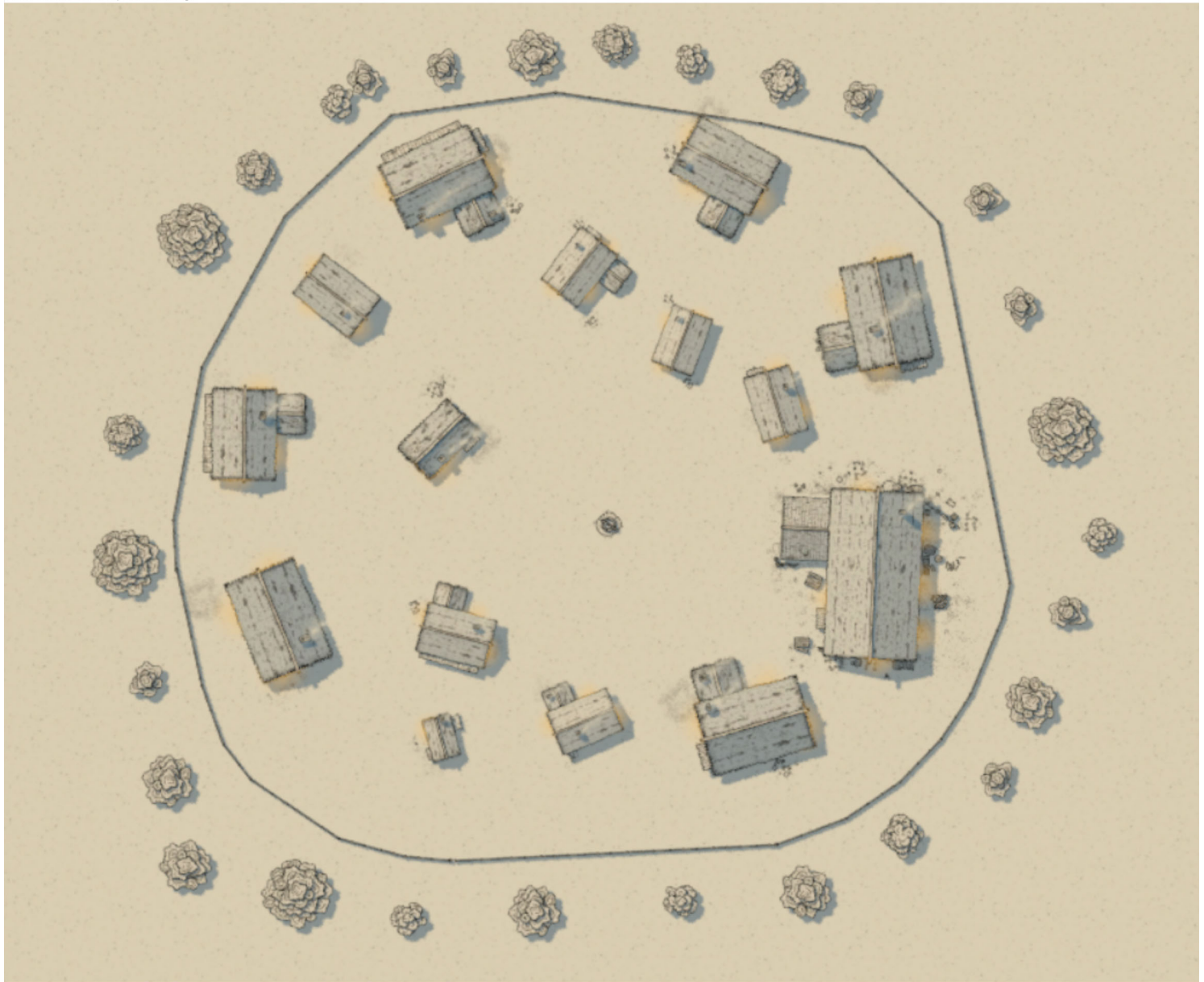


Rolconsistent Spelgedrag bij Tegenstrijdige Personageperspectieven

Een antwoord op de vraag



Johanne Anne Vantorpe

LUCA School of arts

Promotors: Dagmar

Abstract

In deze thesis wordt onderzocht hoe je ervoor zorgt dat een speler zich inleeft in de rol van meerdere personages. Aan de hand van bestaande spellen wordt gekeken hoe die de speler zich in een personage laten inleven. Ook wordt onderzoek over dit thema besproken. Een zelfgemaakt spel test de bevindingen uit het onderzoek.

Keywords: Thesis, Game Design, perspectieven, verhaal, personages, keuzes

Table of Contents

Sectie 1: Inleiding

In dit onderzoek wordt gekeken naar verschillende perspectieven in spellen en verhalen, naar de manier waarop die aan bod komen, naar hoe ze tegenover elkaar kunnen worden geplaatst en naar hoe spelers zich met personages kunnen identificeren.

De onderzoeksvraag voor deze thesis is:

Welke ontwerpstrategieën kunnen ervoor zorgen dat een speler die met twee tegenstrijdige personages speelt, wordt gestimuleerd om steeds rolconsistente keuzes te maken?

Het idee is om een spel te maken waarin de speler twee perspectieven beleeft die tegenover elkaar staan. De bedoeling is dat de speler zich inleeft in beide perspectieven en keuzes maakt die bij het actieve personage passen.

Sectie 2: State of the art

In deze sectie worden bestaande spellen besproken die relevant zijn voor het onderzoek naar rolconforme keuzes bij perspectiefwissels. De voorbeelden tonen hoe verschillende ontwerpstrategieën — zoals perspectief-geheugen, omgevingsgestuurde rolversterking, NPC-reactiviteit en gelijkwaardige dilemma's — het gedrag van spelers kunnen beïnvloeden. Deze inzichten vormen de basis voor de vijf factoren in sectie 4, de design values in sectie 5 en de verhaalstructuur in sectie 7.

Spelen met verschillende perspectieven van personages

Zelfde verhaal, ander perspectief — *NieR Replicant* en *NieR: Automata* ^{1 2}

Een relevant voorbeeld zijn *NieR Replicant* en *NieR: Automata*, waarin de speler hetzelfde verhaal meerdere keren doorloopt, telkens vanuit een ander personage. De tweede playthrough onthult informatie die het eerste personage nooit kende, waardoor gebeurtenissen die eerder vanzelfsprekend leken, moreel ambigu of zelfs tragisch blijken. De speler kan geen andere keuzes maken, maar de *interpretatie* van dezelfde scènes verandert volledig afhankelijk van het perspectief.

Dit ondersteunt meerdere aspecten van dit onderzoek. Ten eerste sluit het aan bij *narratieve motivatie*: het tweede perspectief maakt duidelijk waarom het andere personage handelde zoals het deed. Ten tweede versterkt het de *karakterkaders*: beide personages hebben een eigen manier van denken en reageren, waardoor de speler automatisch anders naar situaties kijkt. Daarnaast toont het hoe *geloofwaardige keuzecontext* ontstaat wanneer dezelfde gebeurtenissen vanuit verschillende motivaties worden ervaren. Ook *NPC-reactiviteit* speelt een rol: vijanden worden niet langer als "vijanden" gezien, maar als personages met eigen doelen, wat de speler subtiel richting ander gedrag stuurt.

Binnen de verhaalstructuur van dit project is dit relevant omdat Airin en Ren dezelfde wereld betreden, maar die anders interpreteren. Net zoals in de *NieR*-spellen ontstaat er een spanning tussen wat de speler weet en wat het personage denkt te weten. Dit sluit aan bij de design value *perspectief-geheugen*. Daarnaast ondersteunt het het principe van *twee gelijkwaardige perspectieven*: beide personages hebben rationele, verdedigbare motivaties die elkaar kruisen zonder moralistische tegenstelling. Tot slot sluit het aan bij *keuze-echo's*: informatie uit het ene perspectief beïnvloedt hoe het andere perspectief wordt ervaren.

Ander personage met een eigen doel — *Baldur's Gate 3* ³

Een tweede relevante categorie bestaat uit spellen waarin de speler een personage bestuurt dat een *eigen, autonoom doel* heeft dat losstaat van het hoofdverhaal. Dit ontwerpprincipe is interessant voor dit onderzoek omdat het laat zien hoe *persoonlijke motivaties* en *karaktergebonden doelen* spelers richting rolconform gedrag kunnen sturen.

Baldur's Gate 3 is een duidelijk voorbeeld hiervan. De speler kan kiezen tussen een zelfgemaakt personage of een van de vooraf geschreven *Origin Characters*. Deze origin-personages hebben een eigen verleden, persoonlijkheid en duidelijke persoonlijke doelen die actief door het spel worden ondersteund. Wanneer de speler een origin-personage kiest, krijgt die toegang tot unieke dialogen, exclusieve verhaallijnen en NPC-reacties die specifiek op dat personage zijn afgestemd. Deze doelen zijn niet optioneel: ze vormen een kernonderdeel van het personage en beïnvloeden hoe de speler de wereld ervaart en welke keuzes logisch aanvoelen. Uit statistieken blijkt dat de meeste spelers toch een eigen personage maken, waardoor zij deze rolgestuurde narratieve structuur niet ervaren. Dit onderstreept hoe belangrijk *duidelijke persoonlijke motivaties* zijn om spelers in een specifieke mindset te brengen. Spelers die wél een origin-personage spelen, rapporteren dat ze vaker keuzes maken die bij dat personage passen, zelfs wanneer die keuzes niet overeenkomen met hun eigen voorkeuren. NPC-reacties versterken dit effect: personages reageren bijvoorbeeld anders op Astarion dan op Shadowheart, waardoor de speler subtiel wordt gestuurd richting gedrag dat bij het actieve perspectief past.

