går det inte lika bra att flytta beräkningen till en annan tråd för att parallellisera, eller göra beräkningen en annan tidpunkt, o.s.v.

För övrigt bör vi tänka mycket på vad vi kan parallellisera, för det är framtiden.

**Fråga:** Vad kan man parallellisera?

## Global state

Global state är dåligt, för alla som rör det globala statet blir complectade med varandra. Dessutom är det svårt att testa funktioner som beror av ett stort globalt state.

## Att inte använda interfaces/protocols

Man bör programmera mot interfaces. T.ex. så är `ISeq` ett interface i Clojure, som gör att man kan calla `first`, `rest`, `next` och `conj` på alla datastrukturer. Om man använder funktioner som fungerar på ett interface kan man byta ut de konkreta typerna och koden fungerar fortfarande. Om man inte programmerar mot interfaces så complectar man Vad med Hur.

## Subsystem som callar varandra direkt

Om A callar B så complectar man dem i tid och rum. A måste veta var B är och B sker alltid direkt efter A. Om man i stället kör message passing i en kö,

så behöver inte A veta om att B finns, eller hur många B:n (konsumenter) som läser från kön, eller när det kommer ske.

En bra sak med köer är att man kan göra en pipeline, där olika steg i pipeline körs med olika trådar. Jag tror t.ex. att pathfinding kommer att bli rätt dyrt, så man kan försöka köra den i en egen tråd.

Kolla [The Value of Values](#).

## DRY-principen

DRY, Don't repeat yourself, är ett specialfall av att complecta saker. Om man skriver samma sak på flera ställen, så har man complectat de ställena med varandra. Så fort man ändrar på ett ställe så behöver man (antagligen) ändra på det andra också för att det ska bli rätt.

## Kameran och världen

En 3D-motor är inte så simple alla gånger. För att t.ex. veta vad man klickar på så behöver man både veta game staten, var saker är, och hur kameran tittar på världen. Scene graphen (där 3D-modeller vistas när de är synliga) är också stateful (såklart). Man säger inte bara "rita detta!" utan man säger "sätt detta i scene graphen!" sedan "flytta detta i scene graphen!". Se vidare [Protocols](#).

## I speldesign och annat

Följande är exempel på simplicity utanför programmering. Vi bör ha simplicity i åtanke när vi designar spelet också.

## Vim-plugins

Vim utan Pathogen eller Vundle complectar plugins genom att stoppa filerna i samma mapp. T.ex. det finns en **syntax**-mapp som alla plugins behöver dela på. Det gör att det kan bli namnkonflikter och det är svårare att ta bort ett plugin, för man behöver gå igenom alla mappar och ta bort bara filer som hör till pluginet man vill ta bort.

## HoN-kartor

S2 har complectat kartor med speltyp och antal spelare. Det går inte att spela vissa kartor på en viss speltyp eller med ett visst antal spelare, fast det finns egentligen ingen bra anledning till begränsningarna. Dåligt!

## Invisibility

När man ger units/classer abilities bör man försöka undvika att complecta för mycket. Ger man t.ex. en class en nuke så gör det inte så mycket för att alla har nukes och det är normalt. Ger man där emot en class invis, som en Rogue i WoW, så måste alla andra classer helt plötsligt ändras för de måste ha någonting för att countra invisen.

## Armor types

Att ha armor types i ett spel är att complecta units med varandra. De som har samma armor typ är complectade med varandra. Om någon unit X ska vara bättre mot Y, men Y och Z har samma armor type, så kommer ju X att automatiskt bli bättre mot Z också.

## Mutalisks

Jag är ingen expert på detta, men jag har hört det sägas att det var svårt att balansera mutaliskt i SC, för det var för all-round. Om de var för bra mot X, och man försökte göra de sämre mot X, så blev de automatisk sämre mot allt annat. Om man i stället har en mer specialiserad unit, t.ex. Dark Templar, så kan man göra den bättre på att snipa saker genom att höja skadan, fast det spelar inte så stor roll vilken skada de har om de är detectade eller mot air units.

## Reducers

Rich Hickeys [reducers](#) är ett bra exempel på hur man kan simplificera saker, och visar också att det inte alltid är så lätt att se att saker är complectade.

## Vim-plugins

Om vim-plugins bindar knappar så complectas de med varandra, för knapparna kan krocka med varandra.

## Skor och skosnören

JohnyTex skulle handla skosnören, men alla skosnören var complectade med skor, så han kunde inte köpa några billigt.

Poängen är att när man tänker på det, så ser man complexity i allt möjligt. Men, bara för att man börjar göra det, betyder det inte att man ser *all* complexity. Man måste ständigt vara på sin vakt mot complexity.

## Protocols

[Protocols](#) i Clojure är som interfaces i Java, fast bättre. Funktioner i ett protocol är polymorfiska på typ. En stor fördel med protocols är att om man har lib A och lib B, som inte är ens egna, och lib A har protocol AP och B har typen BT, så kan man i sin egen kod få BT att implementera AP, genom `extend-protocol`, d.v.s. det är ett öppet system (inte som pattern matching, that's very bad).

Hur ska man använda protocols? Med protocols ska man definiera gränssnitt. Men man ska inte använda dem som interfaces i Java, där Collections har `add`, `count`, `empty`, `addAll`, o.s.v. Man ska fokusera på vad som är fundamentalt och absolut nödvändigt för en abstraktion. Ett protocol för en databas har kanske två funktioner: `load` och `store`, för de är fundamentala. Sen så kanske man vill ha flera load- och storefunktioner som gör lite annorlunda saker, för bekvämlighet, men då bygger man dem med protocols `load` och `store` som grund, *för då behöver man bara implementera de två funktionerna om man byter databas*.

[Datomic](#), Rich Hickeys databas, hade ett protocol `storage`, som hade två funktioner, `get-in` och `store` (eller nått sånt, vet inte exakt hur det var). Men sedan kom de på att de ville ha reverse-lookup, för att kunna hitta t.ex. alla personer som bor i en stad (förutsatt att personer pekar på städer), eller nått liknande. I stället för att lägga till `reverse-get-in` i protoolet så använde de speciella keywords `:_person` i stället för `:person` för att söka bakåt. Protoolet hade då fortfarande bara två funktioner. Det är också *lättare att programmatiskt generera queries om det är datan som styr* och inte vad för funktion man kommer använda. Man kan t.ex. säga någonting i stil med `(get-in [:persons :oskar :city :_person])` för att hitta alla personer som bor i samma stad som Oskar.

