



Unity Developer Flotsam

Plan van aanpak

**Bachelor in Toegepaste Informatica -
Application Development Werktraject**

Johan Torfs

Academiejaar 2022-2023

Campus Geel, Kleinhoefstraat 4, BE-2440 Geel

INHOUDSTAFEL

1	AANLEIDING EN ACHTERGROND	3
1.1	Pajama Llama Games	3
1.2	Flotsam	3
1.2.1	Unieke drifters	3
1.2.2	Verschillende locaties.....	3
1.2.3	Kleurrijk	3
1.2.4	Recycleer en onderzoek	4
2	VERWACHT RESULTAAT.....	5
2.1	De stageopdracht.....	5
2.2	Actuele status van het spel	7
2.3	Functionele vereisten.....	7
2.3.1	Upgraden van walkways.....	7
2.3.2	Kunnen verplaatsen van reeds bestaande gebouwen	7
2.4	Niet-functionele vereisten.....	8
2.5	Semantische code en duidelijke commits	8
3	BUSSINESS CASE EN DOELGROEPEN	9
4	PLANNING.....	10
4.1	Kennismaking met codebase.....	10
4.2	Oplossen kleinere bugs	10
4.3	Upgraden van 'walkways'	10
4.4	'Build overlay' & verplaatsen van gebouwen.....	11
4.5	Monitoring en feedback.....	12
5	RISICOANALYSE EN PROJECTAFBAKENING.....	13
5.1	Conflicten met andere features	13
5.2	Afbakening van de scope	13
5.3	Legacy code	13
6	INFORMATIEVERZAMELING EN RAPPORTERING	14
6.1	Mondelinge rapportering.....	14
6.2	Notion	14

1 AANLEIDING EN ACHTERGROND

Dit eerste hoofdstuk geeft kort de achtergrond weer van mijn stagebedrijf, Pajama Llama Games. Ook geeft het een korte uitleg over hun bekroonde spel, Flotsam, een post-apocalyptisch stadssimulatiespel waarin spelers de uitdaging aangaan om een drijvende nederzetting te bouwen en te overleven in een wereld overspoeld door water.

1.1 Pajama Llama Games

Pajama Llama Games is een klein, maar ambitieus game development bedrijf, gevestigd in centrum Gent. Het bedrijf heeft een echte passie voor games en focust zich sterk op creativiteit en innovatie. Mede door een toegewijd team van ontwikkelaars, heeft Pajama Llama Games zich gespecialiseerd in het creëren van unieke en meeslepende game-ervaringen. Door zijn uniek karakter, heeft het bedrijf zich inmiddels mee op de kaart gezet van de Belgische game-industrie

1.2 Flotsam

Flotsam is hét toonaangevend project van Pajama Llama Games. Het spel is uitgebracht in september 2019. Een spel speelt zich af in een post-acopalyptische wereld, overspoeld met water. Het is aan de speler om het ronddrijvende kostbare afval te verzamelen en dit te hergebruiken in de bouw van een vuilnisstad. Alleen zo kan een speler overleven in deze post-acopalyptische wereld.

Een speler start met enkel een 'Townheart', een boot, die hij gebruikt als basis voor zijn vuilnisstad. Op de 'townheart' leeft een 'townbuilder'. De townbuilder bezeilt de oceanen met zijn 'townheart', terwijl hij zoekt naar voedsel, drinkbaar water en recycleerbaar afval. Hiermee kan hij zijn stad verder uitbreiden. Hoe groter de stad, hoe meer verdrenkelingen ('drifters') een townbuilder kan redden van omliggende eilanden.

Het spel Flotsam heeft veel lof ontvangen van zowel gamers als critici vanwege zijn charmante visuele stijl, inventieve gameplay-mechanieken en het vermogen om een gevoel van avontuur en ontdekking op te roepen. Het spel biedt spelers de vrijheid om hun eigen unieke drijvende samenleving te bouwen, terwijl ze de overblijfselen van de beschaving verkennen en waardevolle materialen uit de zee verzamelen.