Dit principe ondersteunt meerdere aspecten van dit onderzoek. Ten eerste sluit het aan bij de factor *narratieve motivatie*: een autonoom doel maakt duidelijk waarom een personage iets wil, waardoor spelers sneller begrijpen welk gedrag passend is. Ten tweede versterkt het de *karakterkaders*: een personage met een eigen doel en achtergrond wordt herkenbaar en voorspelbaar, wat rolconsistentie bevordert. Ten derde speelt *NPC-reactiviteit* een grote rol: sociale feedback van NPC's helpt de speler te handelen op een manier die logisch is binnen het perspectief van het personage.

Ook binnen de verhaalstructuur van dit project is dit relevant. Airin en Ren hebben elk een persoonlijk doel dat logisch voortkomt uit hun achtergrond, maar dat direct met het doel van de ander botst. Net zoals bij origin-personages ontstaat hierdoor een natuurlijke spanning: de speler moet telkens vanuit een ander perspectief redeneren, zelfs wanneer die als speler meer informatie heeft dan het personage. Dit sluit aan bij de design value *twee tegenstrijdige maar gelijkwaardige perspectieven*: beide doelen zijn rationeel en verdedigbaar, zonder dat het ene moreel beter is dan het andere.

Daarnaast ondersteunt dit principe de design value *perspectief-geheugen*. Een autonoom doel creëert een intern kader waarin het personage handelt op basis van wat het denkt te weten, niet op basis van wat de speler weet. Hierdoor ontstaan keuzes die "juist voelen" voor dat personage. Tot slot sluit het aan bij de design value *keuze-echo's*: persoonlijke doelen zorgen ervoor dat keuzes van het ene perspectief subtiele gevolgen hebben voor het andere, zonder dat het verhaal gecorrigeerd wordt.

Door deze elementen samen te brengen, toont het voorbeeld van een *ander personage met een eigen doel* hoe persoonlijke motivaties, sociale feedback en perspectiefgebonden informatie bijdragen aan rolconform gedrag. Dat is precies het soort gedrag dat in dit onderzoek centraal staat.



balsur's gate origin keuze

Zelfde doelen, ander verleden — *Dragon Age: Origins* en *Dragon Age: Inquisition* ^{4 5}

In *Dragon Age: Origins* en *Dragon Age: Inquisition* kiest de speler een *origin* die bepaalt hoe de wereld op het personage reageert. Hoewel alle spelers uiteindelijk hetzelfde hoofdverhaal volgen, beïnvloedt het verleden:

- hoe NPC's het personage behandelen,
- welke dialogen beschikbaar zijn,
- hoe situaties worden geïnterpreteerd,
- en welke keuzes logisch aanvoelen.

Spelers rapporteren dat ze ondanks verschillende origins toch vaak dezelfde keuzes maken, omdat hun eigen speelstijl domineert. Dit toont dat een origin alleen effectief is wanneer het spel blijft benadrukken wat het personage denkt, voelt en wil.

Dit ondersteunt verschillende aspecten van dit onderzoek. Het versterkt *NPC-reactiviteit*: een Dalish Elf wordt anders behandeld dan een Noble Dwarf, wat de speler subtiel richting ander gedrag stuurt. Het ondersteunt *karakterkaders*: het verleden geeft een kader dat de speler kan volgen. Daarnaast draagt het bij aan *geloofwaardige keuzecontext*: keuzes voelen anders aan afhankelijk van het verleden en de sociale positie van het personage.

Binnen de verhaalstructuur sluit dit aan bij het idee dat Airin en Ren een gedeelde oorsprong hebben, maar totaal verschillende levens leiden. Hun verleden bepaalt waarom ze hun huidige doelen nastreven en waarom hun perspectieven botsen. Dit ondersteunt de design value *twee gelijkwaardige perspectieven*, omdat beide personages een valide achtergrond hebben die hun gedrag stuurt.



dragon age origins: noble, city elf, dalish elf, mage, class less dwarf, noble dwarf

Gameplay

Een ander essentieel onderdeel is de manier waarop de gameplay wordt vormgegeven. Daarbij is inspiratie gehaald uit bestaande spellen. Er is bewust gekozen voor een eenvoudige gameplay-structuur, aangezien de nadruk in dit project ligt op het narratieve aspect en op de mate waarin spelers zich kunnen inleven in de personages. Om die reden is vooral gekeken naar spellen die gebruikmaken van point-and-clickmechanieken of andere eenvoudige interactievormen, waarbij technische vaardigheid geen bepalende factor is.

Dialogen en point-and-clickstructuur — *Gamedec*⁶

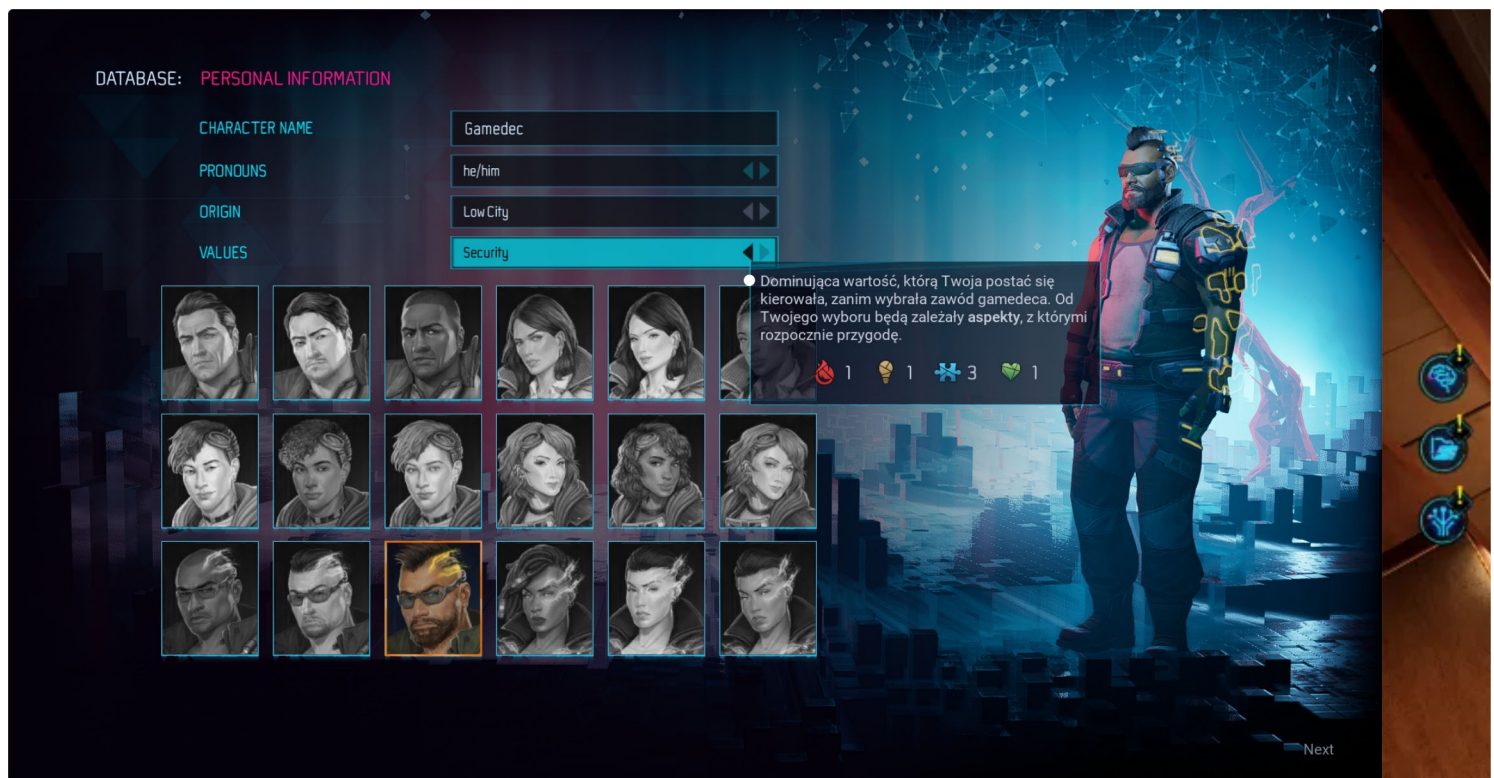
Gamedec is hierbij mijn belangrijkste voorbeeld. Het is een point-and-clickdetectivespel met beperkte vertakkingen. Je kunt het spel met of zonder hulp spelen. Met hulp houdt een in-game journal bij wat je wel en niet hebt geleerd. Verder is de gameplay vrij eenvoudig.

Je loopt rond in de wereld, klikt op wat je wilt bekijken, houdt bij welke informatie je krijgt en maakt keuzes in acties en dialogen die een bepaalde impact hebben.

Een ander aspect van *Gamedec* dat interessant is voor dit onderzoek, is dat je bij het aanmaken van je personage een bepaald verleden kunt kiezen of creëren. Dat heeft weinig impact, behalve op enkele opmerkingen, maar het helpt de speler om op een manier te spelen die bij dat verleden past.

Voor de layout heb ik ook naar andere spellen gekeken om inspiratie op te doen voor de plaatsing van tekst, de verhoudingen en de manier waarop de aandacht van de speler naar de tekst wordt getrokken. Deze spellen waren grotendeels verhaalgedreven.





Gamedec, Gamedec, Project Kat, Roadwarden, Steinsgate, The Deed 2

Omgevingsinteractie — *Final Fantasy IX*⁷ (Vivi en Steiner)

Final Fantasy IX toont hoe animatie, beweging, houding en omgeving de speler subtiel richting rolconform gedrag kunnen sturen. Vivi beweegt klein, voorzichtig en nieuwsgierig; Steiner beweegt groot, hoekig en doelgericht. Spelers rapporteren dat ze automatisch anders spelen, afhankelijk van wie ze besturen.

