

Possessed Redeemer

Dokumentacja deweloperska

Autorzy:

Mateusz Małyszka

Agnieszka Reguła

Jakub Wujkiewicz

Krzysztof Waśko

1. Fabuła gry

Fabuła gry osadzona jest w świecie, na który kilkadziesiąt lat temu ktoś rzucił klątwę, w wyniku czego niebo poczerniało rozpoczynając noc, która trwa do teraz, a wszędzie zaczęły się pojawiać mroczne stworzenia, których jedynym celem jest wyeliminowanie wszystkiego, co żyło na tym świecie przed nimi. Nikt już nie pamięta jak to się stało ani dlaczego. Krążą jedynie legendy zawierające szczątki zapomnianej prawdy. Jedyne, co się w nich wszystkich powtarza to to, że sposób na odczynienie klątwy spoczywa na dnie dołu monstualnych rozmiarów, który wydaje się nie mieć dna. Na przestrzeni lat, wielu ochotników zeszło w głąb pustki próbując uratować świat przed ciemnością, lecz nikt nie wrócił. Wszyscy dawno porzucili nadzieję, a owa przepaść służy za masowy Grób, do którego wrzucane są ofiary bestii terroryzujących świat.

Tak się zaczyna historia głównego bohatera. Jedno z ciał zrzuconych do Grobu zostaje wybrane i opętane przez to, co według legend miało odczynić klątwę panującą na powierzchni. Jego zadaniem jest wydostać się z Grobu pokonując przeszkody na swojej drodze...

2. Technologie użyte w projekcie

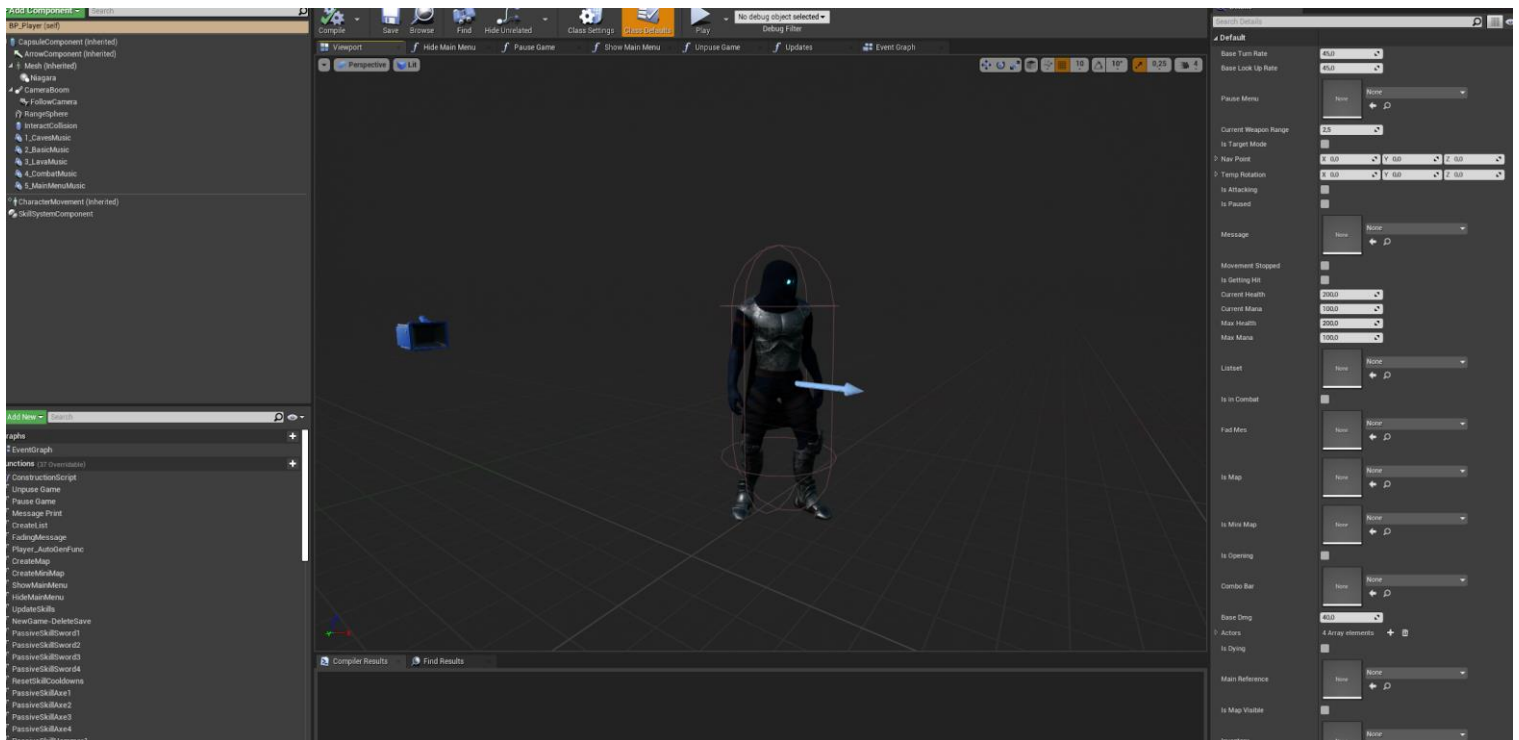
W projekcie Possessed Redeemer korzystaliśmy z silnika Unreal Engine 4, a do kontroli wersji wdrożyliśmy Perforce Helix. Silnik ten ma możliwość programowania w języku C++ lub w Blueprintach. Blueprintem nazywamy skrypt wizualny używane w UE4.

3. Wymagania sprzętowe

- System operacyjny: Windows 10 64 bit
- Procesor: Intel Core i3 4130 vs AMD FX-8350
- Karta graficzna: NVIDIA GTX 550Ti lub AMD RADEON HD 4890
- Pamięć: 6GB RAM
- Dysk Twardy: 5GB
- DirectX 11

4. Bazowe klasy projektu:

BP_Player - klasa postaci gracza zawierająca większość funkcjonalności związanych z inputem i systemem walki.



UI_Hud - interfejs użytkownika zawierający życie, manę oraz inwentarz do zbierania kluczy.