Så, tillbaka till 3D-motorer och grafiksystem. Det ska definitivt finnas ett protocol som definierar en abstraktion av grafiksystem, så att inte resten av ens kod bryr sig om vad för grafikmotor man använder, eller ens om det är 2D eller 3D. Vad är de fundamentala funktionerna hos ett grafiksystem? Bara `draw`? Kanske. Om man tänker på 2D-grafik så burkar man ju rita en bild varje frame. Man kan inte göra så mycket mer. Oavsett om det är 2D eller 3D så borde man inte behöva bry sig om scene graphen och sånt, d.v.s. protoolet borde inte definiera `add-to-scene`, `move`, `remove-from-scene`, o.s.v. Om det bara fanns en `draw` så skulle man kunna, i en 3D-motor, göra så att om man callar `(draw a)` och `a` redan finns i scene graphen så flyttar man (om det behövs) bara `a` till den position den har i stället för att lägga till den. Och om man inte callar `(draw a)` nästa frame så tar man bort `a` från scene graphen. Fast då måste man förstås definiera begreppet "frame" i protoolet. Hur är det med att rita partikeleffekter i stället för modeller? Eller sätta igång animationer? Det kanske inte går med bara en ensam `draw` i protoolet, utan man får kompromissa lite. Men man ska ändå sträva efter att ha så få funktioner i ett protocol som möjligt.

Funktioner får (om det ska bli bra) bara ta enkla värden (ren data) eller abstraktioner som indata.



[Tråd](#) på Clojure google group som förklarar lite mer om protocols.

**Fråga:** Vad för protocol/abstraktioner ska finnas, och hur ska de se ut?

## RFB-metoden

RFB = Rätt från början. Carl-Johan på Campanja brukar ibland säga skämtsamt att man ska använda RFB-metoden. Men jag tror det ligger mer i den än så. Det jag tror avses med RFB är att man ska tänka efter innan man gör nått, och inte ta genvägar som kommer att straffa sig längre fram. T.ex. så ska man använda ett bra VCS, d.v.s. git i stället för SVN. Man bör lära sig hur man bäst använder git. Man ska använda senaste versionen av Clojure även om man inte orkar installera det för tillfället. Man ska lära sig vilka funktioner som finns i `clojure.core`. Man bör börja med Issue Tracking och Code Reviews tidigt så att man kan dra större nytta av dessa. Man bör planera, designa, och förenkla innan man börjar koda så att man inte får göra om allt senare. Vi (jag och Posti) gjorde inte det så bra sist.

## Hammock-Driven Development

Rick Hickey säger i [Hammock-Driven Development](#) att man ska inte bara kasta sig in i programmeringen, utan man måste tänka ut en bra lösning först. Test och typsystem hjälper inte om man har missuppfattat problemet eller har tänkt ut en dålig lösning från början. Speciellt så ska man fråga sig om lösningen är simpel. Han säger att man ska gå igenom följande steg, och skriva ner allt man tänker på.

1. Identifiera och formulera problemet.
2. Förstå problemet – kontext, krav, sakförhållanden, restriktioner. Skriv upp vad man vet och vad man inte vet.
3. Utforska lösningar till problemet och till närliggande problem, granska dem *kritiskt*.
4. När du har kommit på en lösning så granska den *kritiskt*.

Han säger även att detta är svårt, och att man måste fokusera och inte distraheras, och att om man gör det så får man idéer när man sover, o.s.v.

## Genvägar och specialfall

När vi gör spelet borde vi ta genvägar, och inte implementera så mycket saker som inte är allmängiltiga. Ett exempel på dumma specialfall är WoWs quests där man hoppar in i någon maskin som funkar på ett unikt sätt, och som har ett unikt användargränssnitt, fast att questen bara varar i 5 minuter – inte bra om man vill spara tid.

Men, vi borde också försöka ta genvägar så mycket som möjligt. T.ex. hur bestämmer man priser som NPCs köper saker för? Det är väldigt viktigt för den ekonomiska balansen i spelet. Man borde hitta på ett system som gör att prissättningen i stort sätt sköts automatiskt. T.ex. så kan man i första versionen av spelet strunta i att ha NPC-vendors. Men man loggar alla transaktioner mellan spelare, och använder de loggarna för att bestämma hur mycket saker är värda, så att man sedan kan göra vendors med vettiga priser. Man kan också tänka sig att man har något system som automatisk anpassar sig efterhand. Om man lyckas med detta så sparar man otroligt mycket tid, antagligen. Se även [The In-game Economics of Ultima Online](#).

En annan genväg man kan ta för att spara tid är att försöka generera världen automatiskt, en slags random world generator. Om man inte blir helt nöjd så kanske man kan ta en hybrid-approach, men man kan nog spara en del tid ändå. Eller man kanske kan göra ett DSL, eller på något sätt beskriva världen i text, och sedan genereras världen utifrån ens specifikationer.

Vi borde försöka identifiera flera problem man kan lösa mer eller mindre automatiskt. Se även [Worse is Better](#).

**Fråga:** Vad för genvägar kan man ta?

## Speldesign och spelidéer

### Gör kärnan rolig

Det borde vara roligt att grinda. EQ är rätt OK även om det bara är grind. Man borde inte lägga till quests, tradeskills, AH, o.s.v. för snabbt, för då finns det risk att man gömmer problemet. I stället så borde man se till att det är kul att grinda. T.ex. så hade levling i WoW varit jättetråkigt utan quests, men de borde fixat det innan de la till quests, så hade resultatet blivit bättre.