Dit ondersteunt verschillende aspecten van dit onderzoek. Het versterkt *omgevingsgestuurde rolversterking*: beweging, animatie en omgeving beïnvloeden hoe spelers zich gedragen. Het ondersteunt *karakterkaders*: de fysieke presentatie van het personage maakt duidelijk hoe het denkt en handelt. Daarnaast draagt het bij aan *NPC-reactiviteit*: NPC's behandelen Vivi anders dan Steiner, wat gedrag stuurt.

Binnen de verhaalstructuur sluit dit aan bij de visuele verschillen tussen Airins dorp en Rens kazerne. Het ondersteunt het gebruik van UI-kleur, camerastandpunt en omgeving om het perspectief te benadrukken. Daarnaast versterkt het *rolconsistentie door presentatie*: spelers gaan anders bewegen en denken afhankelijk van het actieve personage.

Dit sluit aan bij de design values *twee gelijkwaardige perspectieven*, *perspectief-geheugen* en *keuze-echo's*, omdat de omgeving eraan herinnert hoe het personage denkt en hoe NPC's reageren.



Vivi en Steiner

Sectie 3: Literair onderzoek

Personagecreatie

Om het spel te laten werken, zijn interessante en geloofwaardige personages nodig, zowel voor de spelerpersonages als voor de NPC's. NPC's mogen minder gedetailleerd zijn, maar moeten nog steeds duidelijk van elkaar verschillen en herkenbare eigenschappen hebben die hun gedrag sturen. In *How Not to Write a Novel*^[1] wordt benadrukt dat perfecte personages saai zijn; imperfecties maken een personage juist menselijk en interessant. Imperfecties zorgen ervoor dat spelers aspecten van zichzelf in het personage kunnen herkennen, waardoor empathie en identificatie ontstaan.

Ondersteuning voor onderzoek

Karakterkaders: Imperfecties en duidelijke eigenschappen helpen spelers te begrijpen hoe een personage denkt en handelt.

Narratieve motivatie: Een personage met zwaktes en conflicten heeft een duidelijker drijfveer.

NPC-reactiviteit: NPC's reageren geloofwaardiger wanneer hun eigen imperfecties en doelen zichtbaar zijn.

Ondersteuning voor verhaalstructuur

Airin en Ren hebben elk een eigen achtergrond, zwaktes en motivaties die hun gedrag sturen.

Imperfecties maken hun doelen begrijpelijk en menselijk, wat het conflict geloofwaardig maakt.

Ondersteuning voor design values

Twee gelijkwaardige perspectieven: Imperfecties zorgen ervoor dat beide personages even menselijk en even verdedigbaar zijn.

Perspectief-geheugen: Imperfecties bepalen wat het personage denkt te weten en hoe het situaties interpreteert.

Keuze-echo's: Zwaktes en karakertrekken beïnvloeden hoe keuzes later doorwerken.

Worldbuilding

Voor de worldbuilding zijn enkele essentiële elementen nodig. Het belangrijkste is een *duidelijk conflict* waarin de personages terechtkomen. Conflict houdt het verhaal in beweging en zorgt ervoor dat personages en NPC's een doel hebben waar zij naar streven. Zonder conflict is er geen spanning, geen motivatie en geen reden voor keuzes.^[2]

Daarnaast zijn *geloofwaardige NPC's* cruciaal. NPC's moeten een eigen doel hebben, een eigen mening, en iets waar ze het niet mee eens zijn. Deze aspecten hoeven niet expliciet uitgesproken te worden, maar moeten wel invloed hebben op hun gedrag en dialogen. NPC's zonder diepgang maken een verhaal oppervlakkig en moeilijk om mee te verbinden.^[3]

Ondersteuning voor onderzoek

Geloofwaardige keuzecontext: Een wereld met duidelijke doelen en conflicten maakt keuzes logisch en betekenisvol.

NPC-reactiviteit: NPC's met eigen doelen reageren anders op Airin dan op Ren, wat rolconsistentie versterkt.

Omgevingsgestuurde rolversterking: De wereld zelf communiceert wat passend gedrag is.

Ondersteuning voor verhaalstructuur

Het conflict tussen Ra'ndom en Mithold vormt de basis voor de spanning tussen Airin en Ren.

NPC's in beide steden hebben verschillende doelen, waardoor de speler telkens een ander sociaal kader ervaart.

Ondersteuning voor design values

Twee gelijkwaardige perspectieven: Beide steden hebben valide redenen voor hun gedrag.

Perspectief-geheugen: NPC's geven hints die alleen logisch zijn voor één perspectief.

Keuze-echo's: NPC-reacties veranderen afhankelijk van eerdere keuzes.

Keuzes

Uit de geanalyseerde talks:

[Defining Choice in Branching Narratives](#)

[Making Player Choices Feel like They Matter in Your Narrative](#)

Blijkt dat keuzes interessant worden wanneer ze:

Impact hebben op personages, wereld of omgeving,
de persoonlijkheid en rol van het personage ondersteunen,
niet voorspelbaar zijn maar wel logisch binnen de context,
geen goed/slecht-structuur hebben maar gelijkwaardige risico's,
emotionele betrokkenheid creëren.

Spelers moeten het gevoel hebben dat hun keuzes ertoe doen, zelfs wanneer de uitkomst subtiel is. Een keuze hoeft niet altijd grote gevolgen te hebben; kleine gevolgen kunnen even betekenisvol zijn wanneer ze rolconsistentie versterken.

Ondersteuning voor onderzoek

Geloofwaardige keuzecontext: Keuzes moeten logisch zijn binnen het perspectief van het actieve personage.

Narratieve motivatie: Keuzes worden betekenisvol wanneer ze aansluiten bij de drijfveren van het personage.

Karakterkaders: Keuzes moeten passen bij de eigenschappen van het personage.

NPC-reactiviteit: NPC's moeten reageren op keuzes om sociale feedback te geven.

Ondersteuning voor verhaalstructuur

Airin en Ren krijgen keuzes die passen bij hun achtergrond en doelen.

Keuzes hebben subtiele gevolgen die het andere perspectief beïnvloeden, zonder het verhaal te corrigeren.

Ondersteuning voor design values

Twee gelijkwaardige perspectieven: Keuzes zijn niet moralistisch maar contextueel.

Perspectief-geheugen: Keuzes worden gestuurd door wat het personage denkt te weten.

Keuze-echo's: Keuzes van het ene perspectief creëren gevolgen voor het andere.

Technieken en tools

Uit de talk [*Technical Tools for Authoring Branching Dialogue*](#) blijkt dat een goede structuur essentieel is voor het schrijven van complexe verhalen. Een verhaal moet apart staan van code en gameplay, zodat de narratieve logica intact blijft. Daarnaast zijn tekstcorrectietools, debugmogelijkheden en testbare structuren noodzakelijk om fouten te voorkomen en consistentie te behouden.

Wanneer een systeem te complex wordt, moet het vereenvoudigd worden om de structuur te behouden. Dit voorkomt dat het verhaal onoverzichtelijk wordt en dat keuzes hun betekenis verliezen.

Sectie 4: Onderzoek

Uit het state-of-the-artonderzoek en de geanalyseerde talks blijkt dat spelers hun keuzes vooral afstemmen op *hoe zij het personage begrijpen, hoe de wereld op hen reageert en hoe duidelijk het spel communiceert welk gedrag passend is*. Daaruit komen vijf overkoepelende factoren naar voren die bepalen of een speler consequent handelt in lijn met het actieve personage:

Narratieve motivatie: het verhaal moet duidelijk maken *waarom* het personage iets wil.

Karakterkaders: het personage moet herkenbare eigenschappen hebben die de speler kan volgen.

Geloofwaardige keuzecontext: keuzes moeten logisch, begrijpelijk en situationeel passend zijn.

Omgevingsgestuurde rolversterking: de wereld moet subtiel aangeven hoe dit personage zich beweegt en denkt.

NPC-reactiviteit: reacties van andere personages sturen de speler richting rolconform gedrag.

Deze factoren vormen samen een kader om te testen of spelers daadwerkelijk keuzes maken die passen bij het perspectief dat ze op dat moment besturen.

Het verhaal biedt duidelijke motivatie voor rolconform gedrag

Omdat spelers verschillende voorkeuren hebben, werkt een complex verhaal niet altijd als leidraad voor gedrag. Daarom is gekozen voor een *compact narratief conflict* waarin beide personages een duidelijke persoonlijke motivatie hebben:

Airin wil zijn grootvader helpen.

Ren moet de gestolen documenten onderscheppen.

Dit sluit aan bij inzichten uit de talks: spelers volgen een personage beter wanneer de *emotionele drijfveren helder en begrijpelijk zijn*. Familie, plicht en dilemma's zijn thema's die spelers intuïtief herkennen, waardoor ze sneller begrijpen *wat dit personage zou doen*.

Het personage reageert geloofwaardig op situaties en biedt passende keuzes

Spelers maken rolconforme keuzes wanneer het spel:

duidelijke, geloofwaardige opties biedt,
geen "goed vs slecht"-keuzes presenteert,
gelijkwaardige opties biedt die elk passen bij een ander perspectief.

Dit sluit aan bij de panels: keuzes moeten *gelijkwaardig, contextueel en niet-moralistisch* zijn.

