

# Game Design Document

## Ogólne informacje

### Autorzy

- Kamil Ciągło
- Mateusz Balicki

### Tytuł

Augmented Unit

### Gatunki

Gra będzie pomieszaniem gatunków gier narracyjnych i metroidvanii, w postaci dwuwymiarowej platformówki z widokiem od boku. Skupimy się na narracji (bezpośredniej w postaci logów i środowiskowej). Z metroidvanii zapożyczony zostanie sposób postępu — cała mapa będzie w teorii dostępna od samego początku, jednak dostanie się do niektórych sekcji, może wymagać wcześniejszego zebrania modyfikacji. Modyfikacje te permanentnie odblokują dla gracza umiejętności, które mogą pomóc dostać się we wcześniej niedostępne miejsca, nawet te widziane kilka sekcji wcześniej.

### Odbiorcy

Docelową grupą odbiorców gry będą ludzie zaznajomieni z grami, w wieku od około 16 lat, zainteresowani historiami opowiedzianymi nie wprost, z nutką niepokoju.

### Plaforma i wymagania sprzętowe

Przewidywane wymagania sprzętowe, na podstawie wymagań podobnych tytułów:

Aspekt	Minimalne:	Zalecane:
OS	Windows 7+	Windows 7+
Procesor	Intel i5+	Intel i5+
Pamięć RAM	2 GB RAM	4 GB RAM
Karta graficzna	Nvidia 450 GTS / Radeon HD 5750 lub lepsza	Nvidia GTX 460 / Radeon HD 7800 lub lepsza
Miejsce na dysku	500 MB dostępnego miejsca	500 MB dostępnego miejsca
Dodatkowe uwagi	DirectX 9.1+ lub OpenGL 3.2+	DirectX 9.1+ lub OpenGL 3.2+

## Monetyzacja

Gra zostanie wypuszczona w modelu free-to-play (darmowa), aby zachęcić graczy do zagrania w nią i aby zdobyć odrobinę rozgłosu na rynku. Później mogą zostać wydane płatne dodatki do gry, które rozwiną ją o opcjonalne poziomy i poszerzą historię świata.

## Repozytorium

Aktualna wersja dokumentacji, jak i cały kod źródłowy gry są dostępne na GitHubie ([link do aktualnego GDD](#))

## Tematyka i osadzenie gry

### Lokacje

Gra rozgrywać się będzie na pokładzie wojskowego statku transportowego, którego załoga została wybita, a sam statek dryfuje uszkodzony na środku oceanu. Z uwagi na gatunek gry (metroidvania) cały statek będzie jedną, dużą mapą, ale jego konkretne sekcje będą reprezentować oddzielne poziomy.

Sekcje, które będziemy mogli znaleźć w grze:

Nazwa	Obowiązkowy	Modyfikacja	Logi	Opis
Pokład statku	Y	krzyk	Znikome	przeszkody w postaci pseudo-losowych uderzeń piorunów czy powiewów wiatru
Kajuty	Y	podwójny skok	DUŻO, przybliżające historię postaci	przejście między magazynem a laboratorium, w kajutach znajduje się "easter egg" w postaci zmienianej muzyki w grze
Magazyn	Y	zryw	Kilka, pozwalające zapoznać się z postaciami	początkowa sekcja, w niej znajdziemy przejścia do dodatkowych poziomów i kilka podstawowych informacji, oraz drogi blokowane przez kolce/pożar
Reaktor	Y	-	Kilka, pogłębienie problemów psychicznych postaci	ukryty koło ostatniej sekcji, pozwala włączyć zasilanie, co pozwoli uciec ze statku
Laboratorium	Y	DEM (Deus Ex Machina)	DUŻO, silniejsze podpowiedzi na pętłę czasu i możliwość wysadzenia statku	pokój z logami bezpieczeństwa, które mogą nakierować gracza na prawdziwe zakończenie gry, końcowa część gry

Nazwa	Obowiązkowy	Modyfikacja	Logi	Opis
Zbrojownia	N	-	Kilka, ostatni log Jacka i kilka o tym, po co nam broń nuklearna	w nim można aktywować bombę nuklearną, która rozpocznie odliczanie do samozniszczenia - alternatywne zakończenie
Pokój z łodzią podwodną (Submarine)	Y	-	Dwa, mówiące, po co łódź i że może przetrwać wybuch nuklearny	Dzięki łodzi można uciec ze statku po tym, jak uruchomimy bombę, z łodzi można skorzystać tylko, jeśli alarm został uruchomiony, ale potrzebuje ona zasilania
Pokój treningowy	Y	-	Jeden, przykład mechaniki	Pokój, do którego jesteśmy teleportowaniu w ramach samouczka

Fabula

Wprowadzenie

Gracz jest robotem (AU - Augmented Unit), który aktywuje się (wychodząc z kapsuły) w szczelnie zamkniętym pomieszczeniu. Dowiaduje się tutaj o swoim celu - zabezpieczeniu tajnych danych znajdujących się na statku. Po diagnozie systemów (tutorial) AU dostaje moduł odczytywania logów (narratora). Wyjście się otwiera i AU zostaje wypuszczony na statek, aby wypełnić swój cel.

Fabularne fakty

- Statek "Eterna"
  - dryfuje zniszczony na środku oceanu,
  - jest jednostką transportowo-badawczą,
  - przewozi broń nuklearną, tajne wojskowe dane oraz eksperymentalną broń, nad którą prowadzone są badania,
  - jest napędzany reaktorem, który zasila również jego systemy obronne
- AU,
  - AU — Augmented Unit,
  - AU jest robotem zdolnym do modyfikowania samego siebie, aby mógł wykonać swój cel,
  - celem AU jest odzyskiwanie i chronienie danych w sytuacjach kryzysowych,
  - AU jest zdolny do użycia DEM — eksperymentalnej broni, która pozwala na podróż w czasie,
    - przy wybraniu tej modyfikacji AU zostaje cofnięty w czasie i przestrzeni do momentu, gdy statek jeszcze funkcjonował,
    - na statku zostaje uruchomiony alarm i wszyscy próbują zatrzymać AU niosącego ze sobą DEM,
    - AU zabija wszystkich na statku i wraca do swojej kapsuły, gdzie zostaje zresetowany i może zacząć cykl na nowo,
  - AU po uruchomieniu musi przejść test systemów (tutorial) i otrzymać narratora,
  - AU jest w stanie dowiedzieć się, co stało się na statku poprzez logi i rozmowy z RP

- Postaci drugoplanowe,
  - Robot Porządkowy (RP)
    - RP to mały robot, który pojawia się kawałek za pokojem, w którym AU się budzi,
    - RP pomaga AU poprzez powiedzenie, że dane są w laboratorium,
    - RP będzie pojawiał się i oferował pomoc, gdy gracz spędzi za dużo czasu w jednym miejscu i nie będzie wiedział, co zrobić
  - Jan
    - Martwy żołnierz zaraz przed wejściem do pokoju startowego,
    - Jedyny nie walczący z AU, jest przerażony,
    - Jego log mówi o tym, że coś się zbliża, po czym słychać krzyk,
    - Nie ma większego znaczenia dla fabuły w późniejszych etapach
  - Robert (**autor logów**)
    - Naukowiec, twórca DEM
    - Odizolowany od ludzi wizjoner, pasjonat postępu, który pragnie tylko spokoju w swoim laboratorium,
    - Nielubiany przez resztę załogi z uwagi na jego obsesję na punkcie DEM,
    - Statek istnieje, aby umożliwić mu pracę, wszyscy obecni mieli go chronić,
    - Na początku uczy Szeregowego Jacka, jak działa jego praca, bo to jedyna osoba zainteresowana tym, co robi,
    - Współpraca z Szeregowym prowadzi do konfliktu między Robertem a Admiralem Jabłonowskim,
  - Admirał Jabłonowski (**autor logów**)
    - Główny dowódca wojsk,
    - Skupiony na pracy, stara się udawać, że nie ma życia poza statkiem, ale dba o swoich żołnierzy jak o rodzinę,
    - Zostaje zabity w reaktorze,
    - Wchodzi w konflikt z Robertem, gdyż nie podoba mu się to, że Szeregowy Jacek zaczyna się dziwnie zachowywać przez pomoc w pracy przy DEM,
    - Zabija Szeregowego Jacka, gdy ten próbuje wysadzić statek i nie chce odpuścić,
    - Zabicie Szeregowego Jacka odbiło się na jego psychice i zaczyna się po tym załamywać
  - Szeregowy Jacek (**autor logów**)
    - Zwykły, szary żołnierz,
    - Nie lubi być na tym statku jako żołnierz, więc zaczyna wchodzić w interakcję z Robertem w laboratorium,
    - Podczas jednego z eksperymentów uderza DEM młotkiem, bez zabezpieczeń, co wpływa na jego psychikę,
      - Jacek zaczyna widzieć swoje wspomnienia z różnych momentów w czasie, w tym te, których jeszcze nie przeżył,
      - Dobija go to i sprawia, że chce zniszczyć DEM,
    - Jacek w logach naprowadza gracza na możliwość wysadzenia statku i ucieczki łodzią podwodną,
    - Próbował sam odpalić bombę nuklearną, ale został zabity przez Admirala Jabłonowskiego po kilku ostrzeżeniach