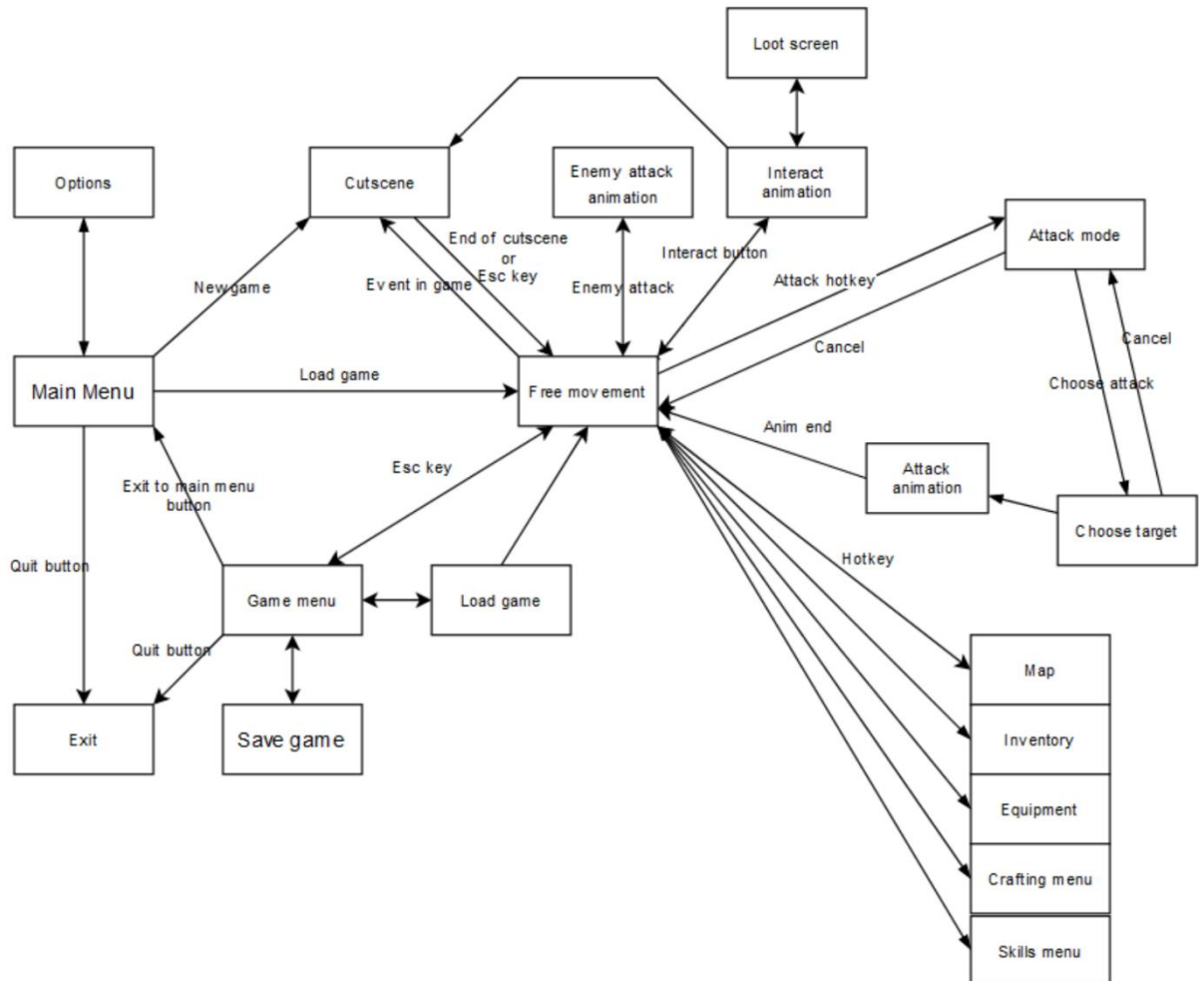
PlayerController - interfejs między bohaterem, a graczem.

GameStateSave - służy do zapisu stanu poziomu (rozwiązane zagadki, pokonani przeciwnicy).

MyGameInstance - przechowuje stan gry w trakcie rozgrywki (życie, mana, zdobyte klucze i inne), przechowuje wartości między poziomami, resetuje się po wyjściu z gry.

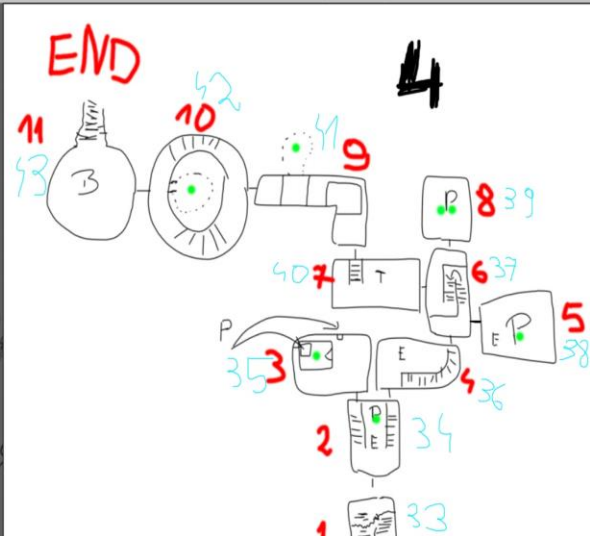
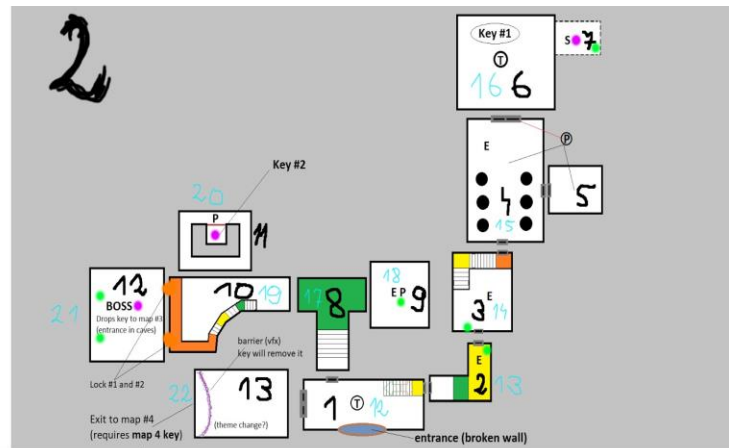
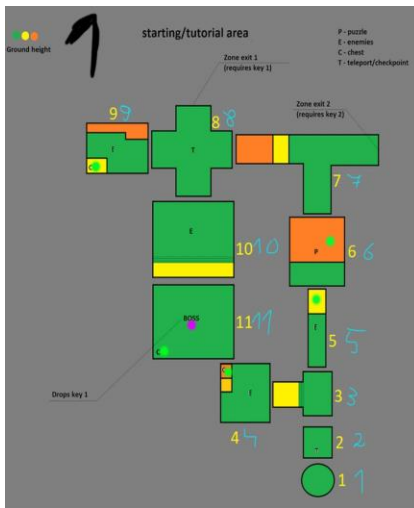
5. Model logiczny gry