In het prototype krijgt de speler bijvoorbeeld meerdere manieren om informatie te verkrijgen over hoe een bepaalde locatie bereikt kan worden. Elke optie:

heeft een logische reden om te werken of te falen,
past bij een andere manier van denken,
en maakt duidelijk *waarom* het personage dit zou proberen.

Daardoor voelt het resultaat geloofwaardig, zelfs wanneer een keuze niet werkt of dezelfde informatie geeft als een andere optie wanneer beide opties werken.

De wereld versterkt het perspectief van het actieve personage

Uit zowel Final Fantasy IX als de heuristics-talk blijkt dat *omgeving en presentatie* een grote invloed hebben op hoe spelers zich gedragen.

Daarom is de wereld in het prototype bewust gebruikt om het actieve perspectief te versterken:

Airin krijgt een omgeving die *groter, ruimer en natuurlijker* aanvoelt → Airin start in het open dorp Ra'ndom.
Ren krijgt een omgeving die *gestructureerd, controlegericht en militair* aanvoelt → Ren start in de militaire kazerne.

Beide perspectieven hebben ook een ander kleurenpalet, kleding en houding.

NPC-reacties sturen de speler richting passend gedrag

NPC's zijn een van de sterkste middelen om spelers te laten begrijpen hoe een personage hoort te handelen. Dit sluit aan bij alle drie de talks: NPC-reactiviteit is cruciaal voor de impact.

In het prototype:

NPC's reageren anders op Airin dan op Ren: vriendelijker, autoritairder, vijandiger of wantrouwiger.
NPC's kunnen zich verstoppen, afstand houden, of juist open zijn.
Sommige NPC's geven hints die alleen logisch zijn vanuit één perspectief. "Pas op voor de Misthold-soldaten" is bijvoorbeeld geen logische hint voor Ren, omdat hij zelf een Misthold-soldaat is. Voor Airin past deze hint wel, omdat Misthold hem in de weg zit.

Daardoor krijgt de speler *sociale feedback*:

Airin merkt dat mensen hem vertrouwen → hij handelt empathischer en informeler, vanuit het standpunt van een dorpsbewoner of dorpsjager.
Ren merkt dat mensen hem wantrouwen → hij handelt formeler en voorzichtiger, vanuit de structuur van een soldaat.

NPC-reacties creëren zo een *sociaal kader* dat de speler helpt om rolconform te blijven.

Samenvattend

Deze vijf factoren — narratieve motivatie, karakterkaders, geloofwaardige keuzecontext, omgevingsgestuurde rolversterking en NPC-reactiviteit — vormen samen een systeem dat de speler subtiel maar effectief richting gedrag stuurt dat past bij het actieve personage.

Ze sluiten aan bij moderne inzichten uit branching narrative design:

Cascades → grote keuzes die perspectief vormgeven

Callbacks → kleine verwijzingen die rolconsistentie versterken

Heuristics → relationele systemen die NPC-reacties dynamisch maken

Contextuele informatie → spelers moeten begrijpen waarom keuzes logisch zijn

Variatie in keuzetypes → niet elke keuze moet groot zijn, maar wel betekenisvol

Sectie 5: Mijn design values

Twee tegenstrijdige gelijkwaardige perspectieven

De twee perspectieven moeten met elkaar in conflict staan, maar ook gelijkwaardig zijn. Geen van beide perspectieven mag absoluut goed of slecht zijn.

Perspectief-geheugen

Het spel benadrukt wat het personage *denkt te weten*, niet wat de speler weet.

Hierdoor ontstaan keuzes die "juist voelen" voor dat personage, zelfs als de speler meer informatie heeft.

Airin weet niet wat Ren weet

Airin kent de militaire situatie niet, weet niet dat documenten gestolen zijn, en interpreteert soldaten als obstakels.

Ren weet niet wat Airin weet

Ren kent Airins familieproblemen niet, weet niet waarom hij naar Sunhold moet, en interpreteert Airin als een mogelijke smokkelaar.

De speler kent beide perspectieven

Maar moet *rolconform* blijven binnen het actieve personage.

Het spel maakt duidelijk wat elk personage wel en niet weet

Via:

- UI-kleur
- perspectiefscherm
- NPC-reacties
- omgeving
- dialogen
- hintsysteem

Dit versterkt keuzes

Omdat de speler begrijpt:

"Airin zou dit doen, zelfs als ik als speler weet dat Ren later problemen krijgt."
"Ren zou dit doen, zelfs als ik als speler weet dat Airin haast heeft."

Keuze-echo's

Keuzes van het ene personage creëren *echo's* die het andere personage beïnvloeden, maar nooit op een manier die het verhaal "corrigeert":

Een keuze van Airin maakt Rens taak moeilijker, maar niet onmogelijk.

Een keuze van Ren vertraagt Airin, maar blokkeert hem niet.

Sectie 6: Prototyping

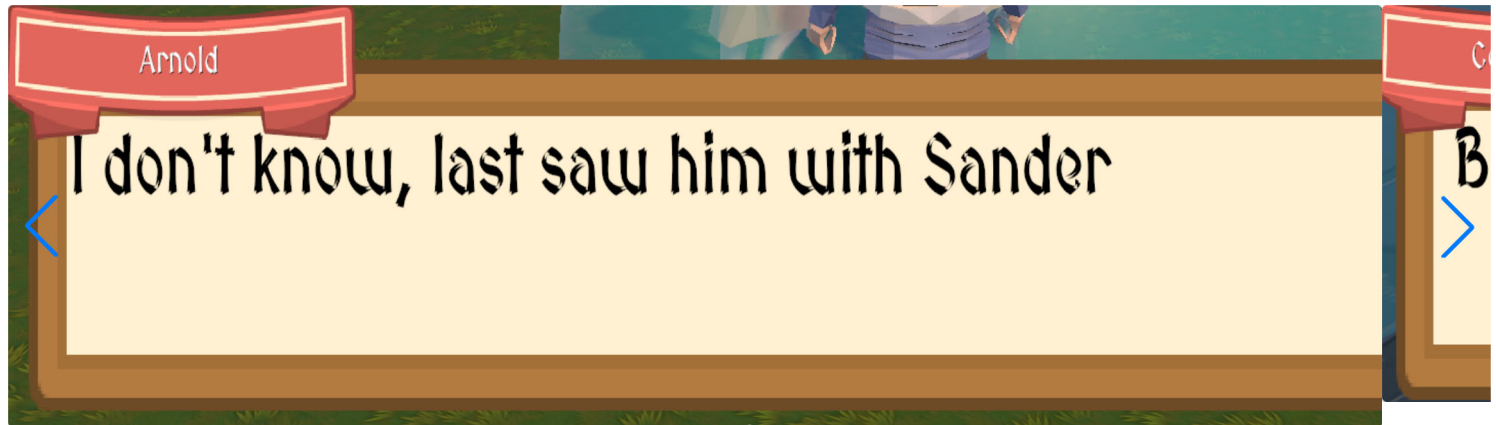
Bij het prototypen ben ik even de weg kwijtgeraakt in wat ik precies wilde onderzoeken. Ik heb daarbij te veel tijd verloren aan het maken van een spel dat niet echt aansloot bij wat ik wilde doen.

Eerste prototypes

Voor mijn eerste prototypes heb ik Twine getest met een eenvoudig verhaal en een keuzesysteem. Op deze manier kon ik het verhaal en de keuzes aanmaken en testen.

Daarbij raakte ik mijn oorspronkelijke idee om de perspectieven tegenover elkaar te plaatsen echter kwijt.

Daarna ben ik veel tijd verloren met het overzetten van Twine naar Unity. Uiteindelijk raakte ik daardoor helemaal de weg kwijt.



Go talk to ...



eerste prototypes screenshots

Daarom ben ik voor de latere prototypes eerst naar de basis teruggekeerd.

Latere prototypes

Daarbij ben ik teruggekeerd naar de basis: een verhaal met twee tegenover elkaar staande perspectieven. Ik heb het verhaal ook verkleind, zodat het zich alleen in de beginzone afspeelt. Het conflict is nog steeds het gevolg van de rivaliteit tussen de twee steden, maar is kleinschaliger. Daardoor verloor ik er minder tijd aan.

Daarna heb ik mezelf meteen de vraag gesteld hoe ik dit van pure tekst naar een spel kon omzetten. Daarbij heb ik inspiratie gehaald uit textadventures en hun evoluties. Ook heb ik naar andere verhaalgerichte spellen gekeken.

Verhaal met gameplay

Het idee was om een spel te maken waarin het verhaal centraal stond, maar waarin ook andere gameplay-elementen voorkwamen.

Daarvoor kwam ik uit op een eenvoudige structuur: een keuzespel waarin de speler eerst op zoek gaat naar bepaalde personen of voorwerpen om een interactie te activeren.