Postaci drugoplanowe oznaczone jako "**autor logów**" będą miały własną kategorię logów. Będą oni przedstawiali historię opisaną wyżej w faktach z własnego punktu widzenia, w postaci nagrań głosowych. Kategorie logów będą potem wyświetlane w jednym z ekranów menu i będą mogły być odsłuchane

ponownie. Wszystkie logi, które nie będą pochodzić od trzech autorów logów, będą miały swoją osobną, wspólną kategorię. Spis logów znajduje się poniżej\*. Treść logów może jeszcze ulec zmianie.

*\*Logi są jedynie sposobem reprezentacji fabuły i wraz z rozwojem gry będą ulegały zmianie (w odróżnieniu od powyższych faktów i samej fabuły). Poniższa lista jest jedynie wstępnym pomysłem na większość z nich.*

Logi

Nazwa	Osoba	Id*	Lokacja	Log
Ostatnie słowa	Jan	Jan_1	Magazyn	Coś tu jest nie tak, coś jest nie tak! Coś pojawiło się na statku... O Boże... idźcie tu! To tu idźcie! Aaaaaaa!!!!
Po co to tu jest?	Robert	Robert_1	Magazyn	Augmented Unit... Nie rozumiem, po co to tu trzymamy... To tak jakby ludzie oczekiwali, że ta cała operacja nie wypali, a tak nie będzie! Ale jeśli już... to lepiej, żeby ten robot odzyskał wszystkie dane, moja praca nie może pójść na marne!
Ci ludzie...	Robert	Robert_2	Magazyn	To niewiarygodne jak dużo trzeba gadać z tymi ludźmi, zanim cokolwiek dostanę z magazynu... Czy oni nie rozumieją, że w mojej pracy CZAS JEST KLUCZOWY?!
Początek znajomości	Jacek	Jacek_1	Magazyn	Warta w magazynie nie jest taka zła, dzięki niej mogę czasem porozmawiać z Robertem. Chciałbym kiedyś pomóc mu z eksperymentami, zamiast tylko siedzieć w różnych częściach statku.
Podziw: bronie	Jabłonowski	Jab_1	Magazyn	Za każdym razem, gdy tu przychodzę, nie mogę się powstrzymać przed podziwianiem tej kolekcji broni... Ten statek to prawdziwe dzieło sztuki!
On musi z kimś porozmawiać	Jabłonowski	Jab_7	Magazyn	On oszalał! Wczoraj gadał jakieś głupoty, a teraz włamał się do zbrojowni! Może ktoś musi z nim porozmawiać...
Podziw: łódź podwodna	Jabłonowski	Jab_2	Submarine	Ta łódź... to najcudowniejsza defensywna technologia na tym statku... ponoć może nawet przetrwać wybuch nuklearny!
Na wszelki wypadek...	Jacek	Jacek_8	Submarine	Ok... Jacek... pamiętaj, łódź podwodna może być aktywowana TYLKO, jeśli alarm został już odpalony, a generator działa, pamiętaj, żeby wszystko przygotować, zanim coś zrobisz! Widziałeś, jak możesz skończyć!

Nazwa	Osoba	Id*	Lokacja	Log
Zrobiłem to...	Jabłonowski	Jab_8	Kajuty	Zrobiłem to... Ja... Już go nie ma... I nikt nie wie... Tyl... Tylko ja mam dostęp do zbrojowni i... ja... zostawiłem go tam... Boże! [płacz]
Waga naszej misji	Jabłonowski	Jab_3	Zbrojownia	Nasza misja jest ważna... to miejsce mi o tym przypomina... Jeden błąd, jeden atak i cały statek musi zostać wysadzony w powietrze! Tylko Admirał ma dostęp do tego pokoju, tylko Admirał ma kody... Tylko ja decyduję, co się stanie na wypadek ataku...
Ostatnie życzenie	Jacek	Jacek_9	Zbrojownia	To słowa pożegnalne... Jeśli ktoś czyta ten log, to znaczy, że znalazł go przy moim ciele i coś poszło nie tak... Proszę... upewnij się, że DEM zostanie zniszczony... To... to jest zbyt niebezpieczne, żeby ktokolwiek mógł tego używać...
Dlaczego statek?	Robert	Robert_3	Pokład	Czasami ludzie pytają się — czemu akurat statek? Czemu nie pracujesz na łodzi, czy w jakimś bunkrze? Odchodzę wtedy bez słowa... Oni nie potrafią pojąć, że pływając, dużo prościej jest pojąć naturę mojej pracy. Zrozumieć fenomen, który jest z nami non-stop, a którego nie dostrzegamy!
Ten szeregowy...	Robert	Robert_4	Kajuty	Ten szeregowy, jak on miał... Janek? Nie ważne! Ostatnio zaczął się sporo koło mnie kręcić, podpytuje o postęp prac, próbuje dowiedzieć się czegoś o DEM... Zabiera mi tylko czas! Chociaż... Wydaje się faktycznie zainteresowany, może się jeszcze do czegoś przydać.
Nazywa to DEM	Jacek	Jacek_2	Kajuty	Ostatnio wypytywałem Roberta o tę jego maszynkę... broń? Nie ważne! Nazywa to DEM, skrót od Deus Ex Machina, mówi, że to dlatego, że to tak jakby na nowo wynalazł czas i przestrzeń... Muszę dowiedzieć się więcej!
Może się przydać	Robert	Robert_5	Kajuty	Ten Janek zdaje się być coraz bardziej zainteresowany moimi badaniami... Kręci się przy laboratorium, dopytuje, przygląda się DEM, gdy jesteśmy w laboratorium... Powoli zaczynam się do niego przekonywać... może być dobrym pomocnikiem

