Moobase exodus

# Bakgrunnshistorie

I nær fremtid, i det 21. århundre, har menneskeheten nådd nye høyder innen romutforskning og teknologiske fremskritt. Jordens ressurser begynte å tømmes, og behovet for å finne alternative måter å sikre overlevelsen og fremtiden på planeten ble mer akutt. Et internasjonalt samarbeid ble dannet, kjent som "Global Space Initiative" (GSI), som hadde som mål å etablere en varig tilstedeværelse utenfor jorden.

Som et resultat ble "Project Luna Nova" født. Dette ambisiøse prosjektet hadde som mål å etablere en permanent menneskelig koloni på månen, kjent som "Luna Nova." Målet var å utforske månens ressurser, som heliosfærisk energi og sjeldne mineraler, og bruke dem til å støtte både jordens behov og fremtidige interplanetariske reiser.

Etter år med hardt arbeid og investeringer ble Luna Nova en realitet. Den var utstyrt med avansert teknologi, inkludert selvstyrte hydroponiske gårder for matproduksjon og avanserte solenergisystemer. Forskere, ingeniører og astronauter fra hele verden strømmet til basen for å delta i denne historiske utforskningen.

Forskerene fant mikrober i kraterene som er utsatt for permanent skygge. Disse mikrobene hadde over lang tid blitt utsatt for sterk kosmisk stråling som resultet i at de muterte. De muterte mikrobene kom på avveie og infiserte forskerene som hurtig be mutert til vesen som ikke var gjenkjennelige som mennesker. De som ikke hadde blitt infisert kalte dem bare for romvesen. Disse "romvesenene" fikk spesielle egenskaper og ble en trussel mot resten av kolonien. For å forsøke å stoppe spredningen av infeksjonen, besluttet de gjenværende overlevende å sprenge forskningslaboratoriet, som også inneholdt store mengder av det farlige muterte materialet.

Kommunikasjon med jorden ble kuttet av romvesene, og Månebasen Luna Nova ble isolert fra resten av menneskeheten.

Spillere tar rollen som en av de gjenværende overlevende, en vitenskapsmann eller -kvinne som ikke er infisert. Du er nå fanget på Månebasen Luna Nova, hvor du må finne en vei ut før oksygenet går tomt og livet svinner ut. Samtidig må du løse gåter, samle ledetråder og forsøke å forstå hva som egentlig skjedde på basen. I tillegg må du takle den konstante trusselen fra de uforklarlige romvesenene som en gang var dine kolleger.

Dette er historien bak spillet "Moonbase Exodus", hvor spillerne står overfor en kamp for overlevelse og en utfordrende reise for å avdekke sannheten bak katastrofen på Månebasen Luna Nova.

# Beskrivelse

"Månebase Exodus: Den Store Flukten" er et spennende 2D-web spill som tar spillerne med på en nervepirrende reise gjennom en isolert og farlig månebase. Spillet kombinerer utfordrende oppgaver, gåter og en gripende historie for å gi spillerne en opplevelse de sent vil glemme.

## Spill Historie

Handlingen utspiller seg i en ikke altfor fjern fremtid, hvor romutforskning har tatt et viktig steg fremover, og mennesker har etablert en base på månen for å utforske dens mysterier. Du spiller rollen som en vitenskapsmann eller -kvinne som plutselig befinner deg i en livstruende situasjon. En ukjent katastrofe har rammet månebasen, og du er fanget der uten mulighet for kommunikasjon med jorden. Nå må du finne en vei ut av basen før oksygenet går tomt og livet svinner hen. Men du er ikke alene - mystiske romvesen har også blitt avdekket i basen, og de utgjør en ekstra trussel for din overlevelse.

## Spill funksjoner

1. **Utforskning**: Spillet lar deg utforske de ulike områdene av månebasen, som laboratorier, kontorer, korridorer og andre fascinerende steder. Hvert område skjuler ledetråder, ressurser og utfordringer som hjelper deg med å komme nærmere en løsning. Du kan også finne ting som du kan bruke til forsvar.
2. **Gåter og Oppgaver**: Du må løse en rekke gåter og oppgaver for å komme videre i spillet. Dette kan inkludere hacking av dataterminaler, reparasjon av livsviktige systemer og avdekking av skjulte passasjer.
3. **Tidspress**: Oksygenetivået og livsforrådene dine er begrenset, noe som gir deg et kontinuerlig tidspress. Du må ta raske beslutninger for å overleve og komme deg ut i live.
4. **Romvesenkonfrontasjon**: De mystiske romvesenene er en trussel som stadig overrasker deg med sin tilstedeværelse og handlinger. Du må lære å overleve og samtidig takle den ukjente fare de representerer.
5. **Intriger og Mysterier**: Underveis vil du oppdage spor om hva som egentlig skjedde på månebasen, og du vil bli trukket inn i en dyptgående historie som avslører skjulte hemmeligheter og intriger.
6. **Flere avslutninger**: Spillet har flere slutter, avhengig av valgene du tar underveis. Dine beslutninger vil påvirke utfallet av historien og din skjebne på månen.

"Månebase Exodus: Den Store Flukten" gir deg en spennende opplevelse som tester dine problemløsningsferdigheter, intelligens og evne til å håndtere stress. Kan du finne veien ut av månebasen og avsløre dens mørke hemmeligheter, eller vil du bli fanget for alltid på månen? Spillet gir deg muligheten til å finne ut av det!