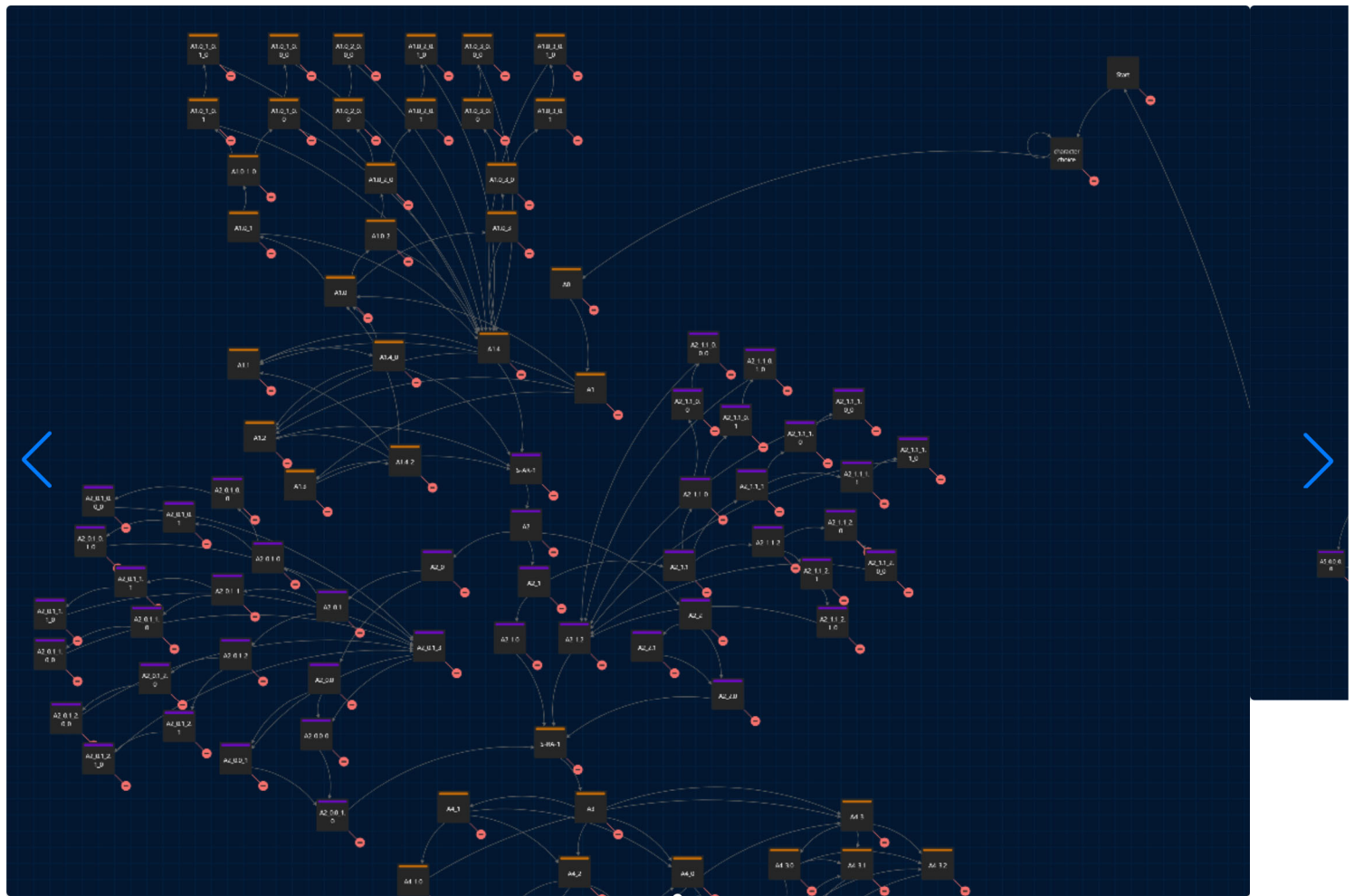
Hoe zag dit er concreet uit?

Ik heb hiervoor inspiratie gehaald uit een Unity-asset die een Twine-bestand speelbaar maakte in Unity. Ik heb de asset zelf niet gebruikt omdat die te beperkend was, maar heb er wel elementen uit gehaald. Zo kreeg ik een eenvoudig aan te passen Twine-structuur die ik in Unity kon plaatsen zonder verdere code aan te passen.

Op deze manier kon ik verhaalelementen en gameplay-elementen apart aanpassen.

Eerste stap prototypes

Mijn eerste prototype was opnieuw een Twine-verhaal. Daarmee wilde ik nagaan wat precies nodig was: waar het verhaal aangepast moest worden, welke elementen niet in Twine konden en in de Unity-versie aan bod moesten komen, en hoe de Twine-structuur precies werkte.



twine structuur en spel

Uit deze prototypes en playtests heb ik het volgende kunnen halen:

Twine werkt met een eenvoudig systeem van *passages*. Deze *passages* linken naar volgende *passages*, maar hebben geen zicht op de volledige structuur. Wanneer het wordt geconverteerd naar een *.twee*-bestand, heeft het een leesbare structuur die aangepast kan worden.

Tweede stap prototypes

Op basis van de resultaten van het eerste prototype heb ik een eenvoudig Python-script kunnen maken dat het *.twee*-bestand naar een *.json*-bestand converteert. Dat kon ik vervolgens makkelijk in Unity gebruiken.

Toen dat gedaan was, kon ik met één script het Twine-spel in Unity spelen.

Op deze manier kon ik gameplay-aspecten in Unity aanpassen zonder aan het verhaal te komen, en verhaalelementen in Twine aanpassen zonder aan de gameplay te komen.

Dit zorgde ervoor dat ik veel minder tijd verloor aan debuggen wanneer er iets aangepast moest worden.

In de gameplay komen tussen keuzemomenten objectieven die de speler moet voltooien. Op dit moment gaat het eenvoudigweg om "vind deze persoon", maar het systeem is zo geïmplementeerd dat ik meer gevarieerde objectieven kan aanmaken zonder het verhaal te moeten aanpassen.

Elk objectief heeft een ID die in het Twine-bestand staat. Op die manier kan ik de objectieven in Unity zonder problemen aanpassen, omdat de ID niet verandert. Dit helpt ook om de structuur te behouden: wanneer ik een objectief wil maken, hoef ik alleen te kijken welke ID ik eraan moet geven. Daarna wordt het vanzelf aan het verhaal gekoppeld.

Momenteel gebruik ik voor de beweging de standaard third-personasset. Zo hoef ik voorlopig geen tijd te besteden aan het maken van een point-and-clickpathfindersysteem. Daar kan ik tijdens het masterjaar aan werken. Het geeft me ook de mogelijkheid om andere manieren van bewegen te onderzoeken.

Uit deze prototypes en playtests heb ik het volgende kunnen halen:

Door het verhaal en de code apart te houden, is het makkelijker om een duidelijke structuur te behouden.

Ik kan veel sneller aanpassingen maken na commentaar van playtesters.

Playtesters vonden het verhaal en de intrige interessant en wilden het vervolg weten.

Commentaar uit de playtests:

Het was niet altijd duidelijk wanneer er van perspectief werd gewisseld.

De camera stond te dicht bij de speler, waardoor spelers moeite hadden om hun objectieven te vinden.
De UI was onduidelijk.

Voor de assets heb ik het dorp uit de vorige versie gebruikt, omdat het al gemaakt was.

Derde stap prototypes

Met dit commentaar ben ik aan de slag gegaan. Ik heb ervoor gezorgd dat er een apart scherm verschijnt wanneer er van perspectief wordt gewisseld. De speler kan pas daarna verder met het spel, zodat de perspectiefwissel de nodige aandacht krijgt. Ook ben ik AI-gegenereerde afbeeldingen gaan gebruiken als placeholders en om een idee te geven van het uiteindelijke resultaat. Het is de bedoeling dat deze afbeeldingen niet permanent zijn. Ze zijn wel al gebaseerd op de beschrijvingen van de personages en de plaatsen. Voor de UI heb ik spelers de mogelijkheid gegeven om het lettertype aan te passen, zodat ze konden aangeven welk lettertype ze het meest passend en leesbaar vonden.



font choice

Ook heb ik ervoor gezorgd dat de kleur van de UI verandert afhankelijk van het perspectief dat op dat moment aan bod is.

Voor de camera heb ik tijdens meerdere playtests voortdurend aanpassingen gemaakt, totdat ik een opstelling had gevonden waar de spelers geen problemen meer mee hadden. Ik heb de camera hoger en verder van de speler geplaatst, zodat spelers een beter zicht hadden en zich minder snel opgesloten voelden.

Uit deze prototypes en playtests heb ik het volgende kunnen halen:

UIToolkit is een sterke tool, maar heeft veel bugs waar je omheen moet werken.

Commentaar uit de playtests:

De layout van de tekst en keuzes werd onlogisch en verwarrend gevonden.

De hints voor de objectieven mochten een optionele toggle zijn, omdat ze de aandacht afleidden wanneer spelers ze niet nodig hadden.

De afbeeldingen hielpen om context te geven bij de tekst.

Finale stap prototypes

In mijn laatste prototypes heb ik de manier waarop de hints worden getoond aangepast zodat de speler zelf kan kiezen wanneer hints nodig zijn.



hint systeem

Ook heb ik verschillende layouts aangemaakt waaruit spelers konden kiezen. Daarna konden ze aangeven welke layout ze het beste vonden en waarom.



layouts

Uit de laatste playtests heb ik het volgende geleerd:

De spelers vonden de layout waarin de keuzes boven de afbeelding verschijnen het beste. Ze gaven aan dat de andere layouts de aandacht van de tekst of afbeelding weghalen. De layout met de keuzes boven de afbeelding zorgt ervoor dat de aandacht tot het laatste moment bij de tekst en afbeelding blijft en pas naar de keuzes gaat wanneer die tevoorschijn komen.

Het nieuwe hintsysteem zorgt ervoor dat spelers beter zien wanneer ze hun objectieven moeten voltooien, zonder dat hun aandacht voortdurend wordt afgeleid.

In this story the player will follow two characters, Airin and Ren.

Airin and Ren are both originally from the small village of Ra'ndom, which lies exactly between the rival cities of Sunhold

Airin chooses
Sunhold as



Layout and hintsysteem

Sectie 7: Verhaalstructuur

Opzet

Om te onderzoeken hoe spelers *rolconforme keuzes* maken wanneer zij twee tegenstrijdige perspectieven besturen, is het verhaal opgebouwd rond twee personages die uit dezelfde context komen, maar een totaal verschillend pad hebben gevolgd. Deze gedeelde oorsprong zorgt voor herkenbaarheid, terwijl hun uiteenlopende levens ervoor zorgen dat de speler telkens naar een andere mindset moet schakelen.