Nazwa	Osoba	Id*	Lokacja	Log
W końcu jest ciekawie	Jacek	Jacek_3	Kajuty	Robert... cały czas myli moje imię, ale całkiem przyjemnie się z nim pracuje. Można dużo się od niego dowiedzieć, a chociaż nie muszę nudzić się na statku. Chciałbym kiedyś zrobić coś przy DEM!
Cieszę się, że jest szczęśliwy	Jabłonowski	Jab_4	Kajuty	Jacek ostatnio spędza bardzo dużo czasu z Robertem, cieszę się, że w końcu ma co robić, widać było, że nie przepada za wartami, a w ten sposób może chociaż uda im się skończyć te badania szybciej.
Zadziwia mnie	Robert	Robert_6	Laboratorium	Z każdym dniem Janek zadziwia mnie coraz bardziej! Dzisiaj próbował dotknąć DEM <b>GOŁYMI RĘKAMI!</b> Wie przecież, że DEM nie jest jeszcze stabilny... potem chciał sprawdzić, co się stanie, gdy spróbuje uderzyć to młotkiem! Powiedziałem mu, że jeśli chce cokolwiek zrobić tej broni, to potrzebowałby broni nuklearnej, a nie młotka!
To było dziwne	Jacek	Jacek_4	Laboratorium	Walnąłem dziś DEM młotkiem, chciałem sprawdzić, czy naprawdę jest tak wytrzymałe, jak Robert mówił... Po uderzeniu poczułem się dziwnie, a Robert zaczął krzyczeć, że nie powinienem się do tego zbliżać bez ochrony i mówił coś o broni nuklearnej.
Nie czuję się najlepiej	Jacek	Jacek_5	Kajuty	Moja głowa... Odkąd uderzyłem DEM, czuję się... dziwnie... jakbym nie był sobą, wszystko mi się miesza, wszystko wydaje się obce, ten statek nie jest już taki, jak pamiętałem!
Tym razem przesadził	Jabłonowski	Jab_5	Kajuty	Właśnie wróciłem z rozmowy z Robertem! Po ostatnim incydencie w laboratorium Jacek dziwnie się zachowuje... Zabroniłem mu już tam eksperymentować i dałem kilka dni na dojście do siebie, a Robert będzie już dużo dokładniej kontrolowany!
Za kogo on się uważa?!	Robert	Robert_7	Laboratorium	Za kogo on się uważa?! Ten Admirał twierdzi, że Janek ma problemy przeze mnie! Sam uderzył młotkiem w DEM, nie słuchał moich ostrzeżeń i to on sobie coś zrobił... Wróci do siebie za dzień czy dwa i powinien dalej mi pomagać, nic mu się nie stanie, jeśli będzie się słuchał!

Nazwa	Osoba	Id*	Lokacja	Log
Trzeba go ostrzec!	Jacek	Jacek_6	Kajuty	Nie mam już wstępu do laboratorium... Chciałem tylko pójść porozmawiać z Robertem... Widziałem dziwne rzeczy, DEM nie jest bezpieczny, nie powinniśmy dalej nad tym pracować!
Jest z nim gorzej	Jabłonowski	Jab_6	Laboratorium	Z Jackiem jest coraz gorzej! Usilnie próbuje dostać się do laboratorium, żeby porozmawiać z Robertem... A on cały czas tylko tam siedzi, jakby ostatni wypadek tylko bardziej go zmotywował...
Muszę to zbadać	Robert	Robert_8	Laboratorium	Udało mi się dzisiaj porozmawiać krótko z Jankiem w kajutach... Mówił o DEM, o tym, że od wypadku widzi dziwne rzeczy... Czy to możliwe, że DEM zaczyna działać, jak powinien? Muszę to sprawdzić!
Nie mogę tak dłużej	Jacek	Jacek_7	Kajuty	Robert nie zaniepokoił się tym, co mu powiedziałem... Tylko bardziej go to zmotywowało do prowadzenia prac nad DEM! Nie można dłużej tego ciągnąć! Muszę wziąć się w garść i zniszczyć to, zanim będzie za późno!
???	???	???	???	???

*\*Id może się zmienić, zorganizujemy to, gdy wszystkie logi będą napisane*

Postaci

Gracz

Augmented Unit — uniwersalny robot odpowiedzialny za przechowywanie sekretów statku w sytuacjach kryzysowych i odzyskiwanie danych. Może się modyfikować, aby ułatwić wykonanie swojego celu.

Przeciwnicy

- ~~Działka automatyczne na statku blokujące dostęp do niektórych rejonów, wymagają zasilania, aby działać.~~
- ~~Załoga statku (po cofnięciu się w czasie), ludzie, którzy próbują powstrzymać AU przed ucieczką ze statku(?), stoją w miejscu i strzelają.~~

*Notka: Po implementacji i wewnętrznych testach gry uznaliśmy, że przeciwnicy spowalniają tylko grę i sprawiają, że będzie ona niepotrzebnie frustrująca*

NPC



~~Robot Porządkowy (RP) — mały robot, który pomaga graczowi, jeśli ten spędzi za dużo czasu w jednym obszarze nie wykonując żadnej konkretnej akcji~~

*Notka: Zrezygnowaliśmy z robota na rzecz podpowiedzi zawsze widocznych w konkretnych miejscach mapy*

## Rozgrywka i mechaniki

### Cel gry (cele/wyzwania/questy)

- **Głównym celem** gracza będzie dostanie się do wspomnianego wcześniej **Laboratorium**. Po drodze napotka różne przeszkody w postaci poruszających się platform, systemów bezpieczeństwa na statku, czy zagrożeń środowiskowych (woda/ogień/prąd).
- **Dodatkowo**, znalezione logi będą mogły oferować informacje, na podstawie których gracz będzie mógł próbować dostać się do opcjonalnych, ukrytych obszarów, które pozwolą mu na odkrycie głębszej historii, czy ukrytych zakończeń.
- Przy **zebraniu ostatniej modyfikacji** pojawi się pierwszy quest w grze - "przeżyj" i gracz będzie musiał wrócić do ~~początkowego pokoju~~ pokoju z łodzią podwodną w określonym czasie, ~~przy okazji unikając zabezpieczeń, które zostaną aktywowane~~ aby uciec ze statku.

*Notka: Zrezygnowano z systemów zabezpieczeń, zamiast tego uruchomiony zostanie odliczanie do samozniszczenia statku*

### Interakcja/kontrolery/sterowanie

Postać gracza będzie kontrolowana za pomocą klawiatury lub (opcjonalnie, nie w MVP) kontrolera. Podstawowe sterowanie zostało przedstawione poniżej:

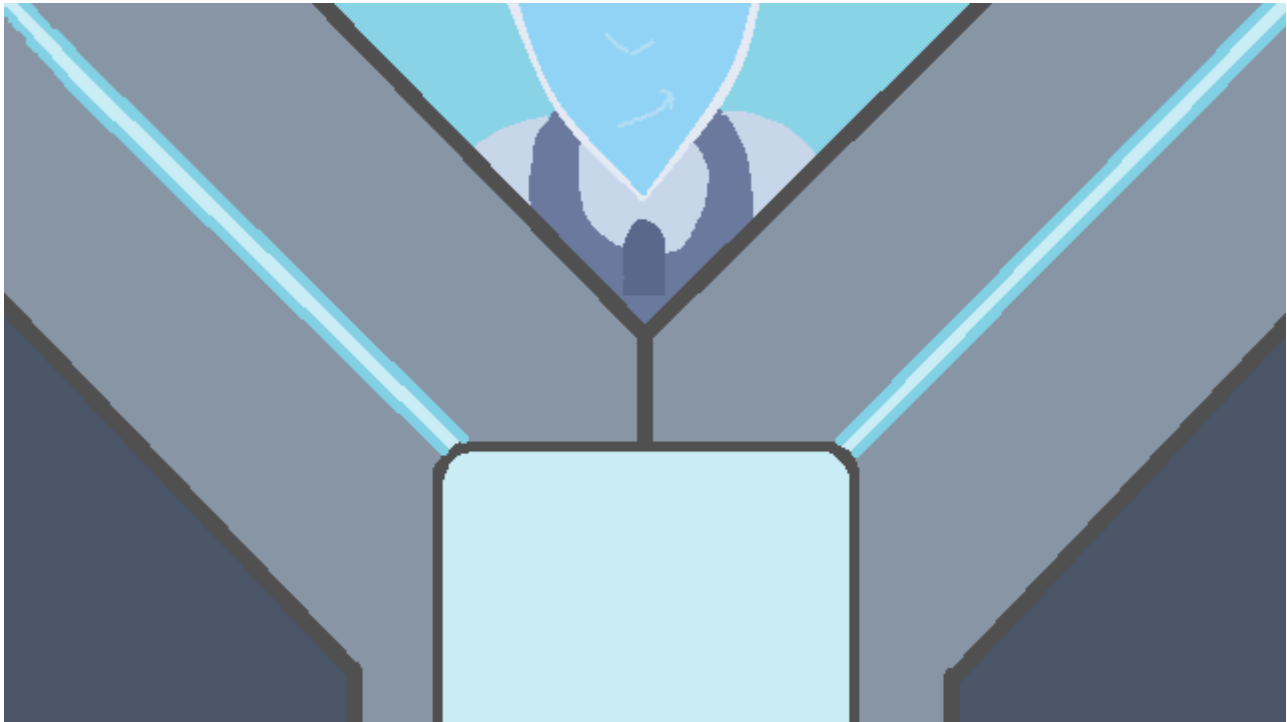
- **AD** - poruszanie się lewo-prawo
- **Spacja** - skok [dodatkowo podwójny skok — jako modyfikacja]
- **LShift** - zryw (modyfikacja)
- **E** - interakcja
- **Q** - krzyk (modyfikacja)

## Multiplayer

Z uwagi na gatunek gry (narracyjna), nie będzie ona wspierała gry wieloosobowej.