GAME LOGIC DIAGRAM



6. Mapy

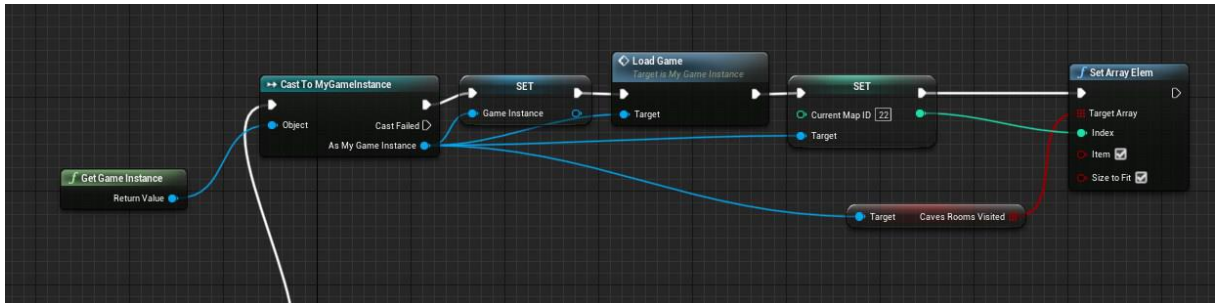
Każda z 4 map składa się z 11-13 leveli. Levelem nazywanym zaprojektowany od podstaw pokój.



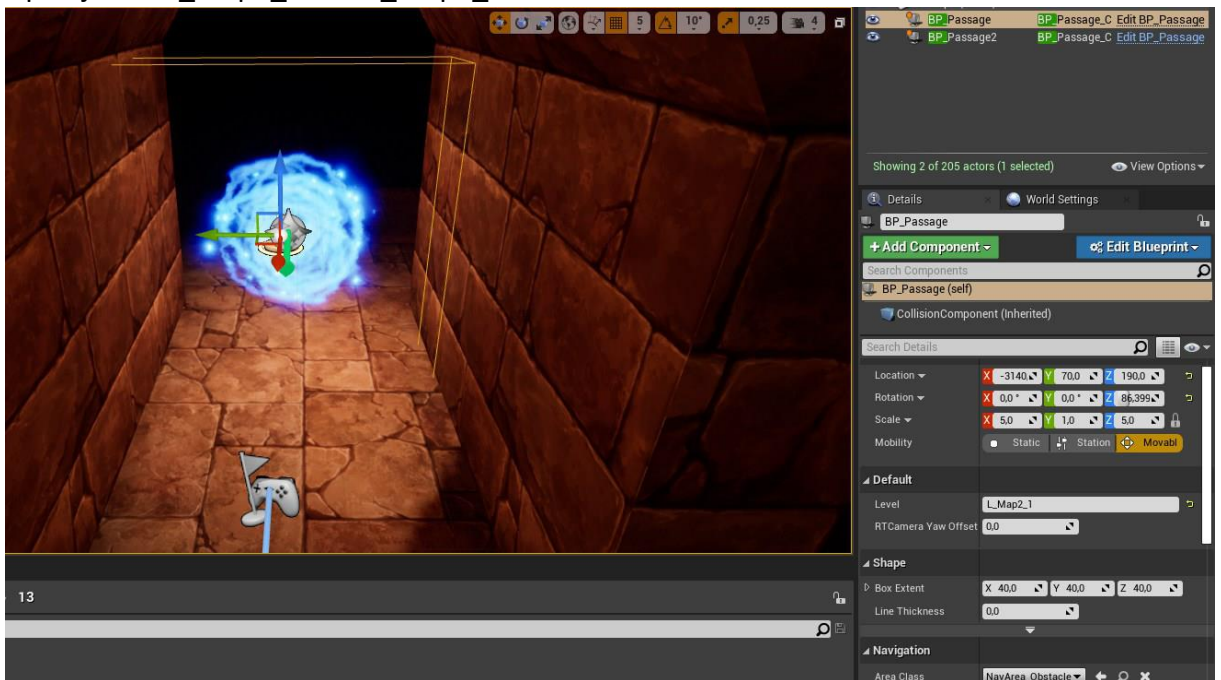
Legenda:

- żółte, czerwone, czarne, różowe cyferki oznaczają numerację pokoi (wliczając część ukrytych pokoi) - te numeracje nie są wykorzystywane tylko do nazewnictwa plików map
- niebieskie cyfry - unikatowe ID każdego z pokoi - przypisane do każdego z nich w Level Blueprint
- zielone kropki - rozmieszczenie kryształów dających punkty umiejętności
- T, niebieskie kropki - rozmieszczenie teleportów
- rozmieszczenie kluczy zostało zaznaczone strzałkami
- kolory: zielony/żółty/pomarańczowy oznaczają ukształtowanie terenu

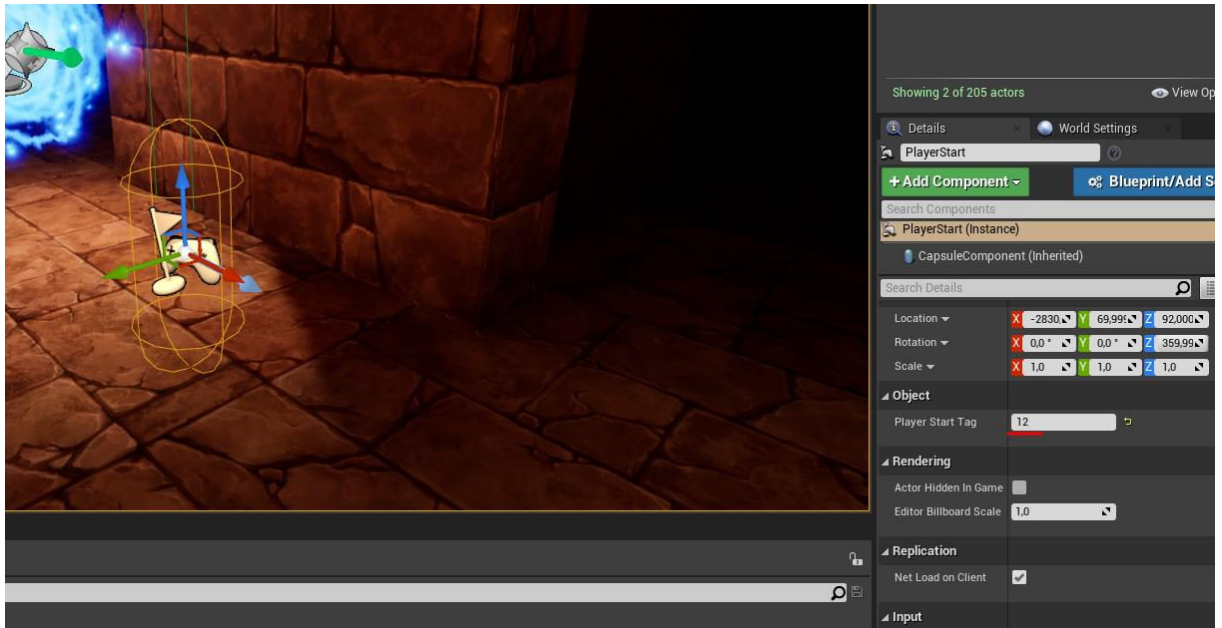
Tworząc nowy pokój należy pamiętać aby ustawić mu unikatowy ID w Level Blueprint wg schematu pokazanego poniżej (na schemacie ustawiony ID: 22)