# Logger

Noen utdrag fra forskernes logger på Månebasen Luna Nova:

## Loggoppføring 1 - Dr. Elena Rodriguez: 24. september, 2099

Dette er Dr. Elena Rodriguez, lederen for forskningsavdelingen på Månebasen Luna Nova. Vi har nettopp oppdaget noe bemerkelsesverdig i en av kraterene som er utsatt for permanent skygge. Mikroorganismer vi har funnet der, eksponert for intens kosmisk stråling over lang tid, ser ut til å ha mutert på en måte vi aldri har sett før. Dette kan være et gjennombrudd i vår forståelse av ekstreme tilpasninger til rommiljøet.

## Loggoppføring 2 - Dr. Thomas Morgan: 10. oktober, 2099

Forskningen på de muterte mikroorganismene fortsetter å fascinere oss. De har utviklet overraskende overlevelsesstrategier i dette tilsynelatende fiendtlige miljøet. Vi er imidlertid forsiktige og sørger for streng sikkerhet, da vi ikke fullt ut forstår potensielle risikoer.

## Loggoppføring 3 - Dr. Sarah Williams: 15. november, 2099

En av våre kolleger, Dr. Markov, har blitt syk. Han har fått utslett og oppfører seg merkelig.

## Loggoppføring 3 - Dr. Sarah Williams: 18. november, 2099

Dette er mer alvorlig enn vi trodde. Det viser seg at Dr. Markov har blitt infisert av mikroorganismene vi fant i kraterene. Det har forvandlet ham til noe... Uhyggelig, han er mer et romvesen enn menneske. Vi har satt hele området i karantene og forsøker å forstå hva som skjedde. Dette er skremmende.

## Loggoppføring 4 - Dr. Elena Rodriguez: 2. desember, 2099

Situasjonen har eskalert. Flere forskere har blitt angrepet og infisert. De har ikke noe likhet med mennesker og vi kaller dem bare for romvesen. De er meget intelligente og de har sabotert kommunikasjon slik at vi har mistet all kontakt med jorden, og vi kan ikke varsle om det som skjer her. Forskningen på muterte mikroorganismer må stoppes. Vi har dessverre ingen annen mulighet enn å sprenge laboratoriet for å forsøke å stanse spredningen.

## Loggoppføring 5 - Dr. Laura Turner: 15. januar, 2100

Vi er nå i full krise. Månebasen Luna Nova er isolert fra omverdenen. Romvesenene, de som en gang var våre kolleger, streifer rundt på basen. Vårt eneste mål er å overleve og finne en måte å komme oss bort fra månen. Vi må finne ut hvordan dette skjedde og om det finnes en kur for de infiserte.

## Loggoppføring 6 - Dr. Michael Chang: 3. februar, 2100

Vi har begynt å merke at romvesenene ikke bare er aggressive, men også synes å ha utviklet en form for koordinering. De samarbeider for å angripe og isolere oss. Vi har måttet bygge improviserte forsvarsbarrikader for å beskytte oss selv.

## Loggoppføring 7 - Dr. Sofia Ramirez: 12. mars, 2100

Matforsyningene er i ferd med å ta slutt, og vi har ennå ikke funnet en måte å produsere mat lokalt. Hydroponiske gårder har vært en nøkkelkomponent i vårt bærekraftige system her, men de har blitt ødelagt av romvesenene. Vi må finne en annen kilde til næring snart.

## Loggoppføring 8 - Dr. John Walker: 5. april, 2100

Vi har gjort fremskritt i å forstå de muterte mikroorganismene og hvordan de infiserer vertene. Det ser ut til at de kan forvandle og kontrollere sine verters atferd. Vi må finne en kur eller motgift før det er for sent. De er også unike fordi de trenger ikke oksygen og dermed kan bevege seg på overflaten uten romdrakt.

## Loggoppføring 9 - Dr. Elena Rodriguez: 20. mai, 2100

Noen av oss har klart å aktivere en kommunikator fra romfartøyet vårt. Dessverre er signalet svakt og ustabilt. Vi har sendt nødmeldinger, men vi vet ikke om de har nådd jorden. Det virker som at vi bare kan motta informasjon fra jorden, men ikke sende. Vi må holde håpet oppe og fortsette å søke etter en vei ut.

## Loggoppføring 10 - Dr. Mark Davis: 8. juni, 2100

Vi har funnet en nøkkelledetråd. Under mineundersøkelser i de nedre nivåene av basen, har jeg oppdaget en skjult forskningsenhet som kan inneholde informasjon om den opprinnelige hendelsen. Det er en risikabel ekspedisjon, men vi har ingen andre alternativer.

# Mekanismer for Moonbase Exodus

## Kjernemekanismer

### 🎯 1. Oksygendrening (Tidspress)

#### 🎯 Formål

Skape et konstant tidspress som motiverer spilleren til å handle raskt og strategisk. Mekanismen tilfører spenning og intensitet til spillet ved å simulere at spillerens oksygennivå reduseres kontinuerlig, noe som kan føre til spillerens død.

#### 🔧 Beskrivelse

* Spillerens oksygennivå (player.oxygen) starter på 100% og synker gradvis over tid.
* Hvis oksygennivået når 0%, dør spilleren, og spillet avsluttes.
* Oksygennedgangen kan midlertidig bremses eller reverseres ved å bruke spesielle gjenstander (f.eks. oksygentanker).
* Hastigheten på dreneringen kan justeres for å balansere vanskelighetsgrad.

#### 🔑 Funksjonseksempel (TypeScript)

function startOxygenDrain(ratePerSecond: number) {

const interval = setInterval(() => {

if (!player.isAlive) {

clearInterval(interval);

return;

}

player.oxygen -= ratePerSecond;

console.log(`Oxygen level: ${player.oxygen.toFixed(1)}%`);

if (player.oxygen <= 0) {

player.oxygen = 0;

player.isAlive = false;

console.log("You ran out of oxygen. Game over.");

clearInterval(interval);

}

}, 1000);

}

#### 🛠️ Parametre

ratePerSecond: Mengden oksygen som skal fjernes fra spilleren per sekund (f.eks. 1.2).