Airin bleef in Ra'ndom en leeft een rustig, dorpsgericht bestaan.
Ren verliet het dorp en werd soldaat in Misthold.

Door beide personages terug te plaatsen in dezelfde omgeving, maar met *verschillende motivaties*, ontstaat een natuurlijke spanning: de speler herkent de wereld, maar moet telkens vanuit een andere rol denken. Dit sluit aan bij het onderzoeksprincipe dat *duidelijke motivatie en context* essentieel zijn voor rolconsistentie.

Daarnaast zijn beide personages *niet absoluut goed of slecht*. Hun doelen botsen, maar zijn elk rationeel en emotioneel verdedigbaar. Dit voorkomt moralistische keuzes en ondersteunt het principe van *gelijkwaardige, contextuele dilemma's*, zoals aanbevolen in de talks.

Doelen

De kern van het verhaal is een gedeeld probleem dat voor beide personages een andere betekenis heeft. Hierdoor ontstaat een *natuurlijk conflict* dat de speler dwingt om telkens vanuit het actieve perspectief te redeneren.

Airin moet Ra'ndom verlaten om zijn grootvader in Sunhold te helpen.
Ren moet ervoor zorgen dat niemand het dorp verlaat, omdat gestolen documenten mogelijk via Ra'ndom worden gesmokkeld. Als dit niet lukt, kan Ren zijn werk verliezen.

Beide doelen zijn:

persoonlijk gemotiveerd,
logisch binnen hun achtergrond,
direct in conflict,
en *niet moreel hiërarchisch* (geen goed/slecht).

Dit sluit aan bij het onderzoeksprincipe dat spelers rolconform handelen wanneer de *motivatie helder, menselijk en begrijpelijk* is.

De keuzes die de speler maakt met het ene personage hebben *indirecte gevolgen* voor het andere. Deze gevolgen zijn niet altijd onmiddellijk zichtbaar, wat aansluit bij het concept van *cascades*: grote beslissingen die later doorwerken zonder dat de speler meteen ziet hoe.

Afhankelijk van de gemaakte keuzes kan een personage zelfs *van groep of alliantie veranderen*, wat de impact van eerdere beslissingen versterkt.

Keuzes en impact

Het verhaal maakt gebruik van verschillende lagen van impact, gebaseerd op de inzichten uit de audiotranscripten:

1. Kleine variaties (callbacks)

Veel keuzes veroorzaken subtiele verschillen in reacties, informatie, of sociale dynamiek. Deze kleine callbacks zorgen ervoor dat de speler voelt dat het spel hun gedrag volgt, zonder dat de structuur explodeert in complexiteit.

2. Middelgrote variaties (heuristics)

NPC's reageren anders afhankelijk van eerdere keuzes, relatie-opbouw en perspectief. Dit versterkt rolconsistentie: de speler past zijn gedrag aan op basis van hoe de wereld op hem reageert.

3. Grote variaties (cascades)

Bepaalde keuzes hebben blijvende gevolgen die later in het verhaal terugkomen. Deze cascades bepalen uiteindelijk hoe de twee verhaallijnen elkaar kruisen en hoe het conflict wordt opgelost.

4. Eindoverzicht

Aan het einde van het verhaal krijgt de speler een overzicht van alle gemaakte keuzes. Dit maakt zichtbaar hoe beide perspectieven elkaar hebben beïnvloed en hoe de speler het conflict heeft vormgegeven.

Samenvatting

De verhaalstructuur is ontworpen om spelers telkens opnieuw te dwingen in de mindset van het actieve personage te stappen. Dit gebeurt door:

duidelijke, persoonlijke motivaties,
tegenstrijdige maar gelijkwaardige doelen,
contextuele en geloofwaardige keuzes,
reactieve NPC's,
omgeving die het perspectief versterkt,
cascades, callbacks en heuristics die samen impact creëren.

Op deze manier ondersteunt de verhaalstructuur direct de onderzoeksvraag: *Welke ontwerpstrategieën kunnen ervoor zorgen dat een speler die met twee tegenstrijdige personages speelt, wordt gestimuleerd om steeds rolconsistente keuzes te maken?*

Sectie 8: Verhaal

Wereldopbouw

Het verhaal speelt zich af in een fantasywereld met twee belangrijke steden: *Misthold* en *Sunhold*, en het dorp *Ra'ndom* dat tussen beide steden ligt. Deze geografische opzet sluit direct aan bij de onderzoeksfactor *omgevingsgestuurde rolversterking*: de omgeving beïnvloedt hoe spelers hun personage interpreteren.

- Misthold is hiërarchisch, militair en gecontroleerd → dit ondersteunt Rens formele, plichtsgerichte perspectief.
- Sunhold is economisch stabiel en handelsgericht → dit ondersteunt Airins persoonlijke motivatie.
- Ra'ndom is een neutrale zone → dit vormt het spanningsveld waarin beide perspectieven botsen.

De rivaliteit tussen de twee merchant guilds weerspiegelt het principe van *gelijkwaardige perspectieven*: geen van beide steden is absoluut goed of slecht, wat aansluit bij moderne inzichten over branching narratives.

- Misthold: probeert meer opportuniteiten te verkrijgen met handel uit Sunhold maar de Sunhold merchant guild staat hun constant in de weg.
- Sunhold: blokkeert Misthold waar mogelijk om de handel in Sunhold en de omgeving onder controle te houden, maar dit zorgt ervoor dat Misthold zich aangevallen voelt en agressievere methodes begint te gebruiken.

De gestolen documenten vormen het centrale conflict, zoals beschreven in het literair onderzoek: een verhaal heeft een duidelijke drijvende kracht nodig om keuzes betekenisvol te maken. Dit conflict activeert de factor *narratieve motivatie*.

- Sunhold: wil de documenten verkrijgen om informatie te krijgen die kan helpen om Misthold effectiever te blokkeren.
- Misthold: wil de documenten terug zodat de handel niet in gevaar komt.

Personages

Zoals eerder vermeld, bestuurt de speler twee personages: *Airin* en *Ren*. Hun gedeelde afkomst uit Ra'ndom sluit aan bij het literaire principe van herkenbaarheid, terwijl hun uiteenlopende levenspaden aansluiten bij de state-of-the-artvoorbeelden zoals *NieR: Automata* en *Dragon Age*.

Airin

Airin is een dorpsjager die in Ra'ndom woont. Zijn dagelijkse taken, zijn band met de dorpsbewoners en zijn grootvader in Sunhold versterken de onderzoeksfactor *karakterkaders*: Airin heeft duidelijke, herkenbare eigenschappen die de speler helpen rolconform te handelen.

Beginsituatie

Airin ontvangt een brief van zijn grootvader waarin staat dat hij dringend naar Sunhold moet komen. Deze persoonlijke motivatie sluit aan bij de design value *twee tegenstrijdige maar gelijkwaardige perspectieven*: de speler begrijpt waarom Airin dit doel nastreeft, zelfs wanneer die later als Ren speelt.

Ren

Ren is lid van de Misthold Guard. Zijn achtergrond als soldaat en zijn respect voor orde en veiligheid sluiten aan bij de onderzoeksfactor *narratieve motivatie* en bij de state-of-the-artinspiratie uit *Final Fantasy IX* (sectie 2), waar beweging en omgeving het gedrag van de speler beïnvloeden.

Zijn band met Sunhold via zijn zus versterkt de factor *karakterkaders*: Ren is niet alleen een soldaat, maar heeft ook persoonlijke relaties die zijn perspectief nuanceren.

Beginsituatie

Ren wordt naar Ra'ndom gestuurd om gestolen documenten te onderscheppen. Deze opdracht activeert de design value *gelijkwaardige perspectieven*: Rens doel is rationeel en verdedigbaar, maar staat volledig haaks op Airins doel.

Verhaal in grote lijnen

Airin en Ren hebben beide een doel dat door omstandigheden rechtstreeks botst met dat van de ander. Dit conflict is ontworpen volgens de vijf onderzoeksfactoren:

- Narratieve motivatie*: Airin wil zijn grootvader helpen; Ren moet zijn plicht vervullen.
- Karakterkaders*: Airin is empathisch en dorpsgericht; Ren is formeel en plichtsgetrouw.
- Geloofwaardige keuzecontext*: beide personages krijgen keuzes die logisch zijn binnen hun perspectief.
- Omgevingsgestuurde rolversterking*: Airin start in een open dorp; Ren in een militaire kazerne.
- NPC-reactiviteit*: NPC's reageren anders op Airin dan op Ren, zoals getest in de prototypes (Sectie 6).

De situatie creëert een natuurlijke spanning:

- Als Ren Airin helpt*, riskeert hij zijn positie binnen de Misthold Guard.
- Als Airin Ren helpt*, riskeert hij te laat te komen om zijn grootvader te helpen.

Dit sluit aan bij de design value *keuze-echo's*: keuzes van het ene personage beïnvloeden het andere, zonder het verhaal te corrigeren.

Daarnaast weerspiegelt dit de inzichten uit de talks: keuzes moeten gelijkwaardig zijn, niet moralistisch, en moeten emotionele impact hebben.

Einde

Het verhaal heeft meerdere variaties, maar twee hoofdopties:

- Airin bereikt Sunhold*, maar Ren faalt in zijn opdracht.
- Ren onderschept de documenten*, maar Airin faalt om Sunhold te bereiken.

Deze eindstructuur sluit aan bij:

- Cascades*: grote keuzes leiden tot blijvende gevolgen.