Aby utworzyć przejście do kolejnego pokoju trzeba dodać BP_Passage i w sekcji "Default" zmienić wartość parametru "Level" na nazwę pokoju do którego chcemy utworzyć przejście
 Przykład przejścia z L_Map2_13 do L_Map2_13



Jeśli w pokoju jest wiele pozycji startowych (Player Start) należy ustawić w nich w sekcji "Object" w ustawieniu "Player Start Tag" unikatowe ID pokoju, z którego się "przychodzi" (to unikatowe ID ustawiane w Level Blueprint)



Aby ustandaryzować poziom oświetlenia w poszczególnych pokojach należy dodać do każdego utworzonego pokoju "PostProcess Volume" a następnie wprowadzić następujące ustawienia:

- Infinite extend unbound "true"
- Metering Mode "true" + Auto Exposure Histogram
- Min Brightness 0.25
- Max Brightness 1

Mapa 1 - Charakterystyka:

Główny kolor: FF2400FF

Opcjonalny kolor - nad grzybami pochodne żółte/niebieskiego/pomarańczowego

Main assets:

- Caves handmade in Blender.
- Torches - BP_Torch - color: FF2400FF
- Blue/yellow mushrooms
- Rocks

Przeciwnicy - Bat + rockelemental:

lvl_name quantity

1_04	3
1_05	1
1_06	1
1_08	2
1_09	1
1_10	1
1_11	1 BOSS

Punkty do umiejętności w:

1_04
1_05
1_06
1_09
1_11

Puzzle:

1_5 - znajdź kamień na ścianie, podejź do niego i naduś "e", aby uruchomić Level Sequence, który pozwoli wskoczyć na półkę skalną

1_6 - znajdź źródła światła pod sufitem, zapamiętaj ich ilość, podejź do kamieni i wprowadź kod, który odpowiada ilości świateł - czytane kolumnami -> Kod 4 4 2

Map 2 - Charakterystyka:

General color: FF6A49FF

Main assets:

- floor: SM_Crypt_Floor / Floor_Plane_01_64
- walls: SM_Crypt_Wall / Wall_Plane_01_064
- ceiling: SM_Ceiling_Flat /
- passage: SM_Crypt_Doorway / Base_Arch_01
- columns: Column_01
- Torches - BP_Torch - color: FF6A49FF

Przeciwnicy - Skeleton:

lvl_name quantity

2_2	2
2_3	1
2_45	2
2_9	4
2_12	1 BOSS

Punkty do umiejętności w:

2_02
2_03
2_67
2_09
2_12
2_12

Puzzle:

2_45 - znajdź list na kolumnie z podpowiedzią. Podpowiedź "Lonely fire will show you the way". Podpowiedź mówi o samotnej pochodni znajdującej się w narożniku pokoju. Po podejściu do niej i wciśnięciu "e" otworzy się sekretny pokój, w którym jest labirynt. Podczas

jego przechodzenia należy znaleźć listy i nadusić "e". Wtedy otworzy się przejście do kolejnego pokoju

2_67 - znajdź list zlokalizowany na stole, po naduszeniu "e" otworzy się sekretny pokój

2_9 - zawalcz z 4 przeciwnikami

2_11 - należy otworzyć pierwszą ścianę, następnie drugą i cofnąć się. Na środku pojawi się sekretny pokój

Map 3 - Charakterystyka:

General color: BC241AFF

Lava color: BC0A00FF

Skillpoint/Puzzle color (green): 22FF0FFF

Lava - M_Lava_01

Main assets:

- Caves handmade in Blender.
- Passage: Base_Arch_01

Przeciwnicy - Firebat + Rockelemental:

lvl_name quantity

3_2	2
3_4	2
3_5	2
3_6	3
3_7	3
3_9	1
3_11	1 BOSS

Punkty do umiejętności w:

3_04
3_06
3_07
3_08
3_09
3_10
3_11
3_11
3_11

Puzzle:

3_4 - znajdź kamień aby otworzyć sekretny pokój

3_6 - znajdź kamień aby otworzyć sekretny pokój

3_7 - znajdź kamień aby otworzyć sekretny pokój

3_8 - znajdź podpowiedź do zagadki w pokoju 3_10 gdzie należy stanąć na O lub X i nadusić "e". Kolejność według podpowiedzi w pokoju 3_8

3_9 - znajdź kamień aby otworzyć sekretny pokój

Map 4 - Charakterystyka:

General color: 76CEFFFF

Torches - BP_Torch color: 76CEFFFF

green: 22FF0FFF

Main assets:

- Caves handmade in Blender.
- Rocks

Exponential fog:

color: 4683FFFF

density 0.02

Przeciwnicy - Wisp + Bat:

lvl_name quantity

4_2	2	
4_3	1	
4_4	1	
4_5	4	
4_6	1	
4_9	1	
4_10	3	
4_11	1	BOSS

Skill points at:

4_02
4_03
4_05
4_08
4_08
4_09
4_10

Puzzle:

4_1 - przy wyjściu znajdują się pochodnie w różnych kolorach. Kolory są podpowiedzią do zagadki w pokoju 4_3

4_2 - policz O i X na ścianach wzdłuż schodów. Następnie podejź do O i X pod mostem i naduś "e" taką liczbę razy ile znalazło się O i X na ścianie. Zaczynij wpisanie od "O". Opuszczenie strefy wpisywanie resetuje licznik

4_3 - za "małym pokojem" wybierz odpowiednie kolory lamp, odpowiadające tym z pierwszego pokoju