#### 💡 Effekt

* Kalles automatisk (f.eks. ved spillstart) og reduserer spillerens oksygennivå kontinuerlig.
* Hvis oksygennivået når 0%, settes spillerens status til død (player.isAlive = false).
* Spilleren får kontinuerlig tilbakemelding om oksygennivået via konsoll eller UI.

#### 📌 Bruksområder

* Skaper en følelse av konstant trussel og tidspress.
* Tvinger spilleren til å handle effektivt og prioritere ressursbruk.
* Kan brukes som en justerbar mekanisme for å balansere spillets vanskelighetsgrad.

#### 📎 Tilleggsnotat

* Mekanismen kan kombineres med visuelle eller auditive varsler (f.eks. alarm eller blinkende lys) når oksygennivået er lavt, for å forsterke stressfølelsen.
* Mekanismen kan utvides til å påvirke andre faktorer som bevegelseshastighet eller synsfelt.

### 🧭 2. Utforskning og Navigasjon

#### 🎯 Formål

Gi spilleren muligheten til å bevege seg rundt i månebasen, oppdage nye områder, finne ressurser og gradvis avdekke spillets historie og hemmeligheter. Utforskning og navigasjon driver fremdrift og engasjement.

#### 🔧 Beskrivelse

* Spillerens posisjon spores via player.location.
* Rommene representeres som noder i et romkart (rooms).
* Hvert rom kan inneholde objekter eller ressurser som spilleren kan plukke opp.
* Navigasjon skjer ved å endre player.location til et nytt romnavn.

#### 🔑 Funksjonseksempel (TypeScript)

const rooms: Record<string, { items: string[] }> = {

lab\_entrance: { items: ["keycard"] },

control\_room: { items: ["oxygen\_canister"] },

};

function exploreRoom(roomName: string) {

if (!rooms[roomName]) {

console.log("You can't go there.");

return;

}

player.location = roomName;

console.log(`You entered ${roomName}.`);

const itemsFound = rooms[roomName].items;

if (itemsFound.length > 0) {

console.log("You found:", itemsFound.join(", "));

player.inventory.push(...itemsFound);

rooms[roomName].items = []; // Fjern objektene fra rommet etter plukking

} else {

console.log("Nothing of interest here.");

}

}

#### 🛠️ Parametre

* **roomName**: Navnet på rommet spilleren ønsker å gå til.

#### 💡 Effekt

* Oppdaterer spillerens posisjon til et nytt rom.
* Henter eventuelle objekter fra rommet og legger dem til spillerens inventar.
* Tømmer rommet for objekter slik at det ikke kan plukkes opp på nytt.
* Gir spiller feedback om funnene i rommet.

#### 📌 Bruksområder

* Fremdrift i spillet: Spilleren må utforske rom for å finne nøkler, oksygen, ledetråder og loggfiler.
* Skaper narrativ dybde: Spilleren får innblikk i månebasens historie gjennom logger og miljødetaljer.
* Tilfører dynamikk ved at rommene kan være trygge eller farlige (f.eks. romvesenene kan dukke opp i visse rom).

#### 📎 Tilleggsnotat

* Kan utvides med:
  + Låste dører (krever nøkkelkort).
  + Fareindikatorer (f.eks. rom med infiserte).
  + Visuelt kart eller minikart.
* Romdata kan lett utvides med f.eks. fiende-tilstedeværelse, miljøeffekter eller gåter.

### 🎒 3. Inventar og Ressurshåndtering

#### 🎯 Formål

Gi spilleren mulighet til å samle, lagre og bruke viktige ressurser som er nødvendige for overlevelse og progresjon i spillet. Mekanismen motiverer utforskning, planlegging og strategisk ressursbruk.

#### 🔧 Beskrivelse

* Spilleren har et **inventory** (en liste over objekter) som lagrer alle ressurser spilleren finner i spillet.
* Objektene kan brukes til:
  + Å fylle på oksygen eller helse (f.eks. oxygen\_canister).
  + Å låse opp dører eller terminaler (f.eks. keycard).
  + Å løse spesifikke oppgaver eller gåter.
* Når en ressurs brukes, fjernes den fra inventaret.

#### 🔑 Funksjonseksempel (TypeScript)

function useItem(itemName: string) {

const index = player.inventory.indexOf(itemName);

if (index === -1) {

console.log(`You don't have a ${itemName}.`);

return;

}

switch (itemName) {

case "oxygen\_canister":

player.oxygen = Math.min(player.oxygen + 30, 100);

console.log("You used an oxygen canister. Oxygen boosted!");

break;

case "keycard":

console.log("You used a keycard. A door unlocks somewhere.");

break;

default:

console.log(`Nothing happened when you used ${itemName}.`);

}

player.inventory.splice(index, 1); // Fjern brukt objekt

}

#### 🛠️ Parametre

* **itemName**: Navnet på objektet spilleren ønsker å bruke.

#### 💡 Effekt

* Hvis objektet finnes i spillerens inventory, utfører funksjonen en bestemt handling (f.eks. oksygenpåfyll, åpne dør).
* Fjerner objektet fra inventaret etter bruk.
* Gir tilbakemelding til spilleren om hva som skjedde.