1.2.1 Unieke drifters

Drifters zijn de overlevenden in de wereld van Flotsam. Ze zijn eigenzinnig en hebben dingen die ze leuk vinden of juist niet. Dit creëert een uniek spel waar de speler rekening mee moet houden. Iedere drifter die hij toevoegt aan zijn stad heeft zijn unieke vaardigheden en eigen voorkeuren. Dit beïnvloedt zo ook de stad zelf.

1.2.2 Verschillende locaties

In de wereld van Flotsam valt er veel te ontdekken. In de oceaan drijven verschillende materialen, zoals plastic en drijfhout. Daarnaast kan een speler ook kleine eilanden en ruïnes van de vergane samenleving ontdekken. Op deze locaties zijn nieuwe drifters of materialen te vinden.

1.2.3 Kleurrijk

De wereld van Flotsam zit vol met eigenzinnige karakters. De kleurrijke, gestileerde kunststijl brengt de wereld tot leven.

1.2.4 Recycleer en onderzoek

Flotsam is gebouwd op het principe van recyclage en onderzoek. Door verschillende technologieën te onderzoeken, zullen de drifters een beter leven kunnen opbouwen. De bronnen die een speler doorheen de wereld vindt, zijn al dan niet recycleerbare materialen. Om zijn stad uit te breiden, moet een speler de kostbare afvalmaterialen recyclen. Deze kan hij dan als bouw materiaal gebruiken voor de aanbouw van zijn vuilnisstad.

2 VERWACHT RESULTAAT

Het doel van deze opdracht is om het spel Flotsam aan te passen, zodat het bouwen van nieuwe gebouwen andere stadsdelen zeer vlot en intuïtief aanvoelt voor een speler. Zo wil het bedrijf het spelen van Flotsam nog aangenamer maken voor de spelers.

Concreet houdt dit in dat ik mee C# scripts zal schrijven om de functionaliteiten binnen het spel mogelijk te maken. Dit gaat van het interactief maken van een knop tot het simuleren van complex gedrag van de drifters.

Mijn stage zal zich echter vooral focussen op het uitbreiden en verbeteren van het constructie-systeem. Dit is het systeem dat de spelers toelaat om gebouwen aan hun stad toe te voegen of om de gebouwen terug te verwijderen uit hun stad.

2.1 De stageopdracht

Mijn opdracht binnen het bedrijf is om mee te draaien in de ontwikkeling en afwerking van Flotsam. Meer specifiek is mijn opdracht om problemen met het constructie-systeem van het spel te onderzoeken en op te lossen. Daarnaast wordt er ook van mij verwacht dat ik andere gevraagde features en verbeteringen implementeer.

Aangezien Flotsam een zeer uitgebreid spel is, kan de uitgebreide beschrijving gevonden worden op de [website van Pajama Llama Games](https://www.pajamallama.com) of op de [Steam¹ pagina van Flotsam](https://store.steampowered.com/app/821250/Flotsam/).

- www.pajamallama.com
- <https://store.steampowered.com/app/821250/Flotsam/>

Hieronder geef ik een korte beschrijving die meer gericht is op mijn stageopdracht.

2.1.1 Speluitleg Flotsam

In Flotsam begint de speler in het midden van de oceaan op een klein schip, de 'Townheart' genoemd. De Townheart is het centrum van de drijvende stad die de speler zal kunnen bouwen.

Op de Townheart wonen 3 'Drifters'. 'Drifters' zijn de bewoners van deze wereld. Ze kunnen verschillende taken uitvoeren, zoals het verzamelen van materialen uit de zee, het produceren van nieuwe materialen en het bouwen van gebouwen. De speler bepaalt welke welke taken een drifter moet uitvoeren. Vervolgens zullen de 'Drifters' deze taken automatisch uitvoeren.

Om een taak te starten, gebruikt de speler de knoppen op de UI². Zo kan de speler bijvoorbeeld aangeven dat de drifters plastic afval moeten verzamelen. Dit doet hij door op de 'Buoy Marker'-knop te drukken onderaan het scherm. Vervolgens plaatst de speler een boei op een plaats met plastic afval. De 'Drifters' herkennen deze boei, zwemmen er naartoe en starten met het verzamelen van het plastic afval. Tot slot nemen de drifters het materiaal mee naar de Townheart. In de onderstaande afbeelding

¹ Steam is een populair digitaal distributieplatform voor computerspellen, waar gamers games kunnen kopen, downloaden en spelen, en waar ze ook kunnen communiceren met andere spelers en deelnemen aan gemeenschappen rondom verschillende spellen.