4_5 - zawalcz z 6 przeciwnikami

4_8 - znajdź kolumny, nad którymi miga światło. Następnie znajdź numer znajdujący się po przeciwnej stronie niż ta, na której znajduje się pochodnia na danej kolumnie. Po znalezieniu i zapamiętaniu wszystkich numerów podejź do ściany z łańcuchami i naduś "e" przy łańcuchach odpowiadającym zapamiętanym numerom. Zaczynij liczenie od lewej strony

4_9 - znajdź pochodnię żeby otworzyć sekretny pokój

7. Interfejs użytkownika

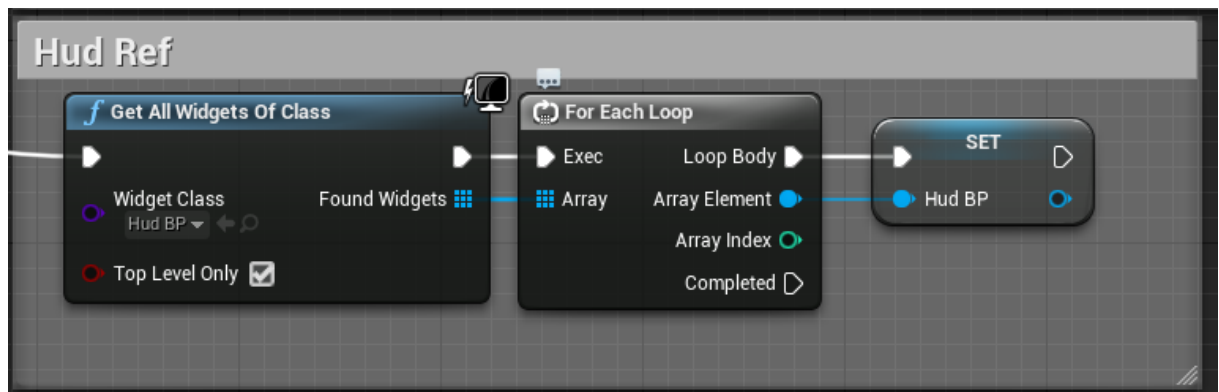
Interfejs użytkownika składa się z:

- Main menu
- Pause menu
- Options (Music and Graphics)
- Health and mana bars
- Inventory
- Skill tree
- Death screen
- Teleport system
- Hints system
- Compass
- Combat mode

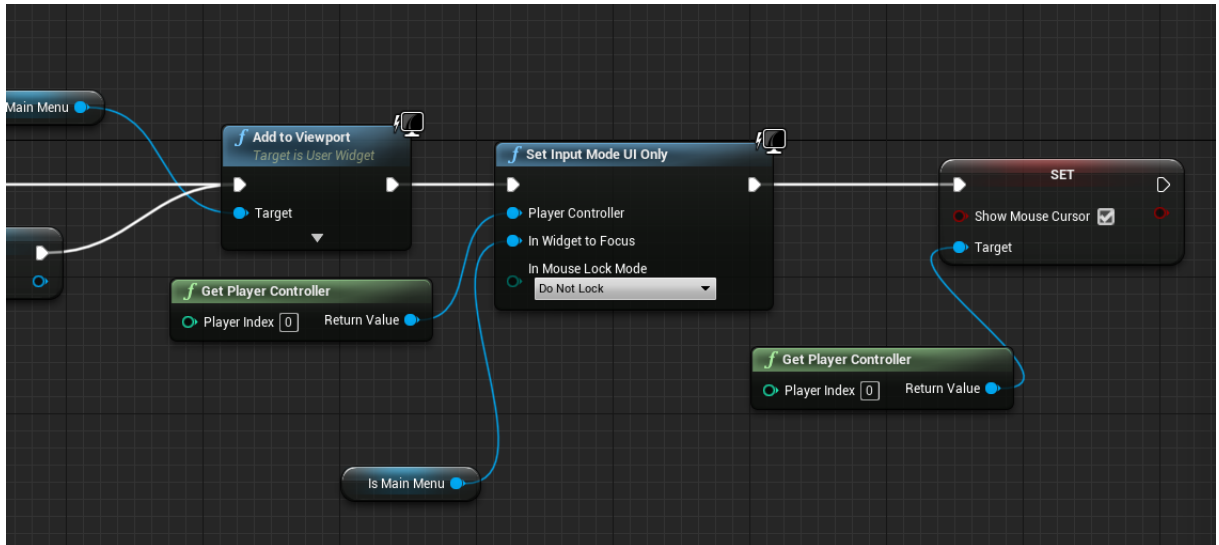
Całość zbudowana jest przy użyciu widżetów.



Edycja przykładowego widżetu.



Przykładowe wywołanie widżetu.



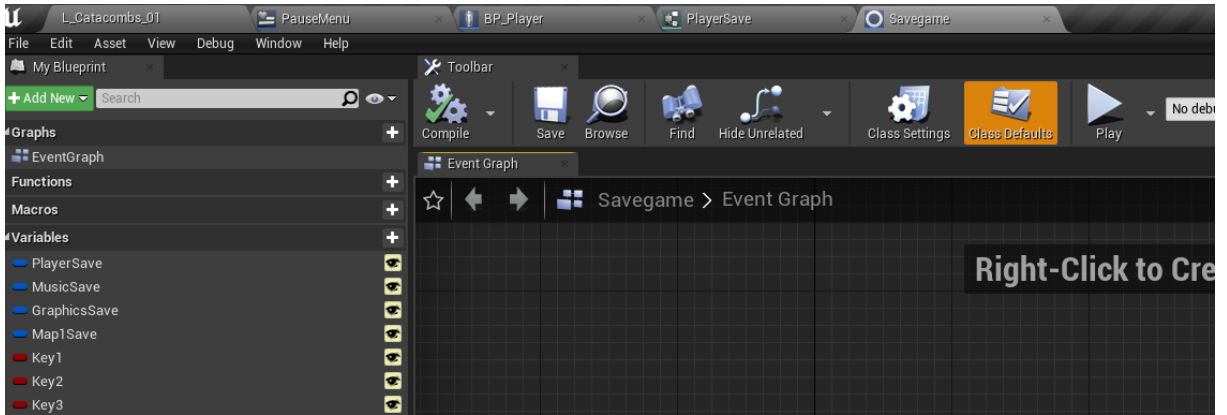
Dodawanie widżetu do widoku. Dla main menu i pause menu widżet dodatkowo należy ustawić Input Mode i pokazać kursor myszy by wybrać konkretną opcję.