### Strävan och beroende

Det som är beroendeframkallande i spel är bra. Folk måste ha någonting att sträva mot, annars är det tråkigt. Exemple på saker att sträva mot:

- Högre level, och level i skills
- Mer pengar
- Bättre items
- Mer items
- Ovanliga items
- Mer makt, vänner, inflytande
- Skill
- Rating

Det finns många saker man kan göra för att utveckla dessa områden. T.ex. kan man ha obegränsad levling, låta folk ha många tradeskills (i WoW kan man bara ha två), **random-kvalitet på items** gör att folk alltid kan hitta lite bättre items, ovanliga items som kanske inte har så mycket praktisk betydelse men som har någon rolig effekt, en leaderboard över PvP, o.s.v.

### Reward-feedback-loop

Man bör göra en "reward-loop" som är beroendeframkallande. Tibia gör det bra. WoW gör det inte bra. Om man går upp i level på WoW, så spelar det typ ingen roll. Man får bara möta bättre mobs också, vilket gör att det känns likadant att spela efter leveln som innan. Det hade varit bättre om man fick nått extra bra på vissa levlar och om det vore som i Tibia att ens egen level inte spelar någon roll när det gäller att få exp. Dessutom är blåa vapen inte så bra, eftersom man levlar ifrån dem, det hade varit bättre om det inte fanns så många vapen att få så att man kan få nya gröna hela tiden. Tibia gör det bättre!

**Fråga:** Hur kan man öka spelets länglevnad på bästa sätt?

### Feedback

En viktig princip i speldesign är att spelaren måste få feedback, d.v.s. veta om det hon gjorde var bra eller dåligt. Några exempel:

- Man vill se vad alla heroes har för hp och mana i Dota. Om man inte skulle sett hp på heroes vore det ju mycket onice.
- Football manager suger för att man inte vet vad som är bra och inte, och det är svårt att få feedback eftersom man burkar ändra på mycket samtidigt, plus att det finns en del tur med i spelet.
- Alla spel som är baserade på skicklighet bygger på att spelaren får feedback hela tiden. Rör man sig fel i Quake så dör man, och så vet man att man inte ska göra så nästa gång – feedback är fundamentalt.

## Theme Park vs. Sandbox

Theme park-MMOs är spel där spelarens väg genom spelet är designad av spelskaparen. Områdets och quests utformning gör att spelaren tar en viss väg genom spelet. Det måste alltid finnas ny content som utmanar spelaren för att det ska vara roligt att spela.

Sandbox-MMOs är spel där spelaren, i större utsträckning än i theme park-MMOs, kan välja fritt vad hon ska göra. Stor fokus läggs också på interaktionen mellan spelare, och det är meningen att spelarna själva ska skapa content och mening i spelet. Ofta har sandbox-MMOs FFA-PvP, byggnader spelare kan bygga eller köpa, mycket crafting och trading, o.s.v. Poängen är att spelet aldrig tar slut, och speldesignerna själva inte behöver anstränga sig så mycket för att hitta på ny content hela tiden.

En sandbox-förespråkare förklarade sandbox-spel såhär:

In sandbox games there are elements that are more or less designed to encourage social content. In most sandbox games they make crafted gear the best gear in the game. So now you have folks that are working hard to make shops, economic empires, massive trade agreements, etc. If you're a merchant you work to try and get agreements with the bigger guilds so you can supply them and make a killing off their war efforts, as in a full loot game people are constantly losing gear.

Unfortunately, now that you have that agreement and you're making the gear for said big guild/alliance you are now a target for their enemies. Their enemies attempt to hunt you down and put you out of business. The guy you have that carries the loot from your shop to their base is being constantly ambushed so now your losing money. You complain to the big alliance and they are forced to bring in a group to protect their asset cause they need that gear just as badly as you need the money.

This leads the alliance after several small skirmishes to go to full scale war with the enemy. They are now trying to kill the other's supply lines while protecting their own and as a merchant you're just trying like hell to keep up.

Det är tydligen (om man får tro en sandbox-spel-skapande person) vanligt att theme park-spel har en användarkurva som ser ut som ett berg, d.v.s. går ner lika brant som den går upp på andra sidan, för att de blir tråkiga när man gjort allt. Sandbox-spel å andra sidan följer inte alls samma typ av kurva. Samma person sa att EVE Onlines kurva bara fortsätter uppåt.

Som jag sa om **genvägar**, så borde man försöka minimera arbetet som krävs för att göra spelet, och om man låter spelarna skapa sin egen “content” så är ju det bra.

Bör man göra ett theme park eller ett sandbox-spel? Det finns fler theme park-spel, och de är lite mer mainstream. Historier från sandbox-spel om personer som blivit backstabbade och bestulna på allt de äger är inte så ovanliga. Jag vet inte hur många potentiella spelare som skulle gilla en sådan miljö. I EVE Online, som är ett sandbox-spel, så kan man inte expa, utan det är mest hur mycket pengar man har som spelar roll. Man slåss alltså inte mot mobs utan mest mot varandra, så vitt jag har förstått. Jag tycker EVE Online verkar jättetråkigt, men man behöver ju inte göra exakt som EVE Online. Jag tror att det bästa är att blanda lite.

## I Am The Boss

Jag har ofta tänkt att om man som spelare kan skapa ett imperium, eller maffia, av nått slag, så vore det nice, och tillräckligt för att spelare inte ska tröttna på spelet. Det är förstås ett problem om det verkligen blir så att det finns en maffia i spelet som gör livet surt för alla andra, men jag tror att man kommer långt om *möjligheten* finns, även om det är osannolikt att någon kommer att bli jättemäktig. Vad behöver man för att det ska gå att skapa maffior och imperier?

- Man måste förlora ganska mycket när man dör, så att fighter och relationer med andra spelare blir meningsfullt.
- Man måste ha PvP, helst FFA-PvP om det ska funka bra.
- Man borde på nått sätt kunna ha kontroll över områden eller resurser.
- Trading bör vara viktigt.
- Det bästa itemsen i spelet måste vara svåra att få/dyra, så att det är någon mening med att skapa ett imperium, och de borde gå att köpa (av andra spelare) annars är det återigen ingen mening med att vara rik.
- Det kanske inte ska finnas ett AH, för då skulle man kunna, i teorin, styra över tradingen mer.