Gelijkwaardige perspectieven : geen einde is absoluut goed of slecht.

State of the Art: vergelijkbaar met *NieR: Automata*, waar perspectieven botsen maar beide valide zijn.

Sectie 9: Technische aspecten

Van Twine naar Unity

Zoals uitgelegd in [Prototyping](#) heb ik gezocht naar een efficiënte manier om Twine samen met Unity te gebruiken. In het begin gebruikte ik daarvoor een asset, maar die was te beperkend. De asset gebruikte het `.twee`-bestand rechtstreeks en er kon geen gameplay tussen de `passages` worden toegevoegd zonder grote delen van de code aan te passen.

Daarom heb ik besloten de asset alleen als inspiratie te gebruiken en mijn eigen converter te maken. Voordat ik hieraan begon, had ik al een asset genaamd Datablocks aangepast zodat die weer met nieuwere versies van Unity werkte. Deze asset maakt het mogelijk om vooraf bepaalde geserialiseerde objecten te importeren en exporteren als `.json` en in andere formaten.

Met deze mogelijkheid om data snel in geserialiseerde objecten te importeren ben ik verder gaan werken.

1 Passage = 1 Datablock

Op deze manier behoud ik de Twine-structuur van object naar object. Ik hoef die structuur alleen in Twine aan te passen en kan ze daarna gewoon importeren.

Van `.twee` naar `.json` naar datablock

Dit is een aspect waar ik even mee vast heb gezeten. Een `.twee`-bestand kan niet rechtstreeks naar Datablock worden omgezet, omdat de structuur van `.twee` en Datablock niet goed op elkaar aansluit. `.json` werkt wel, omdat het met "naam van element => elementdata" werkt. Dat is makkelijk om te vormen tot een geserialiseerd object.

Daarom moest ik het `.twee`-bestand eerst op een specifieke manier omvormen tot een `.json`-bestand. Ik heb het mezelf daarbij niet moeilijk gemaakt en Copilot gevraagd een klein Python-script te maken dat de conversie uitvoerde.

Hiervoor had ik eerst een bepaalde syntaxis voor het `.twee`-bestand bepaald:

```

-- "name" [pov] {"position":"1800,600","size":"100,100" => not needed after parsing}
**POV: Airin**

passageText

--choice text 1|choice passage 1
--choice text 2|choice passage 2
--choice text 3|choice passage 3
--choice text 4|choice passage 4]]
...

--TWO: 1|choice 1
--choice text 1|choice passage 1 is an objective to complete before showing the passageText
--choice text 2|choice passage 2
...

[[LogMessage: message to show in log if this passage was passed(passageName)]]

--choice text
objective description__

```



Zolang deze syntaxis behouden bleef, kon ik het `.twee`-bestand makkelijk omzetten naar een `.json`-bestand dat er zo uitzag:

```

{
  "name": "passageName",
  "pov": "",
  "position": "1800,600",
  "size": "100,100",
  "choiceTexts": [
    "choice 1",
    "choice 2",
    "choice 3",
    "choice 4"
  ],
  "choicePassages": [
    "choice passage 1",
    "choice passage 2",
    "choice passage 3",
    "choice passage 4"
  ],
  "pov": "Airin",
  "objectiveText": "True/False",
  "objectiveDescription": "False",
  "objectiveTitle": "A",
  "objectiveMessage": "A1/A2/A3"
  "LogMessage": "Airin: went into town to gather information about travelling to Sunhold. (A1)"
},

```



Dit kan vervolgens in Unity worden geïmporteerd zonder dat de code aangepast moet worden om verhaalaanpassingen te verwerken.

UI Toolkit

UI Toolkit is voor sommige aspecten erg handig, maar door bugs problematisch voor andere. Ik heb UI Toolkit gebruikt omdat het me de mogelijkheid gaf om heel direct te controleren waar bepaalde elementen op het scherm moesten staan.

Het heeft Unity ook een paar keer laten crashen omdat de "image slice"-slider negatief of nul kan worden en Unity dit niet aankan. Unity komt dan vast te zitten in een oneindige repaint-loop waar je niet uit raakt zonder Unity handmatig af te sluiten via Taakbeheer. Ook heb ik geleerd dat UI Toolkit soms de standaardlayout blijft gebruiken, ook al verander je die specifieke layout in het USS-bestand. Dan moet je benadrukken dat je de standaardinstelling overschrijft, zoals in dit voorbeeld:

`.unity-button.unity-button: hover` . De eenvoudige vorm (`.unity-button: hover`) werkte in dit specifieke bestand niet, terwijl ze in alle andere bestanden wel werkte.

Sectie 10: Toekomstvisie voor de master

Mijn toekomstvisie voor dit project tijdens mijn masterjaar is om verschillende onderdelen verder te verfijnen en zo de totale spelervaring te versterken.

Dit omvat:

- spelers de mogelijkheid geven om Ren al aan het begin te kiezen, waarna zij met andere keuzes en gevolgen geconfronteerd worden
- de huidige AI-assets vervangen door eigen visuele assets
- muziek toevoegen die het verhaal en de sfeer ondersteunt
- de UI verder uitbreiden
- objectieven toevoegen voor de rest van het spel
 - passages die nog geen objectief hebben aanvullen
 - meer variatie aanbrengen in het type objectieven
- een betere transitie ontwikkelen bij het wisselen van perspectief
- een korte introfilm maken in plaats van een tekstblok aan het begin, zodat de sfeer wordt neergezet zonder dat de speler een lange tekst moet lezen.

De code en het verhaal zijn grotendeels op hun plaats en zullen normaal gezien slechts minimale aanpassingen nodig hebben. Het systeem om geluiden af te spelen bestaat al, maar bevat momenteel nog geen audiobestanden.

Conclusie

In deze thesis heb ik onderzocht hoe je een speler twee tegenstrijdige perspectieven kunt laten beleven binnen eenzelfde verhaal. Ik heb gekeken naar bestaande spellen en naar technieken voor personagecreatie en worldbuilding.

Uit de playtests blijkt dat elementen zoals UI, layout en kleur een groot effect hebben op hoe spelers bepaalde perspectieven ervaren. Mijn prototypes tonen aan dat spelers een verhaal interessant kunnen vinden, maar verward kunnen raken door de manier waarop de UI wordt gebruikt.

Door duidelijke verschillen te creëren in de UI voor de personages, versterk je hun verschillende sferen en doelen. Er moet wel een algemene layout voor beide personages bestaan, zodat de speler niet bij elke perspectiefwissel een nieuwe layout moet aanleren en eraan moet wennen.

Het doel van het project is bereikt: spelers krijgen twee tegenstrijdige perspectieven te zien die verschillend aanvoelen en waarvan ze het gevoel krijgen dat beide perspectieven bestaansrecht hebben.

Verdere stappen zijn het uitwerken van details in het spel en het verfijnen van UI- en verhaalelementen, onder andere door objectieven aan de speler toe te voegen.

References

How Not to Write a Novel: 200 Classic Mistakes and How to Avoid Them—A Misstep-by-Misstep Guide [\[link\]](#)
Mittelmark, H. and Newman, S., 2009.

Video Game Storytelling, What Every Developer Needs to Know About Narrative Techniques [\[link\]](#)
Skolnick, E., 2014.

Integrated Storytelling by design [\[link\]](#)
Paulsen, K.S., 2022.

Footnotes

Nier Replicant, door Square Enix. Game link: https://www.square-enix-games.com/en_EU/games/nier-replicant-ver-1-22474487139 [↗]

Nier Automata, door Square Enix. Game link: https://www.square-enix-games.com/en_EU/games/nier-automata [↗]

Baldur's Gate 3, door Larian Studios. Game link: <https://baldursgate3.game/> [↗]

Dragon Age: Origins, door BioWare. Game link: <https://www.ea.com/games/dragon-age/dragon-age-origins> [↗]

Dragon Age: Inquisition, door BioWare. Game link: <https://www.ea.com/games/dragon-age/dragon-age-origins> [↗]

Gamedec, door Anshar Studios. Game link: <https://gamedec.com/> [↗]

Final Fantasy IX, door Square Enix. Game link: https://www.square-enix-games.com/en_EU/games/final-fantasy-ix [↗]

Canvas of Kings, door Might of Merchants. Link: <https://mightofmerchants.com/> [↗]

Appendix

Acknowledgements

Gebruikte softwares:

Twine voor keuzes testen

Unity voor spel

Obsidian voor het bijhouden van informatie

Third Party Assets:

Tools:

[Datablock unity asset](#), van [Peter Simard](#)

[Twine to Unity](#), van [bigalligator](#) als inspiratie

[Reve image making ai](#), voor tijdelijke placeholders

Models:

[Free Medieval Houses 3D Low Poly Models](#), van de groep [craftpix.net](#)

[Free Medieval 3D People Low Poly Pack](#), van de groep [craftpix.net](#)

[Free Environment Props 3D Low Poly Pack](#), van de groep [craftpix.net](#)

[Free Medieval Props 3D Low Poly Models](#), van de groep [craftpix.net](#)

[Low Poly Stylized Nature](#), van de studio [Cody Dream Studios](#)

Skybox

[Fantasy Skybox FREE](#), van [Render Knight](#)

UI

[kenney_emotes-pack](#), van de groep [Kenney](#)

[kenney_ui-pack-adventure](#), van de groep [Kenney](#)

Music

[Music for your rpg](#), van [The sound guild](#)

[RPG Music Pack: Starter Pack II](#), van [Owl Theory Music](#)

Appendix One

Datablocks

Om mijn spel te kunnen structureren heb ik gebruik gemaakt van een Asset genaamd Datablocks. De Asset had al 10 jaar geen updates gehad maar deed wel exact wat ik nodig had. Om deze reden heb ik er eerst voor gezorgd dat de asset weer bruikbaar was en ook een paar extra functionaliteiten toegevoegd.