8. System zapisów

System zapisów jest oparty o bazową klasę GameStateSave i MyGameInstace oraz o dodatkową klasę Savegame. W Savegame przechowywane są wszystkie elementy gry, które muszą zostać zapisane po zamknięciu z gry.

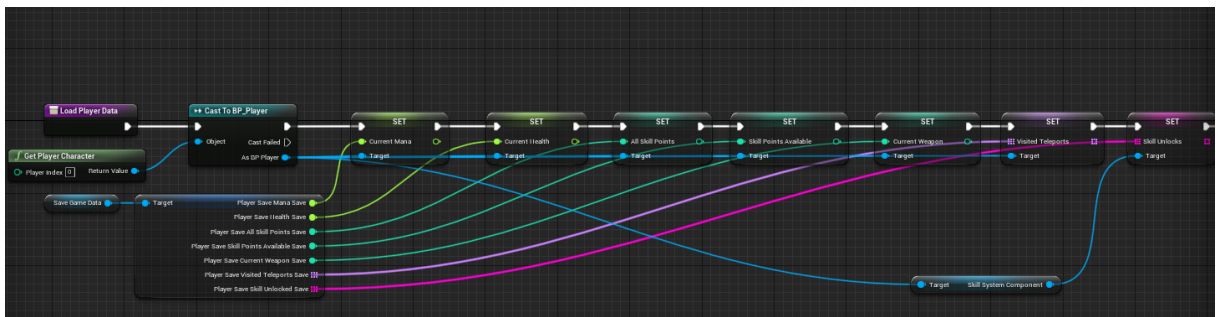


Przykładowa budowa struktury zapisywana w Savegame.

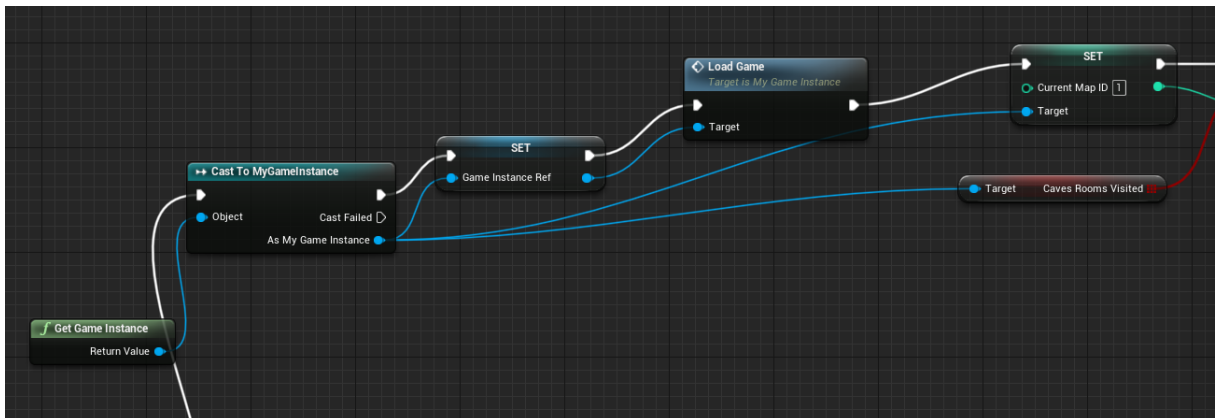


Przykładowe struktury dodane do Savegame.

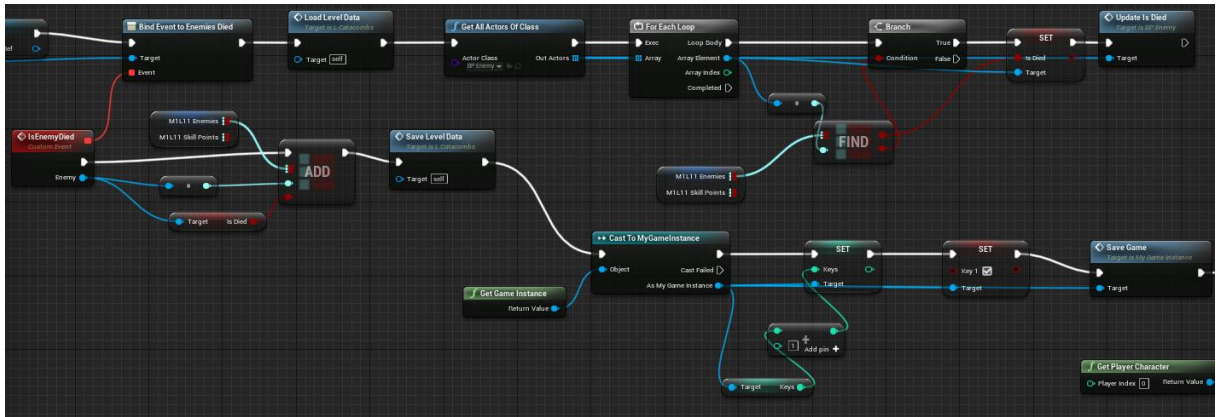
Wszystkie potrzebne zmienne podczas każdego ładowania gry są przekazywane do MyGameInstance, który jest aktywny przez całą rozgrywkę, aż do czasu zapisu.