Notera att punkterna ovan inte utesluter att det finns mycket PvE-content, och därmed tilltalar theme park-människor.

Hur man än gör, så är målet att spelet inte ska bli tråkigt så fort man fått det bästa gearet.

**Fråga:** Hur mycket sandbox och hur mycket theme park vill man ha?

**Fråga:** Är detta med imperium en bra idé? Vilka regler ska man ha för att åstadkomma det i så fall?

## Tibia

Tibia har några sandbox-element. Man förlorar mycket när man dör, det finns FFA-PvP, man kan köpa hus, o.s.v. Tibia har ingen max-level heller, bara det gör att det finns alltid nått att göra, om man vill.

## Story

Det borde finnas någon typ av story i spelet. Det finns vissa spelare som gillar sånt. Jag skrev på gamedev om vårt spel och att det skulle vara ett MMORPG som fokuserade på andra saker än story. Alla på gamedev fick en chock.

Men jag tror att det kan vara nice med story för att skapa känsla och atmosfär. Men storyn borde presenteras på ett sånt sätt att man inte behöver ödsla tid på att följa den, i så fall. Och såklart så kan det inte ta för mycket tid från oss att komma på och implementera den.

**Fråga:** Vad för story ska spelet ha?

## FFA eller lagbaserad PvP

Fördelar med FFA-PvP:

- Som jag skrev **ovan** så medför FFA-PvP en högre grad av sandbox-känsla, vilket kan vara bra för spelets länglevnad.
- Man kan döda personer man inte gillar, eller ta ett camp om man vill ha det (det kan man inte om några på ens lag campar där man vill vara).
- Man måste bry sig mer om vad andra spelare tycker om en – ens rykte spelar mer roll.
- Det kan bli obalans i lagen med lagbaserad PvP.

Fördelar med lagbaserad PvP:

- Om man går med i en grupp med folk man inte känner kan man bli backstabbad om det är FFA-PvP, och om man inte har mycket kompisar att spela med kan det bli jobbigt.
- Det blir lättare att hitta och skapa grupper om inte alla är ständigt misstänksamma mot varandra.
- Det kanske är nice att levla upp i områden som ens lag äger, som man kan känna sig lite mer trygg i, och inte dö stup i kvarten.

Lagbaserad PvP behöver inte innebära att man inte kan snacka, tradea, o.s.v. med det andra laget. Man kanske t.o.m. kan tänkas kunna gruppas med dem.

**Fråga:** Lagbaserad PvP eller FFA-PvP? Man kanske i och för sig kan göra sevrar med olika PvP-former.

**Fråga:** Ska det finnas några levelrestriktioner i PvP?

## Världen geografiskt

Det borde vara relativt lätt att göra världen så stor man vill om det är 2d. Men det finns en poäng med att inte göra den för stor, speciellt när det gäller PvP; man vill inte behöva springa runt och leta hur länge som helst för att få en fight.

Det borde finnas dungeons (de ställen med bra loot) nära båda lagen, om det är lagbaserad PvP, så att även om ett lag "ligger under" så har de ställen som är "deras". De ställen med bäst loot borde ligga mitt emellan de båda lagens städer.

Det borde finnas flera nivåer i grottor (som på Tibia) för det ökar äventyrskänslan, på flera sätt. Det är spännande att gå ner långt i en grotta, med farligare monster ju längre ner man kommer. Dessutom blir det svårare att kolla på en karta över världen och lära sig hur den ser ut. Äventyrskänslan försvinner om man vet hur hela världen ser ut.

## PvE

Spelet måste vara roligt och balanserat även utan PvP. Dels för att då kanske man kan sälja spelet även till de som inte vill spela PvP, men också för att det måste finnas nånting att slåss om i PvP, annars blir det bara "deathmatch."

Jag tycker spelet ska vara ganska svårt, som EQ, och inte så hack-n-slash-lätt som WoW. Jag kommer inte ihåg var, men jag läste nånting om att det måste finnas kullar och dalar i "fun graph:en". Om man bara hack-n-slashar så blir det ganska monotont, medan om det är svårare mobs så måste man först söka upp en lämplig mob, kolla runt om det finns adds, sedan fightas en svårare fight, och sedan få mer rewards – mycket roligare. Det är ungefär så det är på EQ, och där är det t.o.m. roligt att bara se en ensam mob, för man tänker "perfekt, en enkel kill!". Om det är roligt att bara se mobs, så är det ju ganska lyckat.

**Fråga:** Hur starka bör mobs vara? Hur mycket fokus ska läggas på grupp-PvE vs. solo-PvE? Om man lägger mer fokus på grupp-PvE kan det göra att spelarrelationer blir viktigare (se [Theme park vs. Sandbox](#)). Å andra sidan så är det skönt att vara en klass som kan soloa i EQ, man har liksom mer kontroll över sitt öde.

## Items och spelets ekonomi

Spelets ekonomi (se [The In-game Economics of Ultima Online](#)) är mycket viktig. Bland det viktigaste är att det inte blir inflation, speciellt inte av items, men inte pengar heller, för då behöver man ständigt justera NPC:ers priser, o.s.v.

### Sänkor

För att det inte ska bli inflation behövs sänkor, både av items och pengar. Pengasänkor är rätt lätta att hitta på:

- Om det finns ett AH så kan det ta en avgift när man säljer något.
- Potions, mat, spell reagents, repairs, enchanting, o.s.v. Det beror på vad som kommer att finnas i spelet.

Itemsänkor:

- Bind-on-equip/pickup.
- Disenchanting
- Begränsat utrymme (man kanske kan hyra mer, d.v.s. en pengasänka), fast spelar vill ha mycket utrymme, speciellt om samlandet av items är en stor del av spelet (se [The Borderlands 2 Gun Discarders Club](#)). Att betala för mer utrymme är då rätt bra, för det är ytterligare en sak att sträva efter som spelare.
- De kan förstöras av användning (inte så populärt antagligen).
- Man kan ge items till NPCs och få buffar, bättre buffar av bättre items.

Alla ens karaktärer borde ha en delad bank, för det är bara jobbigt att logga in alls för att kunna ha mer items.