#### 📌 Bruksområder

* Oksygentanker: Påfyll oksygennivå.
* Nøkkelkort: Låse opp dører eller sikkerhetssystemer.
* Spesialgjenstander: Aktivere historieprogresjon eller løse spesifikke gåter.
* Potensielt: Våpen eller verktøy for å håndtere romvesener (hvis aktuelt).

#### 📎 Tilleggsnotat

* Inventarsystemet kan utvides med:
  + Vektbegrensning (for å øke strategisk valg av gjenstander).
  + Kategorisering av gjenstander (nøkler, medisiner, verktøy).
  + Visualisering i UI (ikonvisning, antall, info).
* Kombinasjon med andre mekanismer som utforskning, oksygendrening og gåteløsning gir en rik og dynamisk spillopplevelse.

### 4. Utforskning av rom

#### 🎯 Formål

Gi spilleren en strukturert måte å navigere i månebasen på, slik at de kan oppdage nye områder, samle ressurser og gradvis forstå historien. Mekanismen legger til rette for progresjon og skaper et driv for utforskning.

#### 🔧 Beskrivelse

* Månebasen er delt opp i rom, som kan være laboratorier, korridorer eller skjulte områder.
* Hvert rom er representert med et navn (f.eks. "lab\_entrance") og har en liste over objekter som kan plukkes opp.
* Spilleren kan reise fra et rom til et annet ved å velge romnavn.
* Rom kan inneholde spor, loggfiler, ressurser og farer (romvesen).
* Utforskning gir tilgang til informasjon som trengs for å forstå hva som har skjedd og hvordan man kan komme seg videre.

#### 🔑 Funksjonseksempel (TypeScript)

const rooms: Record<string, { items: string[] }> = {

lab\_entrance: { items: ["keycard"] },

control\_room: { items: ["oxygen\_canister"] },

};

function exploreRoom(roomName: string) {

if (!rooms[roomName]) {

console.log("You can't go there.");

return;

}

player.location = roomName;

console.log(`You entered ${roomName}.`);

const itemsFound = rooms[roomName].items;

if (itemsFound.length > 0) {

console.log("You found:", itemsFound.join(", "));

player.inventory.push(...itemsFound);

rooms[roomName].items = []; // Fjern objektene etter at de er plukket opp

} else {

console.log("Nothing of interest here.");

}

}

#### 🛠️ Parametre

* **roomName**: Navnet på rommet spilleren ønsker å utforske.

#### 💡 Effekt

* Oppdaterer spillerens posisjon til det valgte rommet.
* Henter ut og viser eventuelle objekter spilleren finner i rommet, og legger dem til inventaret.
* Tømmer rommet for objekter, slik at de ikke kan plukkes opp igjen.
* Gir spilleren tilbakemelding om rommet og hva som er funnet.

#### 📌 Bruksområder

* Driver spillets progresjon ved å la spilleren finne ressurser og spor.
* Skaper narrativ dybde gjennom logger og miljødetaljer.
* Kan kombineres med andre mekanismer som låste dører, gåter og fiender for å øke spenning og kompleksitet.

#### 📎 Tilleggsnotat

* Kan utvides med:
  + Tilgangskrav (f.eks. keycard nødvendig).
  + Fareindikatorer (romvesen tilstede).
  + Miljøeffekter (f.eks. oksygennivå lavt).
  + Utforskingsbelønninger (skjulte logs eller spesielle objekter).

### 5. Unngå fiender

#### 🎯 Formål

Gi spilleren en følelse av konstant fare og spenning ved å la dem måtte unngå, snike seg forbi eller rømme fra fiender (romvesener). Mekanismen skaper variasjon i gameplay og legger til en overlevelsesdimensjon.

#### 🔧 Beskrivelse

* Fiender beveger seg mellom rom eller patruljerer bestemte områder.
* Hvis spilleren kommer inn i et rom med en fiende, kan fienden angripe spilleren (og potensielt føre til oksygentap, skade eller umiddelbar død).
* Spilleren må lære å observere, lytte etter fiender eller bruke kartinformasjon for å planlegge ruten.
* Snike-mekanikk kan implementeres ved å la spilleren utføre handlinger som å dempe lyder eller distrahere fienden.

#### 🔑 Funksjonseksempel (TypeScript)

const enemies: Record<string, boolean> = {

lab\_entrance: false,

control\_room: true, // fiende tilstede

};

function checkForEnemy(roomName: string) {

if (enemies[roomName]) {

console.log("A mutated alien is here! Danger!");

handleEnemyEncounter();

} else {

console.log("The room is clear.");

}

}

function handleEnemyEncounter() {

// Enkel løsning: spilleren mister oksygen og blir jaget

player.oxygen -= 20;

console.log("You are attacked! Oxygen decreased.");

if (player.oxygen <= 0) {

player.oxygen = 0;

player.isAlive = false;

console.log("You ran out of oxygen and died. Game over.");

}

}

#### 🛠️ Parametre

* **roomName**: Navnet på rommet spilleren ønsker å utforske.

#### 💡 Effekt

* Hvis fiende er tilstede:
  + Spilleren får et varsel om at fienden er der.
  + Spillet kan automatisk initiere en fiendehendelse (f.eks. oksygentap eller snikevalg).
* Hvis ingen fiende:
  + Spilleren kan utforske rommet trygt.

#### 📌 Bruksområder

* Skaper uforutsigbarhet: spilleren må være forsiktig med valg av rute og timing.
* Bygger opp spenning og stress (kombineres med oksygendrening for å øke presset).
* Kan utvides med:
  + Snikevalg (f.eks. trykke en knapp for å gjemme seg).
  + Distraksjonsmekanismer (f.eks. lage lyd et annet sted).
  + Kampmekanismer (f.eks. improviserte våpen).