² User interface. Dit is de verzameling van alle elementen, waarmee de speler kan interageren om acties uit te voeren of informatie te vergaren.

wordt dit proces weergegeven. De speler volgt de 'Drifter' met de camera om het proces duidelijk te maken, maar dit heeft geen invloed op het gedrag van de 'Drifter'. De 'Drifter' zou deze taak uitvoeren, onafhankelijk of de speler hem met de camera volgt of niet.



Afbeelding 1: Verzamelen van plastic afval.

Een ander voorbeeld van een taak die de speler kan toewijzen, is het bouwen van een gebouw. De speler opent het bouw menu, daarna selecteert hij een gebouw en vervolgens plaatst hij een preview van het gebouw. Daarna zal een 'Drifter' de nodige materialen naar het gebouw brengen en het gebouw beginnen bouwen.



Afbeelding 2: Het bouwen van een gebouw.

Dit zijn maar 2 voorbeelden van acties die de speler kan ondernemen. Deze acties vormen de basis van het spel. De speler geeft telkens een opdracht en die opdracht

wordt dan uitgevoerd door de 'Drifters'. Natuurlijk zijn er veel meer mogelijke taken die de speler kan toewijzen, maar aangezien mijn stageopdracht zich focust op het constructie-systeem, beperk ik mij ook tot deze 2 voorbeelden.

2.2 Actuele status van het spel

Momenteel is Flotsam nog in 'Early Access'. Dit betekent dat spelers het spel al kunnen kopen en spelen, maar dat er ook nog veel veranderingen in het spel kunnen komen. Daarbij moet wel vermeld worden dat het spel al heel ver staat voor een 'Early Access'-spel.

Pajama Llama Games is van plan om in september 2023 de 'full release' te doen van Flotsam. Dit betekent dat het spel op dat moment volledig afgewerkt is. Dit betekent niet dat ze daarna geen updates of patches meer zullen uitvoeren, maar dit geeft aan de spelers wel aan dat er geen grote veranderingen meer zullen gebeuren.

2.3 Functionele vereisten

De functionele vereisten kunnen worden opgesplitst in 2 grote delen:

- 1) Het upgraden van de 'walkways'
- 2) Het kunnen verplaatsen van reeds bestaande gebouwen.

Daarnaast zal ik tijdens mijn stage ook bestaande bugs in het constructie-systeem oplossen naarmate ik ze tegenkom.

2.3.1 Upgraden van walkways

Momenteel kunnen de spelers in Flotsam verschillende 'Walkways' bouwen. Dit zijn wandelwegen over de oceaan. Afhankelijk van het gebruikte materiaal en van het stadium van upgrade, kan een walkway uit verschillende materialen bestaan.

De 4 verschillende upgrade-stadia van walkways zijn:

- 'Plastic walkways'
- 'Wood walkways' (+1 beauty)
- 'Tiled walkways' (+2 beauty)
- 'Grass walkways' (+2 beauty)

Deze walkways zien er telkens anders uit. Zo geven meer geupgradede walkways een hogere 'beauty-score' aan de stad. Deze score heeft op zijn beurt weer invloed op de moraal van de drifters.

Aan een walkway, kan een speler een gebouw bouwen. Aan de andere kant betekent dit dat de speler een 'walkway' enkel kan afbreken als er niets meer aan gebouwd is. Dit wil zeggen dat, om een walkway te veranderen, de speler eerst alles wat eraan vasthangt zal moeten afbreken. Dit is natuurlijk niet praktisch. Het is dus de bedoeling om het spel zo te hervormen dat de speler een walkway kan upgraden zonder eerst de aanliggende gebouwen te moeten afbreken.