Jag tycker att man ska ha ganska stora pengasänkor, så att det blir en utmaning att ens ha tillräckligt med repairs, reagents, och potions. Det är roligare att börja spela på en ny server utan att man får massa pengar och saker från kompisar eller andra gubbar man har, och den känslan kan man förlänga med mycket sänkor. Dessutom blir det viktigare att spela bra. Det är också nice på Tibia att det kostar rätt mycket att expa, för man måste ha potions.

Om det är en stor utmaning att ha spell reagents, potions, o.s.v. så kanske man måste kunna gå tillbaka till lättare mobs och spara ett tag, men då måste de ge nått, som på Tibia.

Ett problem med bind-on-equip/pickup är att tradingen lider (se [Theme park vs. Sandbox](#)) av det.

Ett sätt att motverka att man twinklar låglevlade karaktärer är att om de behöver reparera sitt gear, och det är bra gear, så blir det dyrt för dem.



**Pay to improve** Ett problem i WoW är att när man övar tradeskills så produceras massa items som ingen vill ha, fast [tydligen](#) så tycker spelare att de ska kunna sälja dessa förädlade varor till ett högre pris än råvarorna. En lösning på problemet är kanske att man kan öva på tradeskills med råvaror eller pengar, utan att producera items. Man kanske kan gå till en NPC och ta lektioner för guld, eller öva på blacksmithing genom att banka lite på metall utan att göra items. Fast för att det ska funka bra så måste man gå upp i skill mycket snabbare om man gör detta än om man producerar items.

### Random stats som i Diablo

Det finns många fördelar med random stats på items:

- Roligare när man hittar ett item, det blir lite som ett lotteri.
- Enklare för noobs att få OK gear.
- Spelet blir mer långlivat för man kan försöka skaffa bättre och bättre gear.
- Det är nått att [spenderar pengar](#) på.
- Det ger bragging rights att ha bättre gear (se [Token Economies](#)).

Fast när jag säger “som i Diablo” så menar jag det inte fullt ut. I Diablo är itemsen *helt* random, men det är inte så bra. Det borde vara som på WoW och EQ, att de flesta items får man från speciella mobs, t.ex. Executioner's Axe från Executioner, men att de har varierande kvalitet. Då kan man i större utsträckning styra sitt item-letande, och det blir inte total kaos.

### Token Economies

Läs Steve Yegges artiklar [The Borderlands Gun Collector's Club](#) och [The Borderlands 2 Gun Discarders Club](#). Saker han säger:

- Folk gillar att samla på saker, tex vänner på facebook, points på hacker news, exp i Tibia, etc.
- Det ska finnas olika typer av poäng för olika typer av spelare, tex explore-poäng, gear-poäng, pvp-poäng, o.s.v. Man ska gärna få titlar, typ “PvP Champ” eller “Adventurer”.
- Det ska finnas listor eller andra sätt/platser där man kan stoltsera med sina poäng. T.ex. highscorelistor av exp (diablo-laddern) o.s.v.

- Det behöver inte vara poäng, utan andra saker i spelet. Vanligt gear är förstås bra, men man kan göra andra grejer med. T.ex. skins i Tibia. Man skulle t.ex. kunna göra någon “värdelös” spell som ändå är rolig och sällsynt, typ förvandla en annan spelare till nått löjeväckande, fast det bara har en visuell effekt. Eller armor dyes, etc. Man kan göra rare armor dyes som har partikeleffekter. Det skulle vara bra av många anledningar: coolt, unikt, och något som andra lätt ser.
- Silkoraz har nog en poäng med att det är viktigt att se häftigare ut med bättre armor.

## PvP rewards

Ska man få någon typ av poäng när man dödar någon i PvP, som Honor i WoW? Meningen med poängen är givetvis att man ska kunna köpa gear för den. Det är ganska nice att få poäng – man tycker att varje kill betyder mer. Men det finns några nackdelar:

- PvP-gearet kanske gör att man struntar i PvE, och det blir “deathmatch”, det är antagligen inte så kul.
- Om man förlorar poäng av att dö, vilket kan låta rimligt, så kanske alla kommer att zerga.

Jag är lite tveksamt inställd till PvP-poäng. Jag tycker man borde göra PvP mer betydelsefullt på andra sätt, t.ex. genom att man förlorar exp och kanske loot, det blir ett sätt att ta **kontroll över zoner**, o.s.v.

**Fråga:** Ska man ha PvP-poäng för vilka man kan köpa gear? Hur ska man förhindra att det blir “deathmatch” i så fall?

## Straff för att dö

Man måste förlora någonting när man dör, men inte för mycket. Om man inte förlorar någonting på att dö så försvinner äventyrskänslan och även risk-vs-reward beräkningar när man ska gå in i fighter. Hela spelet blir mycket mindre spännande. Å andra sidan så får man inte förlora för mycket, så att man inte vågar spela spelet. Jag tycker att några timmars experience och kanske lite pengar är lagom mycket att förlora.

**Fråga:** Ska det vara olika straff för att dö i PvP jämfört med PvE? Man skulle kunna tänka sig att man förlorar PvP-poäng om man dör i PvP, men det är inte kul att förlora sina PvP-poäng om man blir överkörd av 10 personer. Antagligen är det bäst med samma straff i både PvP och PvE.

Man skulle i och för sig kunna loota items, fast inte gear (d.v.s. saker man kan ekipa). Man skulle kunna loota diverse trade-skill-saker och liknande. Tänk dig följande scenario: Det finns ett ställe som är extra bra för att få tag på spider silk som man behöver för att tillverka en bra robe. De som är i det området har säker samlat på sig spider silk, så då kan man gå dit och PvPa för att få spider silk. Det gör kanske att PvP:n känns mer meningsfull.

**Fråga:** Vad ska man kunna loota i PvP? Items är kanske lite för "random", d.v.s. om man råkar ha ett bra item på sig så förlorar man mycket. Pengar är rätt nice, för då kan man tjäna någonting på att döda någon, och om man är ute och expar så har man ett val: gå tillbaka till banken och garantera att man inte förlorar sina pengar eller stanna och få potentiellt mer exp.