## Przebieg gry (flow)

### Główny splashscreen

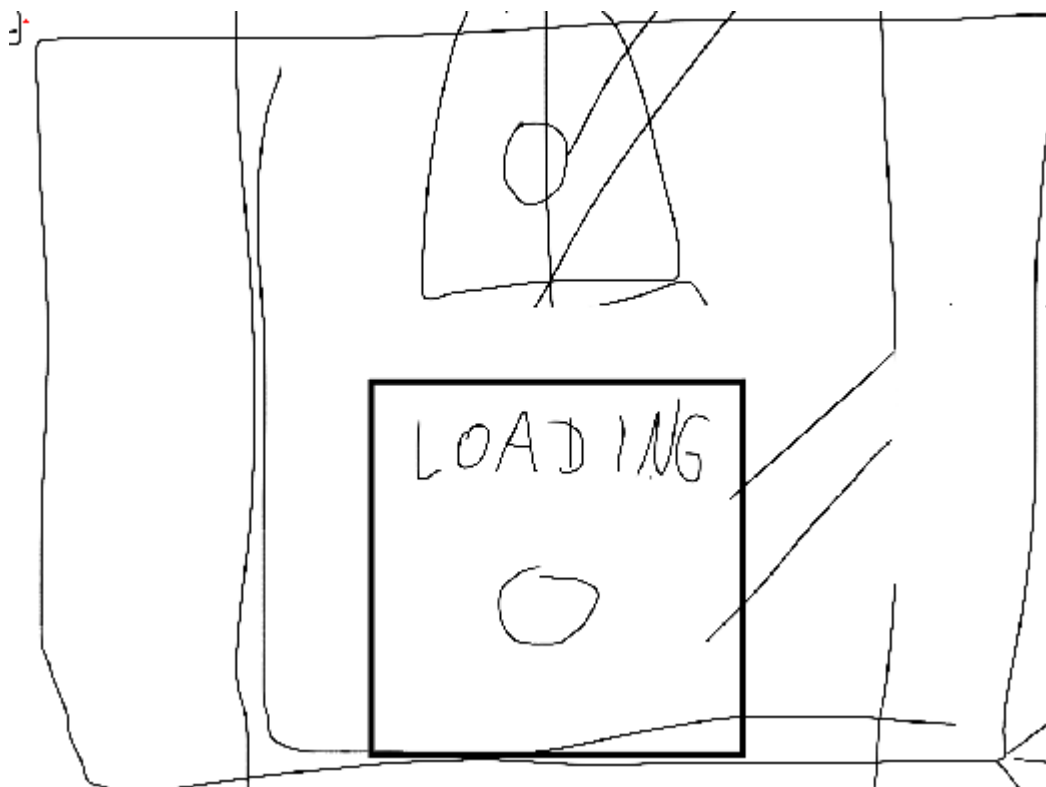


### Cutscenki, narracja in-game'owa

- Cutscenki — nieliczne, w kluczowych momentach fabuły, renderowane w silniku, zabierając chwilowo kontrolę graczowi, lub opcjonalnie animowane (przy zakończeniach)
- Narracja — narrator odczytujący logi znajdujące na statku (lista logów w sekcji [Fabuła/Logi](#))

## Menu

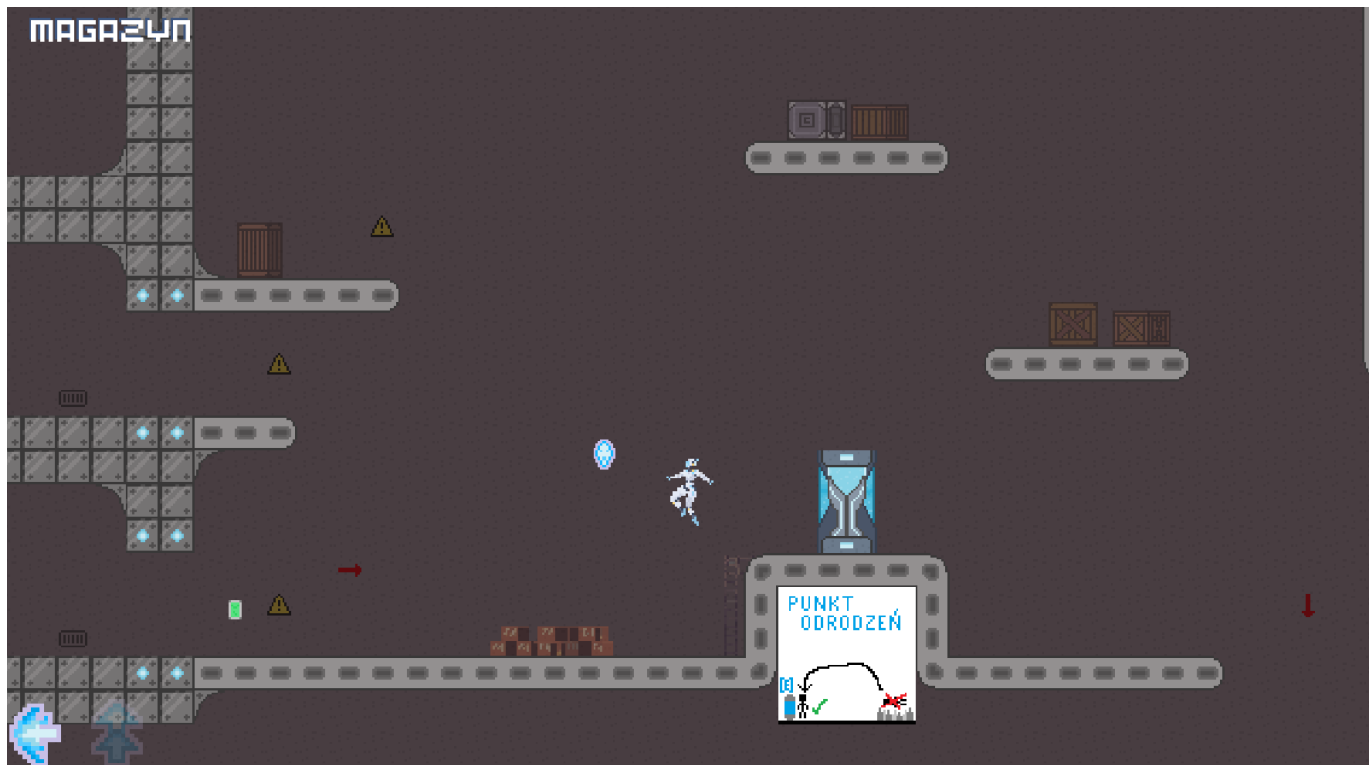
Menu będzie animowanym przybliżeniem splashscreena z opcjami wyświetlanymi na monitorze (in-game hud). Poniżej można zobaczyć concept-art menu i jego finalną wersję