#### 📎 Tilleggsnotat

* Kan integreres med et AI-system som patruljerer rom eller reagerer på lyd/hendelser.
* Kan ha visuelle eller auditive indikasjoner (f.eks. skygger, lyder) for å varsle spilleren om fare.
* Viktig å balansere slik at mekanismen skaper spenning uten å bli frustrerende.

### 💀 6. Spillerens Tilstand

#### 🎯 Formål

Sporer spillerens vitale parametere og status i spillet, slik at andre mekanismer kan bruke og endre disse tilstandene. Dette gjør det mulig å administrere spillprogresjon, ressursbruk, overlevelse og sluttresultat.

#### 🔧 Beskrivelse

* Spillerens tilstand lagres i en datastruktur som inkluderer:
  + Oksygennivå (prosent).
  + Om spilleren er i live eller død.
  + Nåværende lokasjon.
  + Inventar (liste over objekter spilleren har).
* Spillerens tilstand oppdateres kontinuerlig underveis i spillet (f.eks. ved oksygendrening, bruk av objekter, utforskning).
* Tilstanden påvirker alle andre mekanismer som oksygendrening, inventarbruk, fiendemøter og spillprogresjon.

#### 🔑 Funksjonseksempel (TypeScript)

type Player = {

oxygen: number;

isAlive: boolean;

location: string;

inventory: string[];

};

const player: Player = {

oxygen: 100, // starter på 100%

isAlive: true,

location: "lab\_entrance",

inventory: [],

};

#### 🛠️ Parametre

* **oxygen**: Spillerens oksygennivå (0–100%).
* **isAlive**: Boolean som viser om spilleren lever.
* **location**: Navnet på rommet spilleren befinner seg i.
* **inventory**: Liste over objekter spilleren har samlet.

#### 💡 Effekt

* Oksygennivået påvirker spillerens overlevelse (0% = spilleren dør).
* isAlive styrer om spilleren fortsatt kan gjøre handlinger (false = spill over).
* location brukes for navigasjon, utforskning og fiendemøter.
* inventory gjør det mulig å bruke og administrere ressurser.

#### 📌 Bruksområder

* Styrer spillerens overlevelse gjennom mekanismer som oksygendrening og fiendeangrep.
* Gjør det mulig å administrere ressursbruk (bruk av oksygentanker, nøkler).
* Brukes av navigasjons- og utforskingsmekanismer for å vite hvor spilleren befinner seg.

#### 📎 Tilleggsnotat

* Kan utvides med:
  + Helse eller skader hvis spillet skal ha kampmekanikk.
  + Moralske valg eller poengsystem som påvirker historien.
  + Statuser som "infisert" eller "stresset" for å legge til flere dimensjoner.

### 📘 7. Gåteløsning

#### 🎯 Formål

Gi spilleren utfordringer som må løses for å komme videre i spillet, få tilgang til nye områder, eller hente viktig informasjon (f.eks. loggfiler). Gåteløsningen skal også bidra til spillopplevelsen ved å skape variasjon, mestringsfølelse og dybde.

#### 🔧 Beskrivelse

* Gåtene er representert som interaktive objekter eller terminaler som spilleren kan aktivere.
* Når en gåte løses, kan den:
  + Åpne dører eller snarveier.
  + Låse opp ny informasjon (logger).
  + Gi tilgang til nytt utstyr eller ressurser.
* Gåtene kan være alt fra enkle kodeknekkinger til mer avanserte sekvenser eller minispill.

#### 🔑 Typiske funksjoner

* **Start puzzle**: Spilleren aktiverer en gåte.
* **Løsning sjekk**: Spillet sjekker om spilleren har levert riktig svar eller løst gåten korrekt.
* **Belønning**: Låser opp nytt område, utstyr eller historieprogresjon.

#### 📑 Funksjonseksempel (TypeScript)

type Puzzle = {

id: string;

question: string;

answer: string;

solved: boolean;

onSolved: () => void;

};

const puzzles: Puzzle[] = [

{

id: "door\_terminal\_1",

question: "Enter the correct override code:",

answer: "MOON42",

solved: false,

onSolved: () => {

console.log("Door unlocked. You may proceed.");

// Her kunne du endret tilstanden i spillet (f.eks. dør åpnet)

},

},

];

function attemptPuzzle(puzzleId: string, playerAnswer: string) {

const puzzle = puzzles.find(p => p.id === puzzleId);

if (!puzzle) {

console.log("Puzzle not found.");

return;

}

if (puzzle.solved) {

console.log("Puzzle already solved.");

return;

}

if (puzzle.answer === playerAnswer) {

puzzle.solved = true;

puzzle.onSolved();

} else {

console.log("Incorrect answer. Try again.");

}

}

#### 🛠️ Parametre

* **puzzleId**: Identifiserer hvilken gåte spilleren prøver å løse.
* **playerAnswer**: Spillerens svar.

#### 💡 Effekt

* Hvis svaret er korrekt, markeres gåten som løst, og onSolved() kalles (kan åpne dør, gi gjenstand, osv.).
* Hvis svaret er feil, får spilleren tilbakemelding.

#### 📌 Bruksområder

* Åpne sikkerhetsdører.
* Aktivere systemer som lys, oksygen eller dataterminaler.
* Få tilgang til logs eller ny informasjon.
* Skape progresjon og motivere spilleren til å utforske mer.

## 2. Andre nivå (Samspillsmekanismer / Composite Mechanics)

Disse oppstår når flere grunnmekanismer jobber sammen og skaper ny dynamikk.