**Fråga:** Ska det finnas "corpse runs"?

## Combat mechanics

I Tibia så är inte fighterna så intressanta, det enda man gör, speciellt om det är två knights som möts, är att stå och slå på varandra samtidigt som man dricker upp sina 100 healing potions man hade i sitt inventory. Om man får lite mer varierade spells så kan det bli mycket intressantare.

I många spel, nästan alla, så spelar positioner en stor roll. I CS eller DoD så vill man hamna i olika positioner beroende på vapen. Den som har sniper vill vara på långt avstånd från fienden och dessutom fightas 1 mot 1. Den som har en machine gun vill vara nära och kan säkert klara av flera fiender samtidigt. Fightingspel, DotA, o.s.v. fungerar alla på liknande sätt. Om man designar klasserna och spellsen så att vissa klasser har en viss miljö i vilken, och ett visst avstånd på vilket, den trivs bäst så kan det nog bli intressant.

I CS och DoD så används granater för att ta sig till rätt position. På samma sätt skulle t.ex. AE-spells kunna skrämman iväg fiender som kommer för nära, o.s.v. Det ska förstås också finnas attacker som är bra på olika saker; t.ex. en meleeklass kanske har ett slag som gör extra mycket skada, medan ett slag sänker fiendens movementspeed, då måste man såklart överväga vad som är bäst för situationen man befinner sig i.

Frågan är hur långt man kan gå i detta spår, främst med tanke på lagg.

Det borde finnas en gräns för hur många spells man kan ha "memmade" samtidigt. Dels för att man inte ska behöva använda så många knappar, och dels för att det kan vara kul att behöva välja. Dessutom kan det antagligen bli ganska nice mindgames när man försöker lista ut vilka spells motståndaren har memmade.

**Fråga:** Hur ska man göra combat intressant?

## Alla måste ha tillgång till allt

Det borde inte finnas vissa klasser som helt saknar interrupts (för att avbryta spells), klasser som helt saknar movementspeed buffar, eller klasser som helt saknar teleports. Det gör bara spelet till sten-sax-påse mellan klasser och är efterblivet. I fallet med interrupts så är det såklart bra om det finns olika typer av interrupts. T.ex. kan wizards ha stuns de kan lägga på avstånd, fast kanske med en hög cooldown. Melees kan ha slag som kan interrupta med mindre cooldown med de måste vara inom räckhåll. Det kanske finns ytterligare klasser som har counterspell som inte bara avbryter spellen men också förhindrar offret från att lägga samma spell igen inom en viss tid.

## Instanser (av dungeons)

En instans är en zon som är avskärmd från resten av världen, och av vilken det kan finnas flera instanser samtidigt. Det är bra för PvE, så att folk inte behöver trängas i de mest intressanta zonerna. För PvP är det dock efterbilvet, eftersom man skulle kunna gömma sig i instanser och skaffa gear utan att vara rädd för andra spelare. Det skulle förstås gå att göra så att på en PvP-server så finns det bara en instans i vilken alla hamnar när det går dit.

## Levels och expning

Det borde inte ta jättelång tid att bli maxlevel, om det finns en, för det är tråkigt att grinda. Levels borde vara jämnt fördelade över speltiden, d.v.s. det borde inte ta 20 gånger längre tid att gå från 40 till 41 än att gå från 10 till 11. Det blir bara tråkigt längre fram, och dessutom blir låga levlar "värdelösa" i den meningen att det är ingen idé att skaffa gear på låga levlar för att man ändå kommer att levla förbi dem så snabbt. Zoner till för de med låg level kommer också att bli underpopulerade, o.s.v.

Man borde börja med mycket kroppsponing och skada jämfört med vad man får när man levlar. En level-2-karaktär ska inte vara dubbelt så bra som en level-1-karaktär, eftersom de förmodligen kommer att mötas och den med högre level ska inte ha ett för stort övertag.

När man går upp i level så ska man få mer hp/mana, skada med spells och vapen, o.s.v., alltså lite bättre på allt, och såklart kunna använda bättre spells och gear.

**Fråga:** Ska det finnas en maxlevel? Nej, dels för att det blir så stor skillnad på att vara maxlevel och att inte vara maxlevel; man levelar tills man blir maxlevel, sen raidar man. Dessutom är det nice att alltid kunna bli bättre.

Man måste kunna delevla om man dör för annars blir det konstigt. Om man precis dingat så kan man pvpa utan att förlora nått. Och det finns inte riktigt någon bra lösning förutom att delevla.

## Alternate Advancement

Att det inte finns någon maxlevel på Tibia är inte bara bra, men det är ändå nice att man hela tiden kan bli bättre. En grej man skulle kunna göra är någon form av “alternate advancement” (AA). Det fanns på EQ, och funkade så att man kunde lägga expen man fick på AAs i stället för vanliga levels. Av AAs kunde man boosta sin karaktär på olika sätt, t.ex. movement speed, stats, regen, resistanser, matkrav, spellskills, o.s.v. Det fanns både klassspecifika och allmänna AAs.

Man kan fråga sig varför man ska ha AAs när vi sagt att det ska finnas en maxlevel. Det finns inte oändligt med AAs, då det finns fortfarande ett max. Dessutom är AAs inte lika signifikanta som vanliga levels. Dessutom är gear en typ av “advancement” som finns utöver vanliga levels, och det är ju någonting vi definitivt ska ha.

Man vill ju, trots att det kanske inte är “rättvist” eller “competitive”, att spelet ska vara beroendeframkallande; det är ju nästan det som är meningen med MMOs. Anledningen till att just MMOs med PvP är så roliga är att de övertag man kan skaffa med gear och levels verkligen kommer att spela roll, eftersom PvP är så tävlingsinriktat, till skillnad från PvE där man kanske kan klara de svåraste bossarna utan att ha alla AAs.