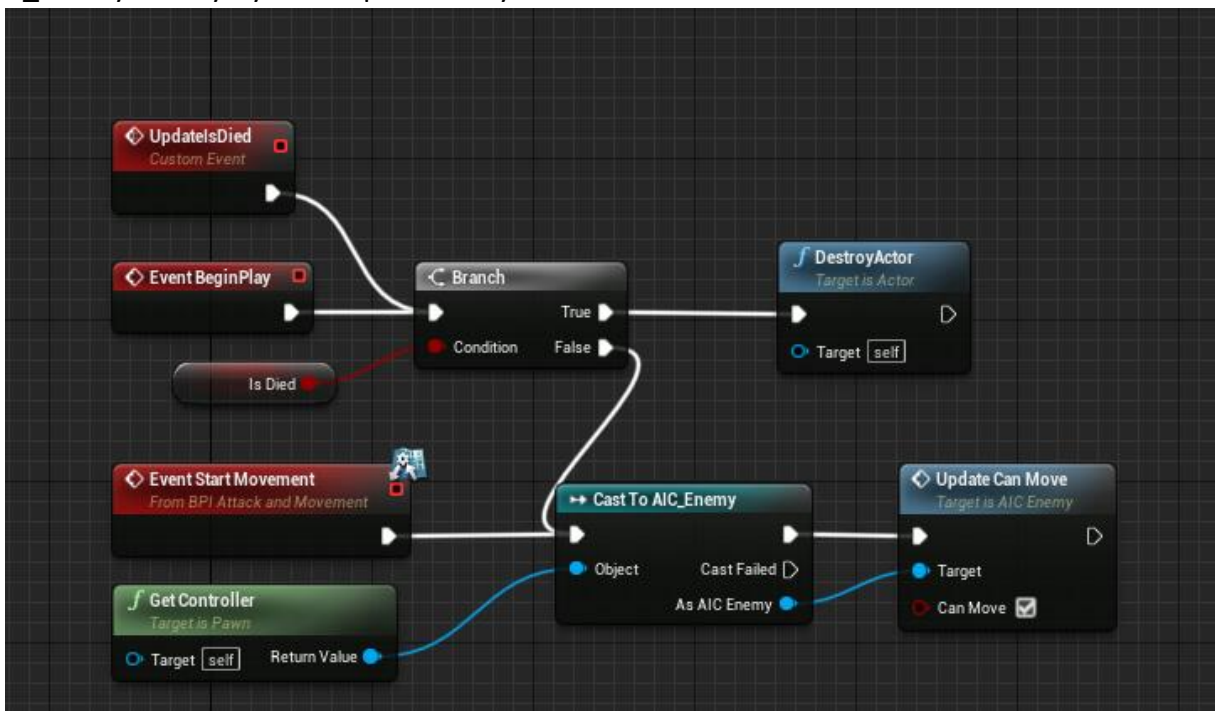
Przykładowe ładowanie danych z Savegame do MyGameInstance zawarte w funkcji LoadPlayerGame.



Przykładowe wywołanie w blueprincie poziomym.

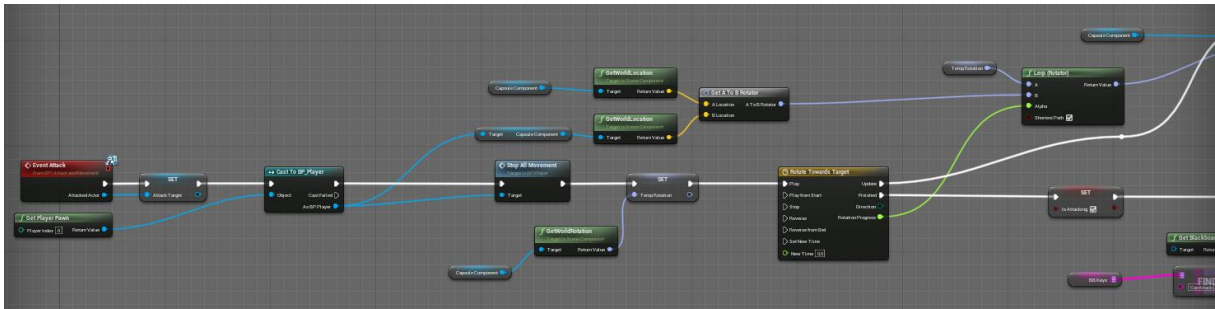


Na każdym poziomie należy sprawdzać stan gry poprzez bindowanie z GameStateSave, w tym przypadku sprawdzamy czy w klasie BP_Enemy zmienna IsDead jest prawdą, jeśli tak to w klasie BP_Enemy niszczy aktor podczas wywołania.

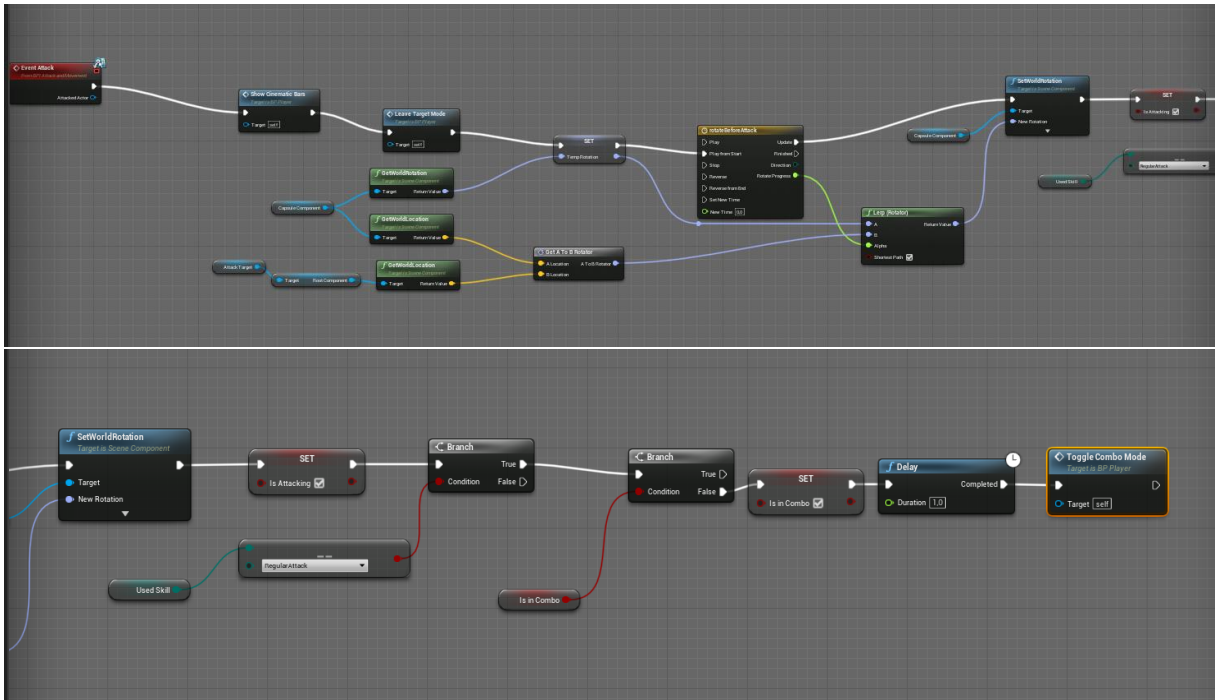


9. System walk

Atakowanie gracza przez przeciwnika i na odwrót odbywa się w większości w Animation Blueprint'ach (Blueprintach Animacji) przy użyciu Anim Notify Eventów. Umieszczone w animacjach ataku anim notify w odpowiednim momencie wykonują odpowiadający im Event w ABP. W przypadku wykonywania ataku, najpierw zatrzymujemy możliwość poruszania się wszystkich przeciwników oraz gracza w pokoju. ABP zawsze ma referencję do BP swojej instancji klasy, więc pobiera z niego referencję do aktora będącego celem ataku. Następnie atakujący wywołuje tzw Interface call na atakowanym, który z kolei przechodzi przez cały proces otrzymania obrażeń i użycia odpowiedniej animacji, która w swoim Anim Notify Eventcie umożliwia na nowo poruszanie się wszystkim postaciom.