2.3.2 Kunnen verplaatsen van reeds bestaande gebouwen

De stad die de speler bouwt, groeit meestal redelijk organisch. Echter, naarmate dat het spel vordert, onderzoeken spelers nieuwe technologieën en kunnen ze dus ook nieuwe gebouwen bouwen. Dit leidt tot situaties waarin de speler de stad volledig wilt herbouwen. Alle gebouwen en stadsdelen afbreken en dan terug opbouwen is echter

tijdrovend en vervelend voor een speler. Vooral omdat onder andere betekent dat bepaalde gebouwen niet beschikbaar zullen zijn op het moment dat de speler het wenst.

De oplossing hiervoor is een systeem opzetten, waarmee de speler reeds geplaatste gebouwen makkelijk kan verplaatsten. Zo moet hij de gebouwen niet eerst afbreken om dan op een andere plaats terug op te bouwen. Het systeem kan best de vorm aannemen van een 'Overlay', zodat de speler duidelijk kan zien dat ze in een andere mode zitten waarin ze gebouwen kunnen verplaatsen.

2.4 Niet-functionele vereisten

Deze vereisten gelden voor het hele constructie-systeem. Het bouwen binnen het spel moet vlot, intuïtief en gebruiksvriendelijk aanvoelen. Dit betekent dat het voor de spelers makkelijk moet zijn om te achterhalen hoe ze een actie moeten uitvoeren. Het is niet leuk voor de speler als die eerst pagina's instructies moet lezen voordat hij in staat is om een gebouw te bouwen of om een gebouw te verplaatsen.

2.5 Semantische code en duidelijke commits

Echte documentatie bij de code wordt er niet geschreven binnen het bedrijf. Wel is het belangrijk dat alle variabelen, functies en klassen duidelijke namen hebben. In het uitzonderlijke geval dat dit toch niet mogelijk is, of dat er een stuk code tussen zit dat een zeer specifiek issue oplost, kan een comment worden bijgeschreven. Dit hoort echter altijd een uitzondering te blijven. De belangrijkste afspraak is om code te schrijven die zonder commentaar of documentatie leesbaar en duidelijk is.

Hetzelfde geldt voor de commits, die gebeuren via SourceTree. Het is belangrijk om de commits logisch te groeperen en duidelijk uit te leggen wat er in die commit bijgekomen en/of aangepast is. Zo kan de rest van het team gemakkelijk opzoeken wanneer en waarom iets is veranderd.

3 BUSSINESS CASE EN DOELGROEPEN

De doelgroep voor dit project zijn de spelers van Flotsam en hopelijk een heel aantal nieuwe spelers die als gevolg van de update(s) het spel ook gaan aankopen. Spelers zijn het bloed dat door de aderen van een game studio stroomt. Een spel wordt steeds ontwikkeld vóór de spelers, is gedreven door feedback en vraag van de spelers en het is betaald door de spelers. Het spreekt voor zich dat een spel geen succes kan hebben zonder spelers.

Daarom dat de business case voor dit project zo voor de hand liggend is. De uitbreidingen die ik ga maken:

1. zijn gevraagd door de spelers
2. zullen het spel aangenamer maken en dus een meerwaarde bieden aan de speelervaring.
3. verhogen de waarde van het spel. Dit wegens de punten hierboven.
4. Zorgen voor een hogere verkoop.

Aangezien de verbeteringen van Flotsam rechtstreeks gecorreleerd zijn met de verkoop van het spel en dus ook met het aantal spelers en het inkomen van Panama Llama Games, kan ik dus alleen maar benadrukken dat dit project een meerwaarde zal zijn voor de business van het bedrijf.

4 PLANNING

De planning voor mijn stage kan worden verdeeld in 4 verschillende fases.

- Kennismaking van de codebase
- Oplossen van kleinere bugs
- Upgraden van 'walkways'
- 'Build overlay' & verplaatsen van gebouwen

4.1 Kennismaking met codebase

In deze fase zal ik mij vooral bezighouden met het verkennen van het spel en de codebase. Hier zal ik voor mezelf vragen beantwoorden zoals:

- Welke acties kan een speler uitvoeren in het spel?
- Hoe wordt de input van de gebruiker omgezet in functionaliteiten in het spel?
- Welke scripts zorgen voor het gedrag van de verschillende gebouwen?
- ...