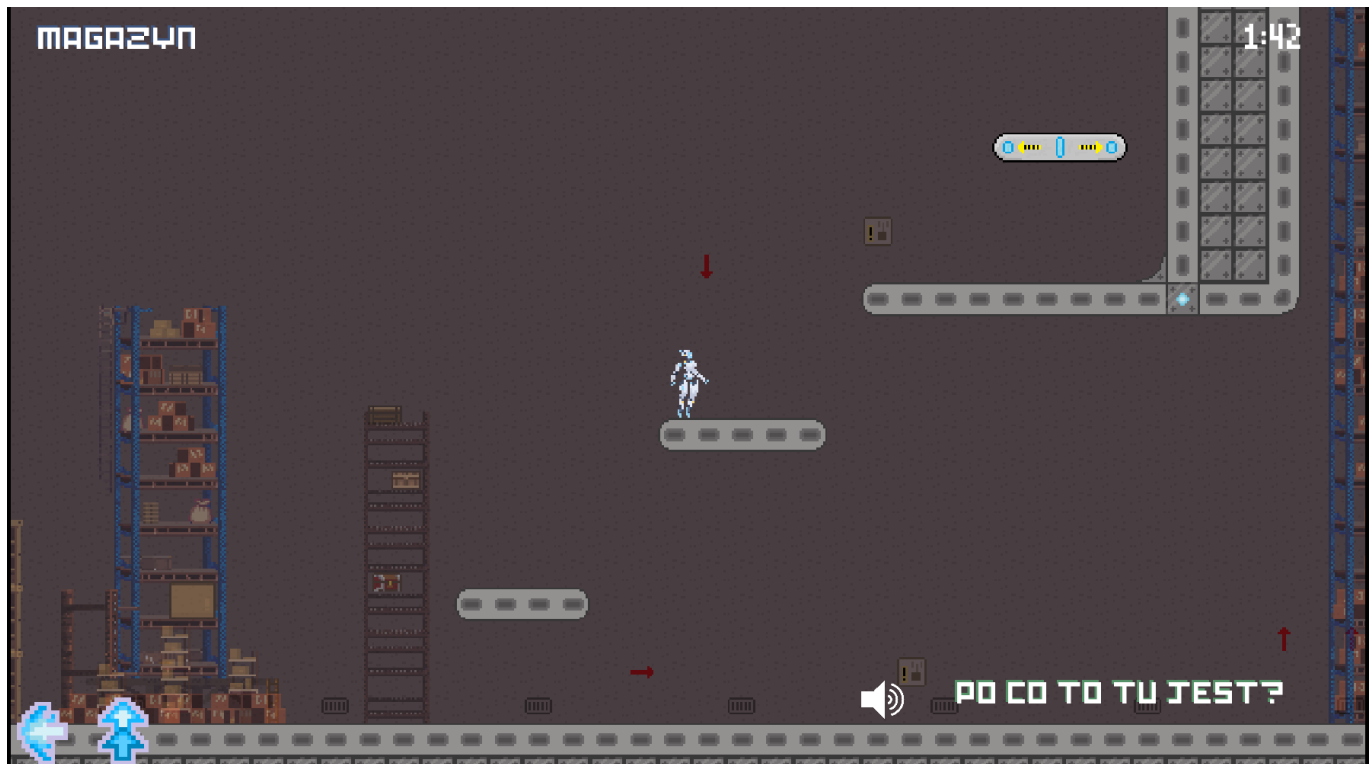


## HUD

Bardzo ograniczony, aby nie rozpraszać gracza. W lewym górnym rogu znajdować się będzie nazwa aktualnego poziomu/sekcji, w lewym dolnym ikonki odblokowanych modyfikacji, przyciemnione, gdy gracz nie

może ich użyć. Prawy górny róg zostanie wykorzystany przy alternatywnych zakończeniach - wyświetli pozostały czas, a w prawym dolnym będą pojawiać się aktualnie grające logi.





## Mapy

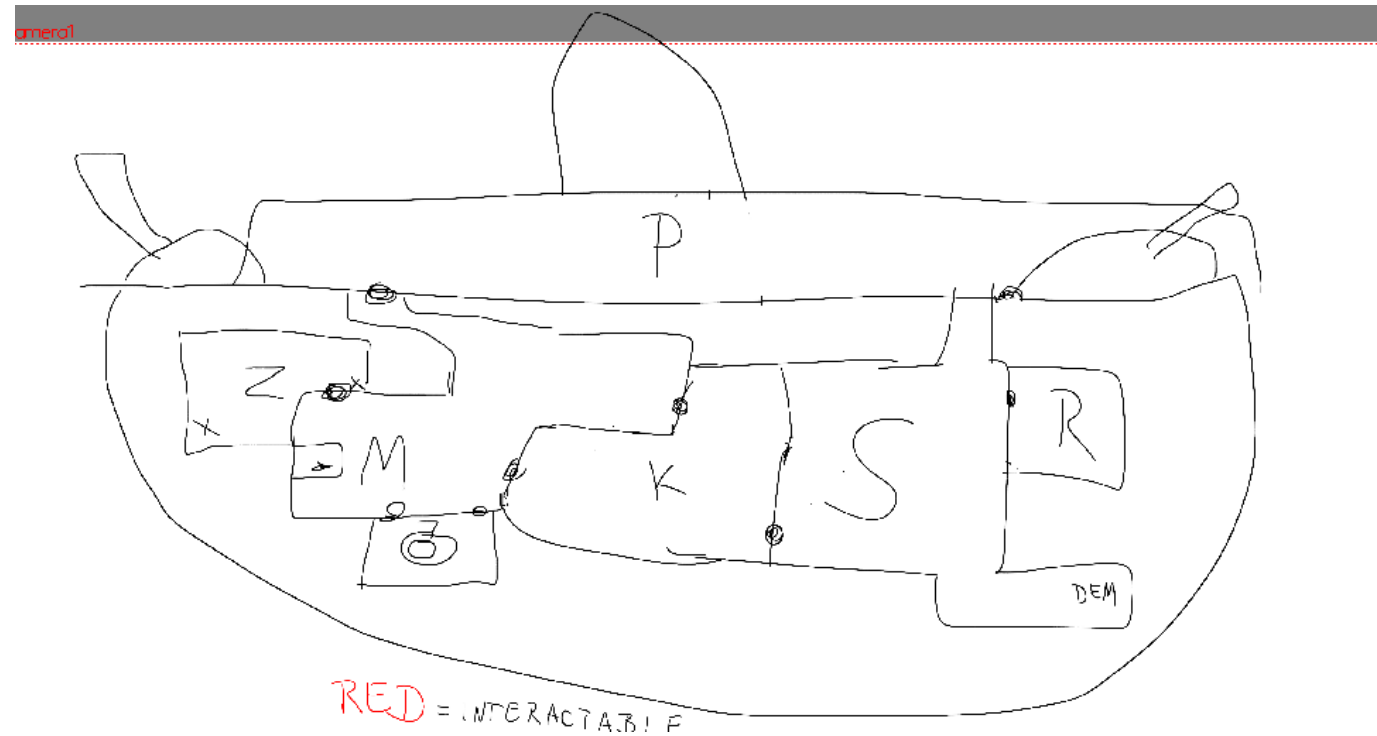
Statek będzie jedną wielką mapą, na której każda **sekcja** będzie osobnym poziomem z "animacją" wygaszania ekranu przy przejściu między nimi (na czas ładowania poziomu).

Gracz powinien w domyśle odwiedzić pokoje w następującej kolejności:

1. Pokój treningowy (tutorial)
2. Magazyn (**M**)
3. Kajuty (**K**)
4. Pokład statku (**P**)
5. Laboratorium, dawniej serwerownia (**S**)

Warto zaznaczyć, że pokoje 4 i 5 mogą być odwiedzone w innej kolejności, jednak przejście gry (zdobywając przedmiot z Laboratorium) będzie wymagać przejścia przez Pokład statku.

Pokoje opcjonalne nie zostały tu uwzględnione, jednak gracz od samego początku będzie miał z Magazynu dostęp do Pokoju z łodzią podwodną (**miniaturka łodzi, pod Magazynem**), z Laboratorium będzie mógł dość łatwo dostać się do Reaktora (**R**), a po jego wyłączeniu cofnąć się i przez Magazyn wrócić do Zbrojowni (**Z**).