Så, varför ha AAs när man kan ha vanliga levels? En skillnad är att man kan välja vad för AAs man vill ha, vilket kan vara nice. En annan anledning är att de, som sagt, inte är lika signifikanta, och därför kan man göra content som passar för både de som är maxlevel utan AAs och de som är maxlevel med AAs. Dessutom är det kanske kul att kunna välja om man vill ha AAs innan maxlevel, för att hjälpa en levela (alltså att skaffa levels + alla AAs kanske går snabbare om man skaffar AAs först), eller om man vill komma fort till maxlevel för att kunna skaffa mer gear och pengar, t.ex.

## Crafting

Crafting är egentligen kanske nått som man inte behöver. Monster kan lika gärna droppa saker man behöver direkt. Det finns dock några anledningar till att ha crafting:

- Man kanske vill att de bästa itemsen inte ska droppas av monster, utan att delar droppas så att man behöver döda flera för att få itemsen.
- Det är en skill som man kan bli bättre på, och är en ytterligare källa av beroende.
- Det gör antagligen trading lite roligare. Folk med låg level kan kanske skaffa saker som behövs för crafting och tjäna pengar på det, även om gearet de hittar inte är så värdefullt (fast det borde ju vara lagom värdefullt i och för sig).

- Låt oss säga att man har en klass som använder pilbåge, och behöver pilar. Då kanske man vill att han ska kunna få tag på pilar medan han är ute och jagar. Men ska alla monster droppa pilar? Eller i alla fall: Ska det finnas monster i alla områden som droppar pilar? Han kanske kan få tag på någon typ av steg och trä, och kan crafta pilar av det i stället. Och en klass som använder meleevapen kanske kan göra "sharpening stones" av samma material i stället för pilar. (Det kanske inte ska krävas pilar om man har pilbåge, men säg att man kan välja om man vill göra mana- eller hp-potions.)

Crafting borde vara relativt lätt att implementera.

Om crafting finns, så ska det vara instant att crafta något. På WoW tar det tid, och om man har samlat eller köpt mycket material kan man få stå 20 minuter och crafta. Man kan köa upp saker, så det man gör är ju typ att gå AFK – helt efterblivet.

## Vapen- och spellskills

I de flesta spel (Tibia, EQ, WoW från början) finns det vapen- och spellskills, d.v.s. man blir bättre på att använda svärd om man använder svärd. Man kan se det lite som en ytterligare level som ens karaktär har. I Tibia finns det ingen maxlevel på vapenskills. I Tibia så är vapenskills ganska centrala, och man måste öva på dem (genom att stå afk och slåss med en annan spelare). Vapen- och spellskills är för det mesta efterblivet och jobbiga. Det enda som talar för dem är att det känns bra när man går upp i skills, precis som när man går upp i levels.

## Klasser

### Roller – Tank, Healer, Dps

Vad är anledningarna till att ha roller egentligen? Om det inte finns några klasser, och alla har allt, d.v.s. heals, meleeskada, rangedskada, interrupts, disables, AEs, dots, buffs, etc. blir det för mycket att hålla reda på; dels för mycket knappar att trycka på, och dels för mycket möjligheter för moståndaren, så att man inte kan förutse vad han kommer göra. Dessutom om alla har samma saker så blir det mirrors – tänk på SC eller WoW-dueller med samma klass; mirrors är ganska tråkigt.

Man skulle kanske kunna tänka sig att olika personer har olika spelstilar fast de har samma klass, men jag vet inte om det är realistiskt i det här spelet. I fotboll, CS, SC, MtG, o.s.v. finns det lite större möjligheter till att utveckla en egen spelstil, eftersom spelen har fler beslut som ska fattas, längre speltid (i en PvP-drabbning) och mer nyanser än detta spel kommer att ha, antagligen.

Dessutom, om man bara har en klass, så kanske man tröttnar snabbare på spelet. Dels eftersom man bara möter samma klass hela tiden, men också eftersom man inte kan byta själv.

Alla klasser måste kunna göra OK skada, så att en duell mellan två healers inte blir oändlig. Dessutom bör varje klass kunna vinna mot vilken annan som helst.

Heals borde inte vara superviktigt. Man borde kunna ha en grupp utan healers, kanske genom att undvika skada, använda potions, eller bandage, och liknande.

**Fråga:** Vad ska det finnas för klasser? Det är nog bästa att tänka på hur combat ska funka i stort innan man bestämmer vilka klasser som ska finnas.

## Specs

Med specs avses någon typ av uppsättning abilities som man bara kan välja en äkta delmängd av. Man kan använda specs till två saker:

1. Som en fördjupning av klasser, större skillnad mellan olika spelare.
2. Om man kan byta spec, som på WoW, så kan man ha flera spelstilar och roller i en karaktär. Om man känner för att spela healer så kan man specca resto som shaman, fast om man vill dpsa så byter man till elemental.

Jag tycker inte att fall 1 är en bra användning av specs. För det första så går det emot **feedback-principen** – om man inte vet vad man möter (om man inte kan se på nått lätt sätt vad för spec motståndaren har) så vet man inte vad man ska förvänta sig. Fast å andra sidan så funkar det i vissa spel, t.ex. Specromancer eller Magic, där man gissar vad för lek motståndaren har beroende på vad han har lagt. För det andra så finns det inte så mycket mening med specs för detta ändamål. Om en klass har två olika specs som skiljer sig så mycket att det ska vara någon mening med att ha specs så skulle det lika gärna kunna vara två olika klasser.

Fall 2 är mer meningsfullt. Det ger spelaren möjligheten att helt byta roll utan att levla upp en ny karaktär, men:

- Levla är roligt (borde vara, det är ju en stor del av spelet) även om folk ibland säger att de inte tycker det. Och att man behöver levla upp en ny gubbe för att testa något nytt gör antagligen att folk kommer spela mer.
- Om man har en klass med flera olika specs känner man att man borde fixa gear till alla, och det är en jobbig känsla.
- Om det ska finnas flera specs så blir det förmodligen så att flera klasser kommer att överlappa varandra. Det verkar dumt – redundans, usch!

**Fråga:** Ska det finnas specs?

## Factions

Ska det finnas “factions”, olika grupper av npcs, man kan få “reputation” med och på så sätt få tillgång till quests, städer, o.s.v.? Det bör ju finnas någon grundläggande typ av factionsystem för att npcs i de olika lagen ska ogilla de andra laget, och mobs ska kunna vara vänliga eller aggressiva.