**Eksempler:**

* **Overlevelsesstrategi:** Spilleren må kombinere utforskning + ressursbruk + sniking for å overleve.
* **Miljønavigasjon:** Bruke kart, løse gåter og finne gjenstander for å komme videre.
* **Taktisk fremdrift:** Tid, fiendebevegelser og tilgjengelige ressurser må balanseres.

### 🎯 Formål

Beskrive hvordan de grunnleggende mekanismene i spillet samspiller for å skape mer komplekse og engasjerende spillopplevelser. Disse mekanismene gir dybde og strategiske valg som holder spilleren engasjert.

### 🔧 Beskrivelse

* Samspillsmekanismer oppstår når to eller flere grunnmekanismer kobles sammen og skaper en *dynamisk helhet*.
* De er ikke alltid direkte synlige i koden, men representerer hvordan spilleren opplever og håndterer mekanismene i praksis.
* I *Moonbase Exodus* oppstår slike mekanismer naturlig når spilleren må håndtere oksygendrening, ressursbruk, fiendeunngåelse, utforskning og gåteløsning samtidig.

### 📑 Eksempler på Samspillsmekanismer i Moonbase Exodus

#### 🔹 Overlevelsesstrategi

**Beskrivelse:**  
Spilleren må hele tiden balansere oksygendrening, ressursbruk og farer fra romvesener.

* Kombinerer:
  + **Oksygendrening** (tidspress)
  + **Inventar og Ressurshåndtering** (fylle på oksygen)
  + **Unngå fiender** (unngå skade eller død)

**Effekt:**  
Spilleren presses til å planlegge handlinger nøye, vurdere risiko og prioritere utforskning av rom.

#### 🔹 Taktisk fremdrift

**Beskrivelse:**  
Spilleren må velge hvilke rom som skal utforskes basert på ressursbehov, tilgjengelige gjenstander og fiendens plassering.

* Kombinerer:
  + **Utforskning av rom**
  + **Unngå fiender**
  + **Gåteløsning** (for å åpne dører eller skaffe informasjon)

**Effekt:**  
Spilleren opplever en taktisk dimensjon der valg av rekkefølge og metode er avgjørende for suksess.

#### 🔹 Narrativ progresjon

**Beskrivelse:**  
Spilleren avdekker gradvis historien gjennom logger, romutforskning og gåteløsning.

* Kombinerer:
  + **Utforskning**
  + **Gåteløsning**
  + **Inventar og Ressurshåndtering** (finne nøkler, aktivere terminaler)

**Effekt:**  
Skaper motivasjon til å fortsette å utforske, med en følelse av belønning når man finner nye biter av historien.

### 📌 Bruksområder

* Skaper dybde i gameplayet ved å kreve at spilleren kombinerer flere ferdigheter og valg samtidig.
* Øker spillerens følelse av mestring, fordi man må bruke både kunnskap om mekanikkene og strategisk planlegging.
* Gir en dynamisk spillopplevelse som utvikler seg over tid.

### 📎 Tilleggsnotat

* Samspillsmekanismer er ofte emergente: de oppstår naturlig når spilleren utforsker og interagerer med de ulike grunnmekanismene.
* De er viktige å identifisere tidlig i designprosessen for å sikre balanse og godt gameplay.
* I undervisning eller dokumentasjon kan de illustreres med diagrammer (f.eks. mekanikk-kart eller flows).

## 🔹 3. Tredje nivå (Metamekanismer / Systemiske mønstre)

Disse er på spilldesignnivå og handler om spillers opplevelse og læring over tid – ofte tverrgående og narrative.

**Eksempler:**

* **Valg og konsekvens:** Valg du gjør i gåter, dialog eller ressursbruk påvirker slutten.
* **Narrativ progresjon gjennom utforskning:** Historien låses gradvis opp ved å bruke flere mekanismer samtidig.
* **Pacing-mønstre:** Spilleren veksler mellom intense og rolige øyeblikk, styrt av spillstrukturen.

### 🎯 Formål

Beskrive de overordnede mønstrene og opplevelsene som oppstår som resultat av samspillet mellom grunnmekanismer og samspillsmekanismer. Metamekanismer skaper helhetsopplevelsen og de emosjonelle og narrative opplevelsene i spillet.

### 🔧 Beskrivelse

* Metamekanismer oppstår når flere mekanismer og dynamikker i spillet jobber sammen over tid og skaper helhetlige systemer som ikke er direkte kodet som én enkelt mekanisme.
* Disse systemene påvirker hvordan spilleren tolker spillet, hvilke strategier de bruker, og hvordan de opplever narrativet.
* I *Moonbase Exodus* skaper metamekanismene en følelse av isolasjon, frykt, utforskning, stress og mestring.

### 📑 Eksempler på Metamekanismer i Moonbase Exodus

#### 🔹 Valg og konsekvens

**Beskrivelse:**  
Spillerens valg underveis påvirker hvordan historien utvikler seg (f.eks. hvem man redder, hvilke logger man finner, om man klarer å advare jorden).

* Bygger på:
  + **Gåteløsning**
  + **Utforskning**
  + **Inventarbruk**
* Effekt:
  + Flere avslutninger gir spilleren eierskap til historien og gjenspillingsverdi.

#### 🔹 Narrativ progresjon gjennom miljøet

**Beskrivelse:**  
Spilleren lærer om hendelsene på basen ved å utforske rom, lese logger og løse gåter.

* Bygger på:
  + **Utforskning**
  + **Spillerens Tilstand**
  + **Unngå fiender**
* Effekt:
  + Historien blir en integrert del av gameplayet, ikke bare en bakgrunn.

#### 🔹 Stress og press

**Beskrivelse:**  
Kombinasjonen av oksygendrening, ressursmangel og fiendetrusler skaper en følelse av desperasjon og hastverk.