Notka: Rozłożenie pokojów uległo zmianie

Assety

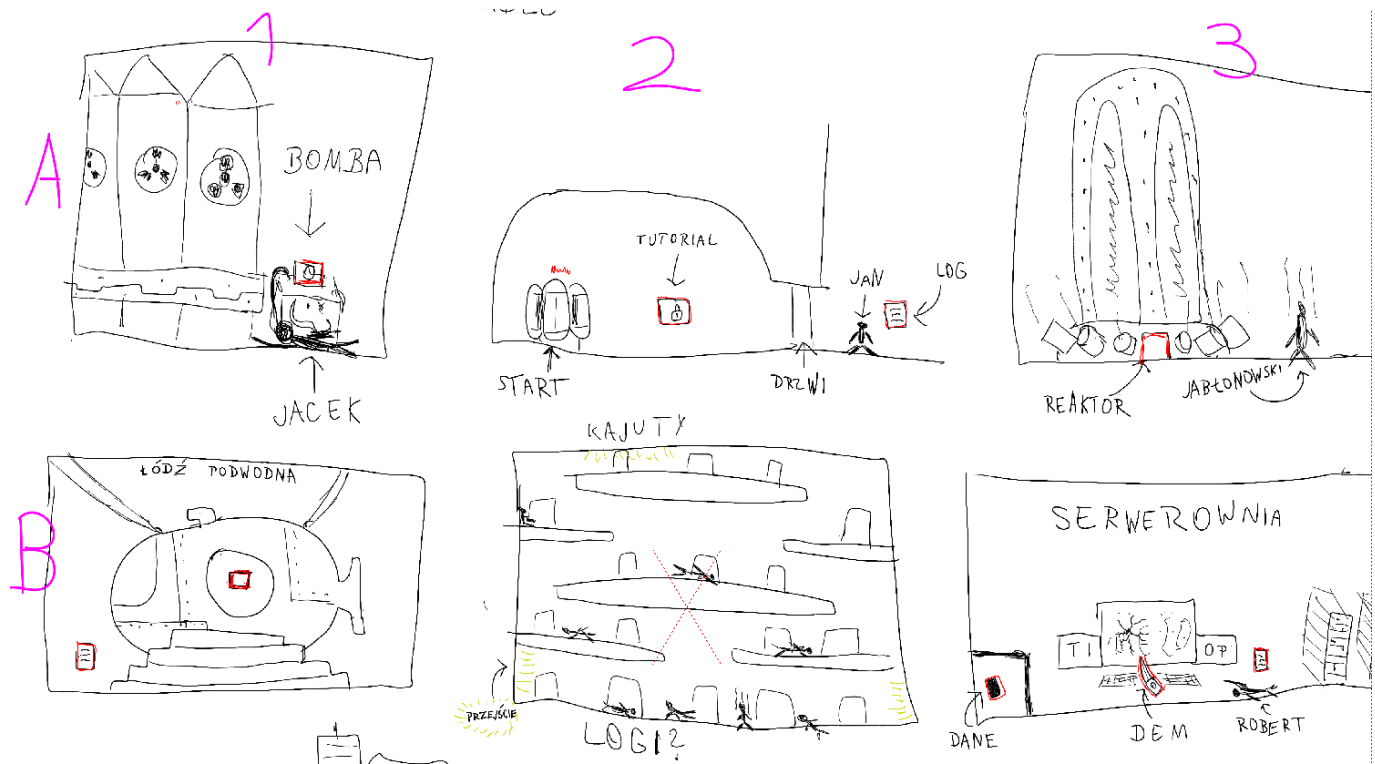
Styl graficzny

Gra będzie tworzona jako pixelart o (podstawowych) rozmiarach 16x16px. Będą dominowały tutaj kolory raczej przyciemnione, tak aby postaci, logi, czy platformy wyróżniały się od otoczenia. Całość będzie wyglądała raczej mrocznie (klimat grafiki podobny do gry [Signalis](#), a ogólny wygląd/feeling do gry jak w [Dead Cells](#) albo [Hollow Knight](#))

Concept-art

Na poniższym zrzucie ekranu widać pierwsze concept-arty kilku pokoi (fragmentów sektorów/lokacji), które są istotne fabularnie. Są to:

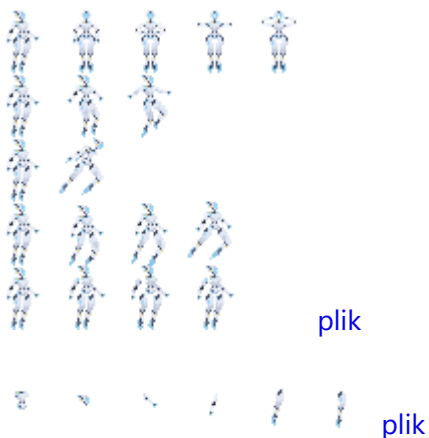
-	1	2	3
A	Zbrojownia - miejsce z bombami	Pokój startowy	Reaktor
B	Pokój z łodzią podwodną	Kajuty	Ostatni pokój w laboratorium



Warto zaznaczyć, że część pomieszczeń, ich plany, czy rozmieszczenie mogą ulec zmianie.

## Spritesheety

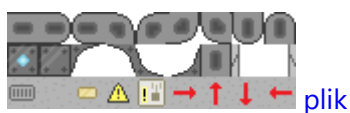
Postać gracza:



Punkt odrodzeń:



Magazyn:



Plik z ozdobami poziomu musi jeszcze zostać przerobiony, aby mógł się zmieścić w GDD

## Muzyka

Cała muzyka w grze została wygenerowana z pomocą [SUNO AI](#). Każda sekcja statku ma własny utwór grający w tle, poza tym istnieją różne utwory odgrywane przy zakończeniu, bądź po znalezieniu "easter egga"

## Narracja

Treść wszystkich logów (będących jedyną formą narracji w grze) znajduje się w sekcji [Fabuła/Logi](#)

Gracz będzie mógł przeglądać i odsłuchiwać zebrane wcześniej logi z poziomu menu.



## PoC/prototyp

Na potrzeby prototypu przygotowane zostaną:

- Postać gracza (z całą maszyną stanów i mechanikami, ale tylko prostymi assetami), mechaniki/stany:
  - skok (+ podwójny skok i "coyote time" - możliwość skoku przez krótką chwilę po opuszczeniu platformy)
  - zryw (szybki ruch w lewo lub prawo, przez krótki moment ignoruje grawitację)
  - krzyk (pozwalający zresetować momentum i przywracający wszystkie zdolności)
  - ruch
  - spadanie
  - stanie w miejscu
  - śmierć (kończąca grę)
- Specjalny pokój (później przerobiony na pokój developerski i finalnie na tutorial)
- Bazowy "prefab" logów z prostym dźwiękiem i tymczasową grafiką
- Wyjście z pokoju (kończące grę, służące później jako drzwi między sekcjami — poziomami)
- Przeszkoda zabijająca gracza
- Punkt kontrolny
- (Opcjonalnie) ruchoma platforma



# Implementacja - do uzupełnienia na bieżąco

Byty (entity) + krótki opis

## **Gracz**

Odpowiedzialność za sterowanie ruchem postaci została w pełni oddelegowana do maszyny stanów, która używa metod aktualnego stanu do obsługi tego ruchu. W celu utrzymania kodu stanów w czytelnej postaci, całość logiki przechowywana jest w osobnym Nodzie (Node) w scenie gracza. Pozwala to też trzymać wszystkie ważne do debugowania zmienne w jednym miejscu i mieć ciągły podgląd stanu postaci.

W poniższej tabeli znajdują się wszystkie przejścia między stanami, nazwy stanów zostały takie jak w kodzie, tj:

- Jumping --> skok
- Dashing --> zryw
- Screaming --> krzyk
- Moving --> ruch
- Falling --> spadanie
- Idle --> stanie w miejscu
- Dying --> śmierć

X oznacza dowolny stan, poza *Dying*.