Eens dat ik een redelijk begrip heb van hoe verschillende scripts samenwerken om functionaliteiten te verwezenlijken, kan ik beginnen met de volgende fase.

De planning voor deze fase is een 2-tal weken.

4.2 Oplossen kleinere bugs

Er zijn binnen het constructie-systeem een aantal gekende bugs. Daarbovenop zijn er ook een aantal bugs die misschien nog niet gekend zijn.

In deze fase zal ik enerzijds zoeken naar bugs om op te lossen en anderzijds zal ik de gevonden bugs samen met de reeds gekende bugs trachten weg te werken.

Deze fase kan zeer snel gaan, maar evengoed heel lang duren. Dit komt door de aard van de opdracht in deze fase. Er moet onderzoek gedaan worden naar de verschillende bugs en hoe die op te lossen. Sommige bugs kunnen zeer oppervlakkig zijn, terwijl andere een moeilijk op te lossen oorzaak kunnen hebben.

Om toch een inschatting te maken voor de duur van deze fase, denk ik dat dit ongeveer 4-tal weken in beslag zal nemen.

4.3 Upgraden van 'walkways'

In deze fase zal ik zorgen dat de 'walkways' in de stad upgradebaar zijn. Hiervoor begin ik met na te kijken hoe de 'walkways' momenteel gebouwd worden. Daarna kijk ik ook na hoe gewone gebouwen geüpgraded worden. De bouw van walkways is namelijk ingewikkelder dan het bouwen van gewone gebouwen. Op basis van het systeem dat momenteel wordt toegepast voor de upgrade van gewone gebouwen kan ik een systeem bedenken om de 'walkways' te kunnen upgraden.

Vervolgens schrijf ik de code die het upgraden van de 'walkways' mogelijk maakt, ik voeg assets toe als die nodig zijn en maak bepaalde aanpassingen aan de instellingen van de 'walkways' waar nodig.

Ten slotte test ik de functionaliteit en kijk ik na dat mijn aanpassingen geen negatieve invloed hebben gehad op andere functionaliteiten. Ik kijk ook na of er geen problemen zijn met het inladen van oude 'save games'.

Ik schat dat ook deze fase een 4-tal weken werk zal vragen.

4.4 'Build overlay' & verplaatsen van gebouwen

In deze fase bouw ik een nieuw systeem, waarmee spelers gebouwen kunnen verplaatsen. Dit is een redelijk grote feature om toe te voegen en is daarmee ook ineens het grootste focuspunt van mijn stage. Gezien de grootte, zal ik deze fase dan ook opdelen in verschillende stappen.

Eerst zal ik onderzoek doen naar de werking van de verschillende 'CursorProperties', aangezien het zeer waarschijnlijk nodig is om een nieuwe klasse te maken die hiervan overerft. Ik zal ook onderzoeken hoe de 'CursorProperties' op een correcte manier te activeren.

Daarna zal ik de nodige 'CursorProperties' klasse aanmaken en een UI element toevoegen om de 'CursorProperties' te activeren.

Vervolgens zal ik de verschillende nodige functies uitschrijven:

- Het kunnen oppakken van een gebouw door erop te klikken. Dit betekent dat het gebouw de cursor zal volgen.
- Het kunnen neerplaatsen van een gebouw op een plaats waar het terug vasthangt aan een 'Hookable' ('walkway', ponton or Townheart) en waar het niet overlapt met een ander gebouw. Het gebouw moet zich daarbij ook binnen de grenzen van de stad bevinden.
- Het kunnen droppen van het gebouw in de 'Buildmenu', zodat het tijdelijk is opgeslagen en later terug geplaatst kan worden.