* Bygger på:
  + **Oksygendrening**
  + **Fiendeunngåelse**
  + **Ressurshåndtering**
* Effekt:
  + Gir en intens og uforutsigbar spillopplevelse som holder spilleren på tå hev.

#### 🔹 Emergent gameplay

**Beskrivelse:**  
Spilleren finner kreative løsninger ved å kombinere mekanismer (f.eks. bruke en nøkkelkortdør som fluktrute fra en fiende).

* Bygger på:
  + **Utforskning**
  + **Inventarbruk**
  + **Fiendeunngåelse**
* Effekt:
  + Spilleren føler eierskap til strategiene de utvikler selv.

### 📌 Bruksområder

* Skaper en meningsfull helhet av spillets mekanikker.
* Gir spilleren emosjonelle høydepunkter og motivasjon til å fortsette.
* Bidrar til narrativ dybde og replay-verdi.

### 📎 Tilleggsnotat

* Metamekanismer er ofte ikke kodet som egne funksjoner, men som resultater av hvordan mekanikkene er designet til å samvirke.
* Viktig å teste og justere for å sikre at opplevelsen matcher intensjonen (f.eks. at stressnivået er spennende, ikke frustrerende).
* Kan visualiseres med et “mekanikk-kart” for å vise hvordan de ulike mekanismene leder til ulike opplevelser.

## 📐 Bonus: MDA-rammeverket og nivåene

I *MDA (Mechanics–Dynamics–Aesthetics)* henger nivåene ofte slik sammen:

| **Nivå** | **Tilsvarer i MDA** | **Eksempel fra spillet** |
| --- | --- | --- |
| Grunnmekanismer | Mechanics | "Spilleren mister oksygen hvert sekund." |
| Samspillsmekanismer | Dynamics | "Oksygen, sniking og gåter skaper stress." |
| Metamekanismer | Aesthetics | "Følelsen av panikk, mestring, frykt." |

# 🎮 Player Experience (PX)

## 🎯 Formål

Definere hvilken opplevelse vi ønsker at spilleren skal ha mens de spiller *Moonbase Exodus*. Dette inkluderer hvilke følelser, motivasjoner og handlinger vi ønsker å fremkalle for å sikre et engasjerende og minneverdig spill.

## 🔧 Beskrivelse

Player Experience beskriver helhetsopplevelsen spilleren har – fra emosjonelle reaksjoner til motivasjon og læring. PX er et resultat av hvordan alle mekanismene (grunnmekanismer, samspillsmekanismer og metamekanismer) henger sammen og skaper en dynamisk, engasjerende opplevelse.

## 📝 PX-mål

### 🔹 Spenning og stress

* Spilleren skal føle et konstant press fra oksygendrening, fiender og ressursmangel.
* Skape et adrenalinfylt gameplay med intensitet og raske beslutninger.

### 🔹 Nysgjerrighet og oppdagelse

* Spilleren motiveres til å utforske månebasen rom for rom.
* Oppleve gleden av å finne ledetråder, logs og ressurser som kaster lys over mysteriene på basen.

### 🔹 Overlevelsesfølelse

* Spilleren må konstant balansere risiko og ressursbruk.
* Prioritere handlinger for å unngå døden – skaper en realistisk og desperat overlevelseskamp.

### 🔹 Mestring og tilfredsstillelse

* Spilleren skal oppleve mestring ved å løse gåter, finne ressurser og overleve fiender.
* Følelsen av fremgang, kontroll og progresjon er sentral.

### 🔹 Frykt og isolasjon

* Spilleren skal føle seg alene og utsatt, omgitt av ukjente farer.
* Ubehagelige fiender og mørke, øde omgivelser forsterker atmosfæren.

### 🔹 Historiedrevet engasjement

* Spilleren skal trekkes inn i mysteriet bak mutasjonen og basens skjebne.
* Avdekke historien gradvis gjennom logs og miljødetaljer – skaper motivasjon for å fortsette.

## 📌 Kort oppsummering

|  |  |
| --- | --- |
| Opplevelseskategori | Beskrivelse |
| Spenning | Oksygendrening og fiender skaper stress og intensitet. |
| Nysgjerrighet | Utforskning gir belønninger i form av spor og historie. |
| Overlevelse | Ressurshåndtering og risikoavveiing driver gameplayet. |
| Mestring | Løs gåter og overkomme hindringer for progresjon. |
| Frykt | Mørk atmosfære og uforutsigbare fiender skaper uro. |
| Historie | Opplev en gradvis avdekket narrativ med dypere mening. |

**📎 Tilleggsnotat**

* Player Experience bør vurderes kontinuerlig under utviklingen for å sikre at mekanismene faktisk leverer på ønsket opplevelse.
* Kan styrkes med visuelle eller auditive virkemidler (f.eks. pustelyder når oksygennivået er lavt, skygger, alarmlyder).
* Spillet bør balansere intensitet (spenning og frykt) med rolige øyeblikk (mestring og oppdagelse) for å unngå overveldelse.

# 📘 Dokumentasjon: Game Balance

## 🎯 Formål

Sikre at spillet gir en rettferdig, utfordrende og engasjerende opplevelse for spillerne, uavhengig av ferdighetsnivå. Game Balance handler om å justere spillets mekanismer slik at ingen elementer er for lette eller for vanskelige, og at alle valg og strategier har meningsfulle konsekvenser.

## 🔧 Beskrivelse

Game Balance definerer forholdet mellom spillmekanismer, fiender, ressurser og progresjon. Målet er å skape en flyt der spillet er utfordrende, men samtidig overkommelig – og at spillets vanskelighetsgrad øker naturlig i takt med spillerens ferdigheter.