Aktualny stan	Następny stan	Warunek
X	Dying	Gracz otrzymał obrażenia
X	Falling	Gracz nie stoi na żadnej platformie
X	Screaming	Gracz użył krzyku
Falling	Idle	Gracz wylądował na ziemi <b>nie</b> poruszając się
Falling	Moving	Gracz wylądował na ziemi poruszając się
Falling	Jumping	Gracz posiada jeszcze dostępne skoki i nacisnął przycisk skoku
Falling	Dashing	Gracz posiada jeszcze dostępne zrywy i nacisnął przycisk zrywu
Moving	Idle	Gracz przestał się ruszać
Moving	Jumping	Gracz nacisnął przycisk skoku*
Moving	Dashing	Gracz nacisnął przycisk zrywu*
Idle	Jumping	Gracz nacisnął przycisk skoku*
Idle	Dashing	Gracz nacisnął przycisk zrywu*
Idle	Moving	Gracz nacisnął przyciski odpowiadające za chodzenie w prawo lub lewo
Jumping	Jumping	Gracz posiada jeszcze dostępne skoki i nacisnął przycisk skoku**
Jumping	Dashing	Gracz posiada jeszcze dostępne zrywy i nacisnął przycisk zrywu

\*Wejście do stanów *Moving* oraz *Idle* odnawia skoki, zrywy i krzyki gracza, więc będąc w tych stanach, gracz musi posiadać dostępne skoki/zrywy/krzyki

\*\*Przejście ze stanu *Jumping* do *Jumping* to jedyny przypadek, w którym jawnie przechodzimy ze stanu w samego siebie (zamiast zostawać w nim!), sprawia to, że podwójny skok jest możliwy, gdyż nadanie prędkości dzieje się na wejściu do stanu

Postać gracza ma też dwa różne źródła dźwięków, aby uniknąć problemów z kolejkowaniem albo przerywaniem dźwięku. Jedno służy do odtwarzania logów, drugie (zarządzane przez specjalnego menadżera audio) odtwarza inne dźwięki takie jak skok, śmierć czy zryw. Aby odegrać dźwięk za pomocą menadżera audio, należy najpierw dodać odpowiednią wartość do enuma *Sound* i mapować go na plik z dźwiękiem w słowniku *mapping*.

### Logi (główne narzędzie narracji)

Logi dziedziczą z klasy *Interactable* i posiadają tylko *id* (typu *Logs.LogId*) oraz metodę *interact*. Klasa *Logs* znajduje się w folderze *res://src/scripts/common/Utils* i posiada tylko enum *LogId*, stałą *logs* (słownik łączący *LogId* z wszystkimi potrzebnymi danymi logów — tytuł, treść i załadowany plik audio) oraz statyczne metody pomagające pracować z logami (*get\_log\_audio*, *record\_log\_pickup* i inne, jeśli zajdzie taka potrzeba).

### Punkty odrodzeń

Dziedziczą z klasy *Interactable*, posiadają *Hardpoint* służący do odradzania gracza. Po wejściu w interakcję z punktem odrodzeń gra on krótki dźwięk i staje się aktywny, zmieniając kolor na zielony (z niebieskiego). Tylko jeden punkt może być aktywny w danym momencie, więc przy wejściu w interakcję z następnym punktem, ten aktywny wcześniej wyłącza się.

### Poziomy i ich specyficzne mechaniki

Poziomy zostały ogólnie opisane w sekcji [Tematyka i osadzenie gry/Lokacje](#). Więcej informacji pojawi się w tej sekcji, gdy będą one implementowane.

#### Poziom 1 - Magazyn

Mechaniką specyficzną dla tego poziomu są poruszające się platformy. Jedna scena (prefab) platformy może być skonfigurowana, dodając odpowiednią ścieżkę i zaznaczając czy jest ona zamknięta (początek to ten sam punkt co koniec), czy otwarta. Platformy mają domyślne prędkości, ale mogą być one zmieniane (osobno dla ścieżek otwartych i zamkniętych).

Na poziomie zostały też ustawione pułapki w postaci kolców i płomieni, które pozwalają zablokować postęp gracza, zanim odblokuje on zryw, umieszczony przy końcu poziomu. Do ułatwienia gry, na tle obecne są strzałki, wskazujące kierunek ewentualnego przechodzenia poziomu. Mimo to gracz może pójść w innym kierunku niż ten wskazany i spróbować przejść grę odblokowując rzeczy w innej kolejności.

#### Poziom 2 - Kajuty

W kajutach specyficzną mechaniką są "trampoliny", które wystrzeliwiają gracza we wskazanym kierunku. Każda trampolina ma własną siłę i czas, na który interakcja gracza zostaje wyłączona (aby tworzyć pułapki i specjalne przejścia).

Za jedną pułapką został umieszczony pokój, do którego można się dostać jedynie po odblokowaniu wszystkich modyfikacji, w nim znajduje się "easter egg" zmieniający muzykę w tle do czasu powrotu do menu lub włączenia alarmu (koniec gry).

Odblokowujemy tu podwójny skok.

### Poziom 3 - Pokład

Na pokładzie specyficzną mechaniką jest burza - co jakiś czas w okolicy gracza uderzają pioruny, które mogą go zabić. Dodatkowo - jest to jedyna sekcja, która nie jest w pełni zamknięta - gracz może wyskoczyć poza burtę, co również zakończy się śmiercią postaci gracza.

Na tym poziomie również znajduje się ukryty pokój, do którego można wejść, jeśli dostaliśmy się na pokład, znajduje się w nim jeden dodatkowy log - "easter egg", który jest jedynie szczeknięciem psa.

Odblokowujemy tu krzyk.

### Poziom 4 - Laboratorium

Na tym poziomie pojawiają się anomalie, które aktywują się losowo co jakiś czas i zostają aktywne na 5 sekund. Przy wejściu w aktywną anomalię nakładane są na gracza losowe efekty - aktualnie zamiana kierunków sterowania albo teleportacja do innej losowo wybranej anomalii.

Na poziomie został ukryty pokój, do którego można się dostać, jedynie korzystając z anomalii.

Znajduje się tu DEM - przedmiot, którego zebranie kończy grę.

### Poziom 5 - Reaktor

Mały poziom będący "podpoziomem" laboratorium (przechodząc między nimi nie zmienia się muzyka). Znajduje się tutaj reaktor, który możemy aktywować, aby przywrócić zasilanie na statku i móc uciec łodzią podwodną.

### Poziom 6 - Pokój z łodzią podwodną

Mały poziom, "podpoziom" magazynu. Znajduje się tu łódź podwodna, która pozwala uciec ze statku - wygrać grę.

### Poziom 7 - Zbrojownia

Mały poziom, pozwala aktywować sekwencję samozniszczenia (a tym samym alarm), co pozwoli na odblokowanie alternatywnego zakończenia gry.

### Poziom 8 - Samouczek/Trening

Mały poziom pełen "podpowiedzi", które pozwalają graczowi zapoznać się ze sterowaniem i podstawowymi mechanikami wykorzystywanymi przez resztę gry.

## Modyfikacje

Zbierając modyfikacje, gracz będzie mógł dostać się do nowych stanów, bądź odblokować nowe przejścia między stanami

- **Podwójny skok** — pozwala wykonać jeden dodatkowy skok w powietrzu

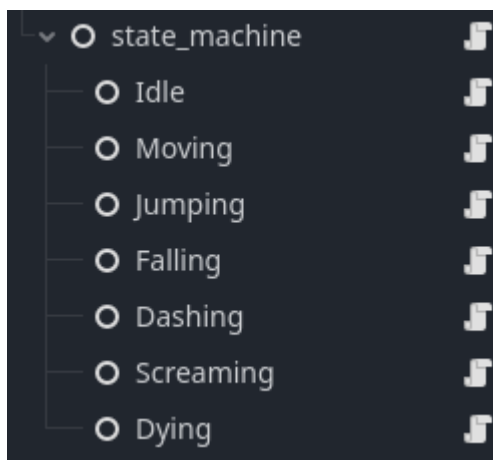
- **Zryw** — pozwala bardzo szybko przemieścić się horyzontalnie i przejść przez miejsca niemożliwe do pokonania z użyciem samego skoku
- **Krzyk** — pozwala zatrzymać się w miejscu na jakiś czas (niezależnie od grawitacji, prędkości itp.), odnawia Zryw i podwójny skok bez potrzeby dotknięcia platformy.

## Ogólne, reużywalne skrypty/narzędzia

### Maszyna stanów

Klasa `StateMachine` oraz bazowa klasa `State`, za pomocą których można zbudować maszynę stanów opartą o Node'y. Klasy zakładają, że logika poruszająca postacią zostanie oddelegowana do stanów. Maszynę należy zainicjować, wstrzykując zależność — kontrolowaną postać (klasy `Actor`).