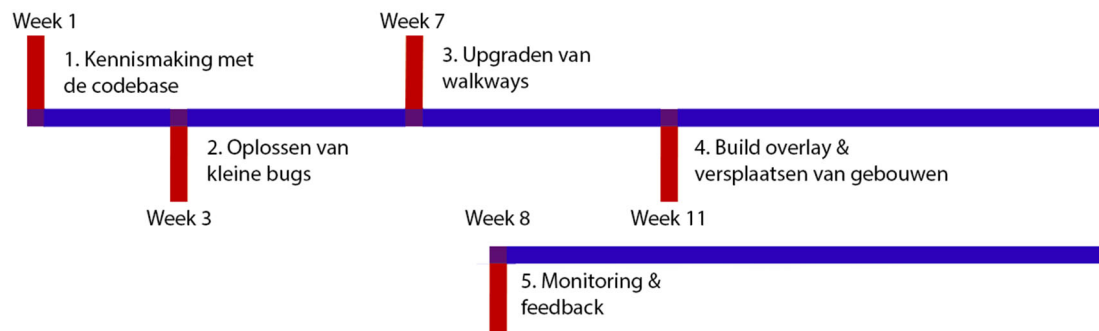
Ten slotte zal ik ook hier alle verschillende functies testen en nagaan of er geen andere problemen bij zijn bovengekomen.

Deze fase bevat meer complexe features dan de vorige fases. Ik schat voor deze toepassing dat ik een 9-tal weken nodig zal hebben.

4.5 Monitoring en feedback

Deze fase begint in principe nadat de eerste feature of bug fix live is gezet. Vanaf dat een speler met de aangepaste functionaliteiten heeft kunnen spelen, zal ik de feedback van de speler over deze feature verwerken en analyseren. Indien nodig zal ik nog aanpassingen maken aan de geïmplementeerde features.

Deze fase loop vanaf het live zetten van de eerste feature tot het einde van de stage, maar zal gelijklopen met fase 3 en 4.



Afbeelding 3: Tijdslijn planning stage

5 RISICOANALYSE EN PROJECTAFBAKENING

Er zijn drie belangrijke risico's waar ik tijdens mijn stage rekening mee moet houden:

- 1) conflicten met andere features
- 2) een (on)duidelijke afbakening
- 3) de aanwezigheid van legacy-code.

5.1 Conflicten met andere features

Een eerste risico is dat er tijdens de loop van het project nog andere dingen veranderen aan Flotsam. Dit is een gegeven, vermits het spel nog in ontwikkeling is. Er zullen dus 100% zeker aanpassingen gebeuren tijdens mijn stage. De kans dat deze aanpassingen invloed zullen hebben op dit project is dus ook redelijk groot. Het risico bestaat er dus in dat de aanpassingen aan het spel en de aanpassingen binnen dit project met elkaar in conflict treden. Dit risico kan worden beperkt door een goede interne communicatie tussen de ontwikkelaars die aan de verschillende features werken.

Om de werken van alle aanpassingen vlot te doen lopen, wordt er gewerkt op verschillende git-branches en met version control. Als ik andere aanpassingen van het spel goed opvolg en regelmatig andere branches merge, zouden zich hier geen problemen mogen voordoen.

5.2 Afbakening van de scope

Het is zeer belangrijk dat ik samen met mijn stagementor vanaf het begin het project goed afbaken. Voor dit project is de afbakening duidelijk. We beperken ons tot het uitbreiden en verbeteren van het constructie-systeem. Features die met andere zaken te maken hebben, vallen dus buiten de voorop afgesproken scope.

Binnen deze scope moeten we echter nog een aantal grenzen stellen. Zoals in de planning reeds aangehaald, beperken we ons in dit project tot bugfixes en het uitbouwen van een nieuwe feature, namelijk de 'Blueprint Overlay'. Alle andere mogelijke features vallen initieel buiten deze scope.

Extra features kunnen eventueel nog worden toegevoegd als ze relevant blijken te zijn naar het doel van het project. Als dit het geval is, zullen deze eerst nog moeten worden geanalyseerd, voordat ze kunnen worden opgenomen in de scope van het project.

5.3 Legacy code

Een laatste risico waar rekening mee gehouden moet worden is de legacy code. De ontwikkeling van Flotsam loopt al een aantal jaren. Het is dus onvermijdelijk dat er in de codebase veel legacy code zit met zeer specifieke doeleinden.

Het is belangrijk om tijdens het ontwikkelen van nieuwe features rekening te houden met die legacy code om ervoor te zorgen dat nieuwe aanpassingen niet in conflict zullen treden met die oude code. Het kan dus bij momenten zijn dat er delen van die legacy code herschreven moeten worden om ervoor te zorgen dat een nieuwe feature geïmplementeerd kan worden, zonder een andere feature te breken.