## 📑 Viktige balanseområder i Moonbase Exodus

### 🔹 Oksygendrening

* **Utfordring:** Oksygendrening skal skape konstant tidspress uten å være urettferdig.
* **Balansering:**
  + Hastighet: Justere oksygendrening per sekund.
  + Ressursfrekvens: Hvor ofte spilleren finner oksygentanker.

### 🔹 Fiender

* **Utfordring:** Fiendene skal være truende, men ikke umulige å unngå.
* **Balansering:**
  + Antall fiender per rom.
  + Fiendens skadepotensial (f.eks. oksygentap).
  + Muligheter for sniking eller flukt.

### 🔹 Ressurshåndtering

* **Utfordring:** Spilleren må prioritere hvilke ressurser som brukes når.
* **Balansering:**
  + Antall ressurser i hvert rom.
  + Variasjon i ressursbruk (f.eks. nøkkelkort, oksygentanker).

### 🔹 Gåteløsning

* **Utfordring:** Gåtene må være utfordrende, men ikke frustrerende.
* **Balansering:**
  + Vanskelighetsgrad tilpasset progresjon (f.eks. første gåter enklere).
  + Hintsystem eller logger som gradvis gir hint.

## 📌 Balansestrategier

|  |  |
| --- | --- |
| Område | Tiltak |
| Oksygennivå | Test med ulike dreneringshastigheter og ressursfrekvenser. |
| Fiender | Juster patruljemønster, skadeverdier og antall fiender per rom. |
| Ressurser | Varier tilgangen til ressurser slik at spilleren må utforske. |
| Gåter | Implementer vanskelighetskurve og eventuelt hintsystem. |

**📎 Tilleggsnotat**

* Game Balance er en iterativ prosess: Spillet bør testes jevnlig for å justere vanskelighetsgrad basert på tilbakemeldinger fra spillere.
* Balanse kan også tilpasses gjennom vanskelighetsgrader (f.eks. "Easy", "Normal", "Hard") med parametere for oksygendrening, fiendens aggressivitet og ressursmengde.
* Viktig å balansere slik at spillet er utfordrende, men at spilleren alltid føler at de har en reell sjanse til å overleve med de riktige valgene.

# Dokumentasjon: Level Design

## 🎯 Formål

Skape en spennende og variert månebase som spilleren kan utforske. Level Design sørger for progresjon, variasjon og gradvis økende utfordringer ved å strukturere rommene, plassere gåter, ressurser og fiender på en måte som driver historien og gir spilleren en meningsfull opplevelse.

## 🔧 Beskrivelse

Level Design handler om hvordan månebasens rom og områder er utformet og hvordan spilleren navigerer gjennom disse. Det inkluderer:

* Fysiske rom: laboratorier, korridorer, kontorer, lagringsområder osv.
* Tilkoblinger: Hvordan rommene er knyttet sammen (lineært, ikke-lineært, snarveier).
* Ressursplassering: Hvor spillerne finner viktige ressurser (f.eks. oksygentanker, nøkkelkort).
* Fiendeplassering: Hvor fiender dukker opp og hvordan de beveger seg.
* Narrativ fordeling: Hvor spilleren finner logger og hint om historien.
* Progressjon: Hvordan spilleren låser opp nye områder ved hjelp av nøkkelkort eller løste gåter.

## 📑 Viktige elementer i Moonbase Exodus

### 🔹 Struktur

* Rommene organiseres slik at spilleren må utforske og gradvis avdekke historien.
* Noen områder er låst bak dører som krever nøkkelkort eller løste gåter.
* Spillerens progresjon er styrt av:
  + Nøkkelkort
  + Avanserte gåter
  + Ressursbehov (f.eks. finne oksygentanker)

### 🔹 Variasjon

* Kombinasjon av trygge områder (pustepauser) og fareområder (fiender, lav oksygen).
* Blandet layout med korridorer, små rom og større laboratorier for variasjon i gameplay.
* Miljødetaljer som blinkende lys, lekkasjer og mørklagte områder skaper atmosfære.

### 🔹 Ressurser og belønning

* Ressurser plasseres strategisk slik at spilleren må utforske for å finne dem.
* Belønninger som logger eller spesialobjekter gir insentiv til å ta risiko.

### 🔹 Fiender og trusler

* Fiender plasseres slik at de skaper spenning og utfordring uten å blokkere fremdriften totalt.
* Noen områder har fiender, mens andre er trygge (varierer intensitet).

## 🛠️ Eksempelstruktur (enkelt)

Startområde: lab\_entrance

↓

Kontrollrom (første fiende, nøkkelkort)

↓

Hydroponisk sone (ressursbelønning, logg)

↓

Forskningsterminal (gåte som åpner nytt område)

↓

Lagringsområde (fiender, oksygentank)

↓

Skjult lab (nøkkelledetråd, boss-fiende eller større trussel)

↓

Sluttområde (evakuering, avslutning)

## 📌 Bruksområder

* Sørge for en gradvis oppbygning av spenning og utfordring.
* Gi spilleren pauser for ressursforvaltning før nye utfordringer.
* Bruke miljødetaljer og logs for å formidle historien.
* Skape et miljø som føles autentisk som en forlatt månebase.

## 📎 Tilleggsnotat

* Level Design bør testes iterativt med spillere for å sikre god flyt og at det føles rettferdig og spennende.
* Kan kombineres med kartvisning eller guiding (f.eks. hintsystem) for å hjelpe spilleren med navigasjonen.
* Viktig å balansere lineære og åpne områder for å skape variasjon.