**Przykład** zaimplementowanej maszyny stanów w drzewie postaci gracza:



`state_machine` to Node z podpiętym skrypcem maszyny stanów, z kolei każde dziecko to stan, z własnym skrypcem rozszerzającym klasę `State` i posiadającym logikę specyficzną dla tego stanu.

### Interactable

Klasa bazowa dla wszystkich rzeczy, z którymi gracz może wejść w interakcję. Posiada jedynie zmienną `log_blocks_interaction`: boolean, oraz metodę `interact(entity: Actor) -> void`. Zmienna decyduje o tym, czy gracz może wejść w interakcję z danym przedmiotem, jeśli log jest odtwarzany. Domyślną wartością jest `false` (odtwarzanie logów nie blokuje interakcji), jednak niektóre klasy (`Log` oraz `Door`) nadpisują tę wartość w metodzie `ready()`, gdyż w ich przypadku wejście w interakcję mogłoby spowodować przerwanie odtwarzania logu na rzecz odtworzenia następnego, bądź z powodu zmiany pozycji gracza między poziomami. Jeśli byt nie rozszerza tej klasy, gracz nie będzie w stanie wejść z nim w interakcję (klikając "E")!

### Hardpoint

Klasa odpowiedzialna za reprezentację niewidzialnych punktów na stałe ustawionych na mapie. Każdy hardpoint ma swoje Id, aktualnie hardpointy używane są tylko przez menedżera poziomów do odradzania albo ładowania do poziomu postaci gracza.

### ProgressionManager

Klasa zajmująca się zbieraniem informacji o postępach gracza. Aktualnie zapisywane tam są tylko informacje o zebranych logach i modyfikacjach, jednak z czasem może ona zostać rozbudowana. Informacje te są aktualnie wykorzystywane przy przejściach między poziomami, tak, aby zebrane wcześniej przedmioty nie pojawiały się drugi raz na mapie po powrocie do tej samej strefy.

Ponadto zajmuje się ona wszystkimi "licznikami czasu" takimi jak ogólnie spędzony czas w grze i licznik pojawiający się na koniec gry. Zakończenia gry, jak i oddelegowanie zapisu danych również są tutaj obsługiwane

### **LevelManager**

Klasa zajmująca się przejściami pomiędzy poziomami, w tym zmianami muzyki (konsultując aktualny stan gry z ProgressionManagerem) oraz animacjami przejść.

### **SaveData**

Statyczna klasa zajmująca się ściśle zapisem danych. Otrzymuje dane od ProgressionManagera i zapisuje je w odpowiednim pliku. Pozwala też odczytywać dane z pliku i resetować zapis do domyślnych wartości. Wykorzystuje wbudowaną klasę ConfigFile, gdyż dane o Logach i Modyfikacjach nie mogły zostać zapisane w standardowym pliku zapisu.

### **Hint**

Mała klasa rozszerzająca wbudowane Area2D. Sprawia ona, że gdy gracz wejdzie w daną strefę, pojawia się nad jego głową tekst - odpowiedź. Silnie wykorzystana w samouczku i przy modyfikacjach.

# Zakres projektu

## harmonogram i podział prac

Przy każdym zadaniu znajdzie się jedna z trzech liter, która oznacza, przez kogo zostało ono wykonane:

- O — oboje
- K — Kamil
- M — Mateusz

Zadania będą przypisane konkretnym tygodniom i uzupełniane na bieżąco w poniższej liście, tygodnie po aktualnym to jedynie przewidywane prace, prawdopodobnie ulegną zmianie.

- **Tydzień 1 (29.02.2024 - 07.03.2024)**
  - (O) Przygotowanie wstępnej wersji GDD i ogólnej wizji gry
  - (O) Przygotowanie concept-artów
  - (K) Przygotowanie ogólnego zarysu fabuły
  - (M) Research podobnych historii i inspiracji do napisania szczegółowej fabuły
  - (K) Wstępne napisanie kilku logów
- **Tydzień 2 (07.03.2024 - 14.03.2024)**
  - (K) Przygotowanie prototypu gry bazującego na starych assetach
  - (M) Przygotowanie oprogramowania do tworzenia pixelartu
  - (M) Stworzenie pierwszych projektów poziomów
  - (K) Przygotowanie repozytorium i projektu gry w silniku
  - (K) Kontynuowanie pracy nad fabułą i logami
  - (O) Aktualizowanie GDD
  - (M) Przygotowanie pierwszych wersji części assetów i animacji
- **Tydzień 3 (14.03.2024 - 21.03.2024)**
  - (K) Dopracowanie prototypu
  - (O) Aktualizowanie GDD
  - (M) Kontynuowanie prac nad assetami i animacjami
  - (K) Kontynuowanie prac nad fabułą i logami
- **Tydzień 4 (21.03.2024 - 28.03.2024)**
  - (O) Aktualizowanie GDD
  - (M) Kontynuowanie prac nad assetami i animacjami (stany gracza)
  - (K) Dokończenie prac nad logami i nagranie ich pierwszych wersji
- **Tydzień 5 (28.03.2024 - 04.04.2024)**
  - (O) Aktualizowanie GDD
  - (K) Implementacja systemu poziomów
  - (K) Implementacja systemów pomagających w obsłudze gry (np. manager audio, manager poziomów itp.)
  - (M) Implementacja efektu śmierci gracza w silniku gry
  - (M) Kontynuowanie prac nad assetami i animacjami (inne postaci w grze)
- **Tydzień 6 (04.04.2024 - 11.04.2024)**
  - (O) Aktualizowanie GDD
  - (O) Tworzenie pozostałych assetów (poziomy, środowisko itp.)
  - (K) Przygotowanie pierwszego poziomu w silniku (potrzebne zmienne, sceny itp.)

- **Tydzień 7 (11.04.2024 - 18.04.2024)**
  - **(O)** Aktualizowanie GDD
  - **(M)** Projektowanie poziomów
  - **(K)** Implementacja systemów wspomagających przejścia między poziomami
  - **(K)** Początek implementacji kolejnego poziomu
- **Tydzień 8 (18.04.2024 - 25.04.2024)**
  - **(O)** Aktualizowanie GDD
  - **(M)** Projektowanie HUD i menu
  - **(K)** Implementacja kolejnych poziomów
  - **(K)** Implementacja HUD i menu
- **Tydzień 9 (25.04.2024 - 02.05.2024)**
  - **(O)** Aktualizowanie GDD
  - **(M)** Implementacja systemu load/save
  - **(K)** Implementacja systemów umożliwiających mierzenie czasu gracza
- **Tydzień 10 (02.05.2024 - 09.05.2024)**
  - **(O)** Aktualizowanie GDD
  - **(M)** Tworzenie cutscenek
  - **(K)** Wprowadzanie poprawek i dokończenie brakujących systemów
- **Tydzień 11 (09.05.2024 - 16.05.2024)**
  - **(O)** Aktualizowanie GDD
  - **(M)** Implementacja samouczka
  - **(K)** Implementacja easter-eggów, minigierek i alternatywnych trybów sterowania
- **Tydzień 12 (16.05.2024 - 23.05.2024)**
  - **(O)** Aktualizowanie GDD
  - **(O)** Poprawki ewentualnych błędów
- **Tydzień 13 (23.05.2024 - 30.05.2024)**
  - **(O)** Aktualizowanie GDD
  - **(O)** Poprawki ewentualnych błędów
- **Tydzień 14 (30.05.2024 - 06.06.2024)**
  - **(O)** Aktualizowanie GDD
  - **(O)** Poprawki ewentualnych błędów
- **Tydzień 15 (06.06.2024)**
  - **(O)** Aktualizowanie GDD
  - **(O)** Przeprowadzenie playtestów
  - **(O)** Wprowadzanie poprawek po playtestach
- **Tydzień 15 (06.06.2024)**
  - **(O)** Aktualizowanie GDD
  - **(O)** Wprowadzanie poprawek po playtestach