Functionaliteiten

De Datablock asset heeft een paar handige functionaliteiten die helpen met structuur en het snel toevoegen en aanpassen van data.

Categorieën

Categorieën geven de mogelijkheid om verschillende datablocks in een categorie te zetten, zodat je ze makkelijk kunt opzoeken in de datablockmanager. Een voorbeeld hiervan is dat ik voor de quests twee verschillende datablocks had, de QuestDataBlock en de QuestStepDatablock. Beide kan ik dan de categorie "quest" geven, waardoor ik makkelijk alles wat aan quests gelinkt is kan opzoeken en aanpassen.

Fields

De fields zijn de effectieve data die wordt opgeslagen. Deze kunnen ook verwijzen naar een andere Datablock.

In editor aanpassingen

Als je de datablocks in de Unity-editor wilt aanpassen zonder een JSON- of ander soort bestand te importeren, is het mogelijk om bepaalde informatie alleen te tonen als aan bepaalde voorwaarden is voldaan. Dit is waar DatablockEnum en DatablockBool van pas komen. Beide tonen alleen de fields die eronder staan als ze aan hun waarde voldoen.

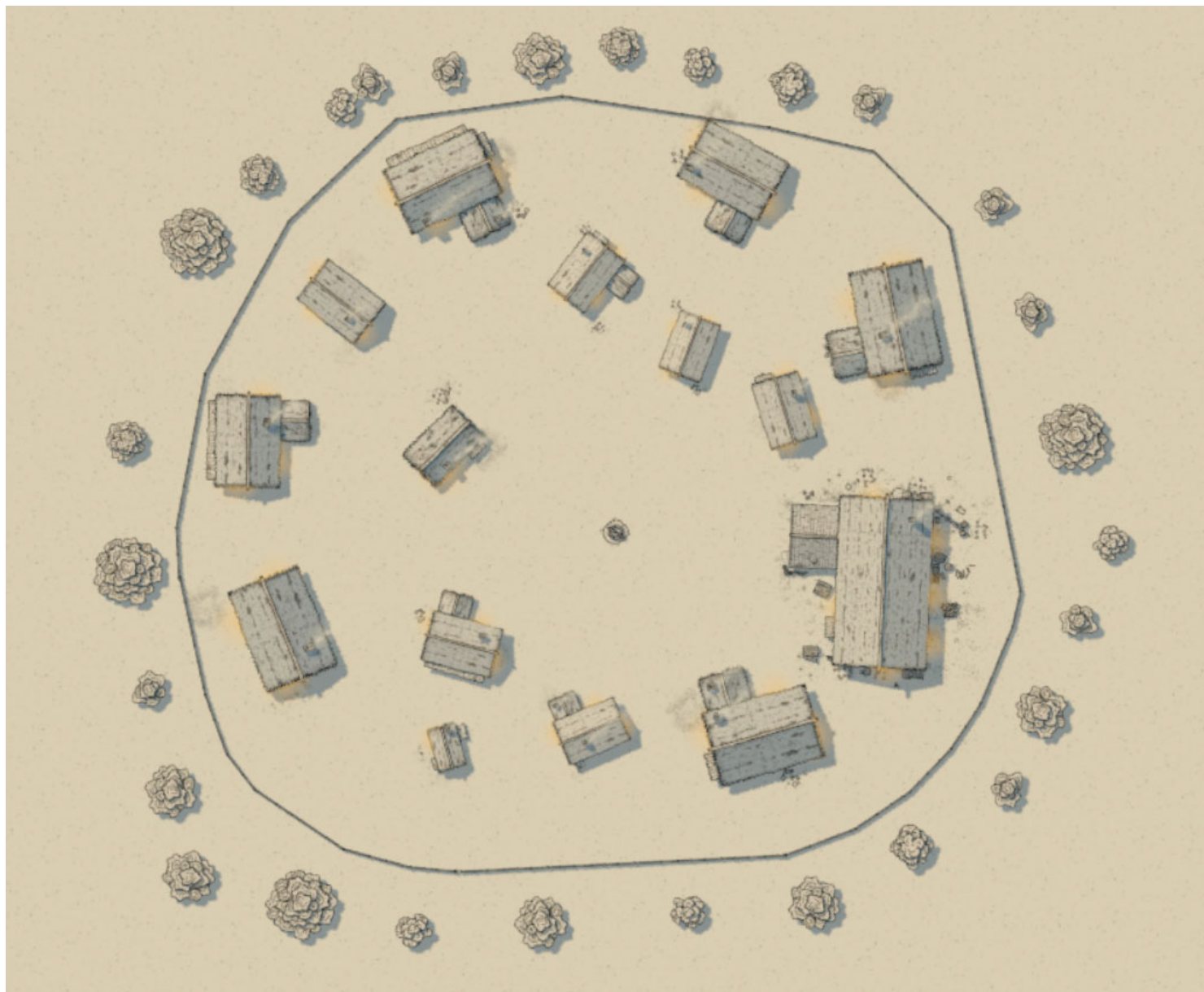
Import/Export

Voor het makkelijk en snel toevoegen van data is er een importfunctie die data kan importeren uit een JSON-, XML- of spreadsheetbestand. Op deze manier kun je snel een groot aantal datablocks toevoegen. Als je bestaande datablocks wilt aanpassen, kan dit ook door ze eerst te exporteren, het document aan te passen en het daarna weer te importeren.

Locaties

Voor het bijhouden van de layout van de locaties heb ik snelle kaarten gemaakt. Deze heb ik snel kunnen maken met de app Canvas of Kings.⁸

Ra'ndom



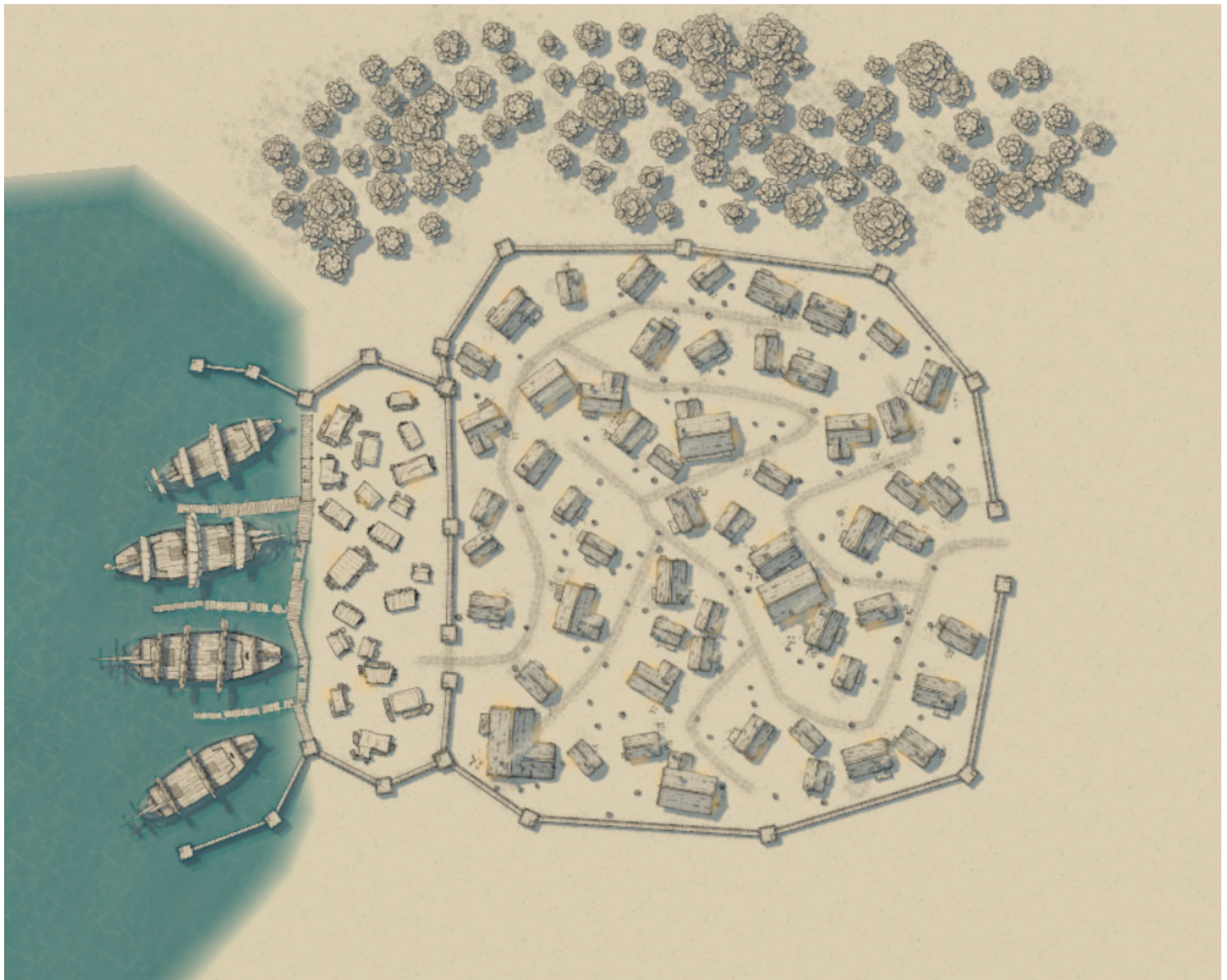
Ra'ndom map

Sunhold



Sunhold map

Misthold



Misthold map