## Quests

Ska det finnas quests? Att, som i WoW, ha så mycket quests att man gör dem hela tiden är antagligen orealistiskt, men att ha några quests för att få gear är kanske nice. På EQ finns det t.ex. en uppsättning “noob armor” man kan få för varje klass, och på Tibia finns det t.ex. ett quest för noobs som går ut på att samla döda råttor så man kan få lite cash. Det är definitivt någonting man borde försöka ha, för det är roligt. Quests behöver förstås inte bara vara för noobs.

Man kan ha quests bara för att skicka spelaren till nya områden som är lagom level. T.ex. om man kommer till en ny stad så kan det finnas minst 1 quest för varje huntingområde så att man kan upptäcka dem lättare. I tibia så har man typ ingen aning om vart man kan gå om man inte kollar upp det på webben.

**Fråga:** Hur ska man göra med quests?

## Auction House

**Fråga:** Ska det finnas ett AH? Det finns några nackdelar:

- Kontroll över zoner spelar mindre roll.
- Nästan för lätt att få tag i gear, så det är nästan som att bli twinkad. Tänk på Diablo 3!

## Random idéer och tankar

### Superitems

Det kanske kan finnas bara en instans av det bästa vapnet i spelet samtidigt. Om personen med itemet dör så droppas det. Så kan det finnas ett sånt item för varje gearslot. När man samlat alla blir man en gud?

### Resistanser

**Fråga:** Ska det finnas resistanser? Ska man kunna missa spells? Kanske resistanser tar bort skada och duration fast man inte missar helt?



### Några nice saker med Tibia

- Att man kan välja mer fritt än i andra spel vad för mobs man vill hunta. Man behöver inte ha “rätt” level, utan kan själv välja risk/reward/typ av reward själv.
- Att man behöver använda pots om man ska hunta svåra saker, och de är rätt dyra.
- Att spells är ganska dyra. Iofs inte så dyra efter ett tag, men när man levlar upp.
- Att gear spelar en så stor roll. Med bra gear behöver man färre pots, och sparar pengar, så strävan efter bra gear är viktigare.

## Implementation – frågor och lösningar

### Lista med features ordnade efter viktighet

#### Klass 1

Nätverksfunktionalitet  
Spelare  
Karta  
Mobs  
Melee-attacker

#### Klass 2

HUD/UI  
Pathfinding  
Drops/Items

#### Klass 3

Exp  
Pengar  
Chat

#### Klass 4

Spells  
Friends list  
Map editor Quests

## Klass 5

Factions  
NPC-vendors  
Guilder

## Klass 6

Crafting  
Auktionhouse  
Koda medan spelet körs —————

Vi borde verkligen fixa så att man kan ändra koden medan spelet körs, som [Notch gör här](#). Det borde vara enklare i Clojure eftersom man kan ha en repl uppe medan man kör. Man borde kunna bara göra en nrepl och göra mainloopen i den och sedan connecta till processen med flera klienter.

## Mobs-tabell

För att hålla ordning på alla mobs i spelet så kan man ha en stor tabell där man anger hp, skada, exp de ger, o.s.v. Man borde dock ange dessa värden som en procentsats av vad ett standardmob på samma level har. Notera att om man inte hade procentsatser, utan skrev absoluta värden, så hade man complectat varje mob med svårighetsgraden av medelmob:en på leveln.

## Drops

Hur ska man göra för att enkelt kunna ange vad för drops ett mob ska ha? Man måste kunna ange specifika items, speciellt för boss mobs, men de ska även kunna droppa lite slumpmässiga saker. De borde också kunna tillhöra en grupp – t.ex. alla humanoids på level 1-5 droppar leather armor-delar.

## Pathfinding

Pathfindingen borde kunna hantera mobs av olika storlekar, för att stora mobs är ganska viktigt. Det ger än annan känsla. Man kan inte ha drakar o.s.v. som är lika små som spelare, det håller inte. Pathfinding borde man dock kunna hålla ganska oberoende av resten av spelet, så att man kan göra nått enkelt först.

## Sidoeffekter, state, och deltas

Säg att vi har några olika funktioner i spelet, t.ex. funktioner som hanterar attacker, spells, o.s.v., t.ex. `attack(attacker, victim)`, `cast-spell(caster,`

`target`, `spell`). De ska inte ha sidoeffekter, d.v.s. updatera game state:et, skicka nätverksmeddelanden, o.s.v. De ska i stället vara pure, d.v.s. ta den data de behöver som indata, och returnera något resultat. De borde dock inte returnera *hela* det nya game statet, utan något slags delta. Om de skulle returnera hela det nya game state:t så skulle man inte kunna köra dem parallellt, för hur kombinerar man flera returnvärden (nya states) till ett nytt state? Operationen att “apply:a” ett delta till game state:t borde vara kommutativ, d.v.s. ordningen av delta:sen borde inte spela någon roll. T.ex. betyder det att man inte kan returnera `{:victim id :new-health hp}` för då blir inte appliceringen kommutativ. Man borde i stället returnera `{:victim id :health-change amount}`. Om man använder `new-health` så complectar man deltat med den gamla staten.

## Spells

**Fråga:** Hur ska vi implementera spells?

## TCP vs. UDP, o.s.v.

Ska vi använda TCP eller UDP? Det kanske finns något bibliotek som kan använda UDP men har nån typ av reliability också. Vi borde undersöka detta lite mer. Dessutom, kan man ha flera TCP-strömmar till en klient för oberoende data, är det värt det?

## Referenser

[Simple Made Easy](#)

[Simplicity Matters](#)

[Hammock-Driven Development](#)

[Clojure Reducers](#)

[The Rise of “Worse is Better”](#)

[Clojure Protocols](#)

[Datomic](#)

[The In-game Economics of Ultima Online](#)

[Pathfinder Online-video](#)

[The Borderlands Gun Collector’s Club](#)

[The Borderlands 2 Gun Discarders Club](#)

[Tråd på Clojure google group om protocols](#)

Notch debuggar minecraft medan det körs

The Value of Values