6 INFORMATIEVERZAMELING EN RAPPORTERING

Rapportering zal enkel mondeling gebeuren. Gezien de grootte van mijn stagebedrijf en de hechte samenwerking met de rest van het team, is dit de makkelijkste en efficiëntste manier van rapportering.

Daarnaast zal het voor bepaalde delen van het project ook nodig zijn om informatie te verzamelen over reeds bestaande systemen of over bugs binnen het spel. Dit zal gebeuren via het projectmanagement systeem Notion.

6.1 Mondelinge rapportering

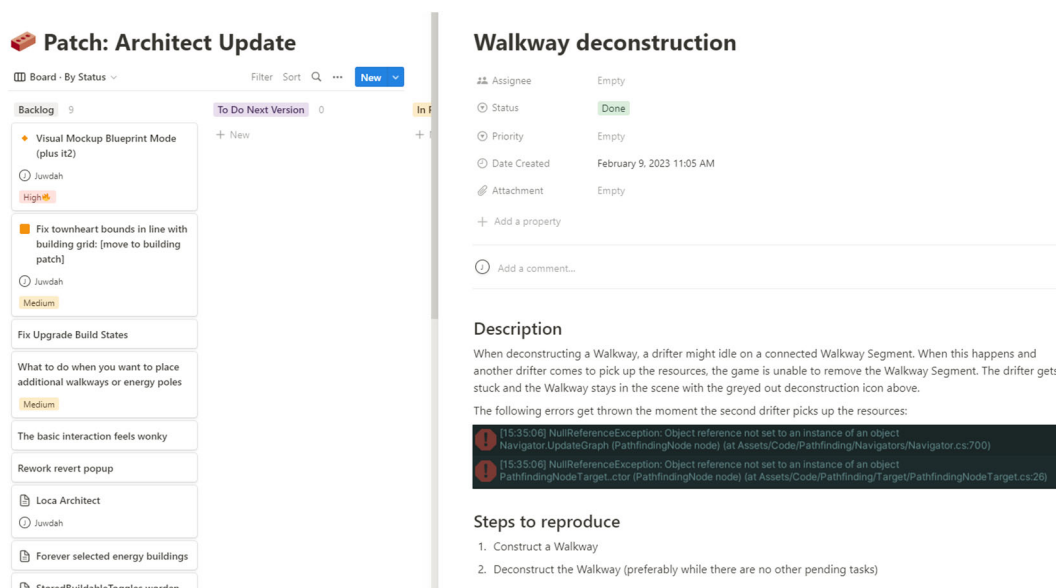
Iedere ochtend voorziet het bedrijf een stand-up. Dit is dan ook de voornaamste vorm van rapportering. Pajama Llama werkt in een klein team, waardoor iedereen op de hoogte blijft van wat er zich in het bedrijf afspeelt.

Elke dag dat ik werk neem ik dus ook mee deel aan de stand-up. Op dit moment rapporteer ik wat ik vorige keer heb gedaan en wat ik die dag wil verwezenlijken.

6.2 Notion

Naast rapporteren, zal het verzamelen van informatie ook cruciaal zijn bij sommige delen van mijn stage. Voor het eerste deel van mijn stage zal ik mij focussen op het oplossen van verschillende bugs in het spel. Deze zullen eerst en vooral getest en gereproduceerd moeten worden. Vooral voor dit deel zal Notion een belangrijke rol spelen.

Notion is een alles-in-één productiviteitsplatform dat gebruikers in staat stelt om notities te maken, projecten te beheren, samen te werken en informatie te organiseren op een flexibele en intuïtieve manier. Voor het doel van mijn stage gebruik ik vooral de 'Board'-pagina. Deze pagina laat de gebruiker toe om verschillende kaartjes aan te maken, die dan op basis van status in kolommen kunnen worden geplaatst.



Afbeelding 4: Board pagina in Notion met een open kaartje.

Iedere bug die is gerapporteerd, staat genoteerd in Notion. Bij een bug hoort een beschrijving van de bug en de stappen tot reproductie. Zo is alle informatie over de bug voor iedereen in het bedrijf beschikbaar.