Atak w BP_Enemy.



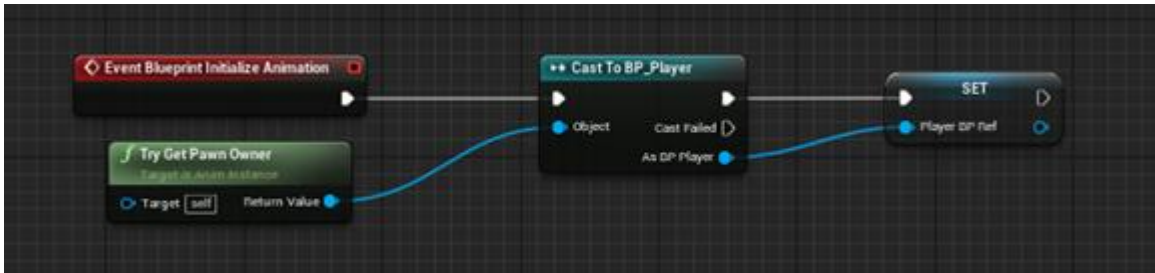
Atak w BP_Player.

System walki zawiera w sobie również drzewko umiejętności i tryb combo.

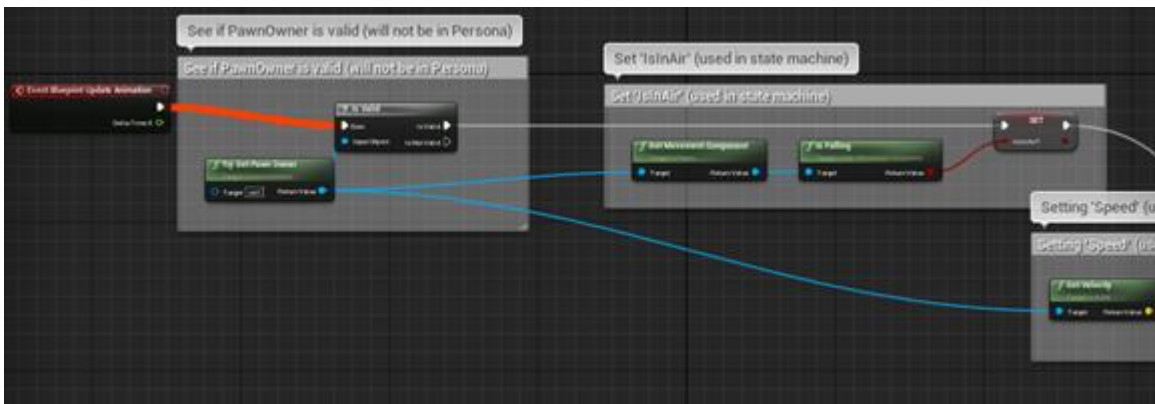
10. Animacje

Komunikacja BLUEPRINT - ANIMATION BLUEPRINT

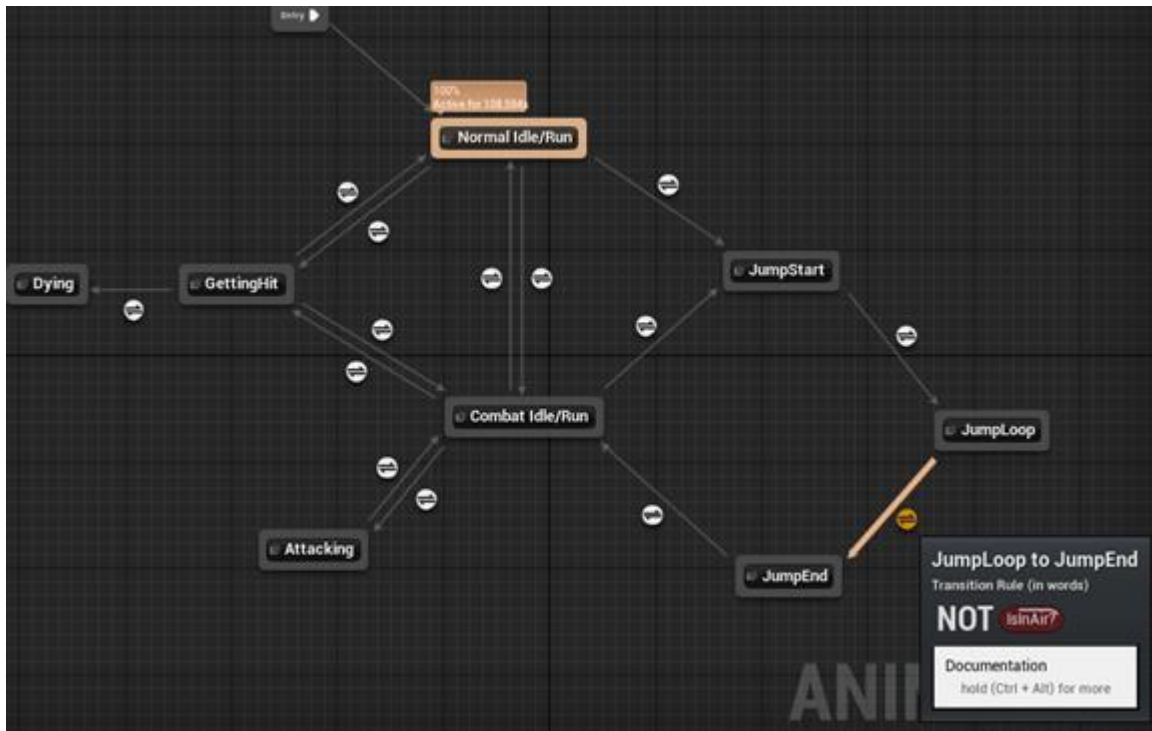
Animation blueprint odpowiada za przełączanie pomiędzy poszczególnymi animacjami skeletal mesha. Przy inicjalizacji pobiera referencję do blueprinta postaci (Event Blueprint Initialize Animation).



Następnie co klatkę pobiera wybrane dane o stanie postaci z jej blueprints i na ich podstawie decyduje jaką animację w danym momencie której animacji użyć. (Event Update AnimationBlueprint)



Samo wykorzystanie danych (głównie boolean i enum) odbywa się w sekcji Anim Graph, gdzie przejścia między animacjami są jasno określone na podstawie wcześniej pobranych danych.



11. AI

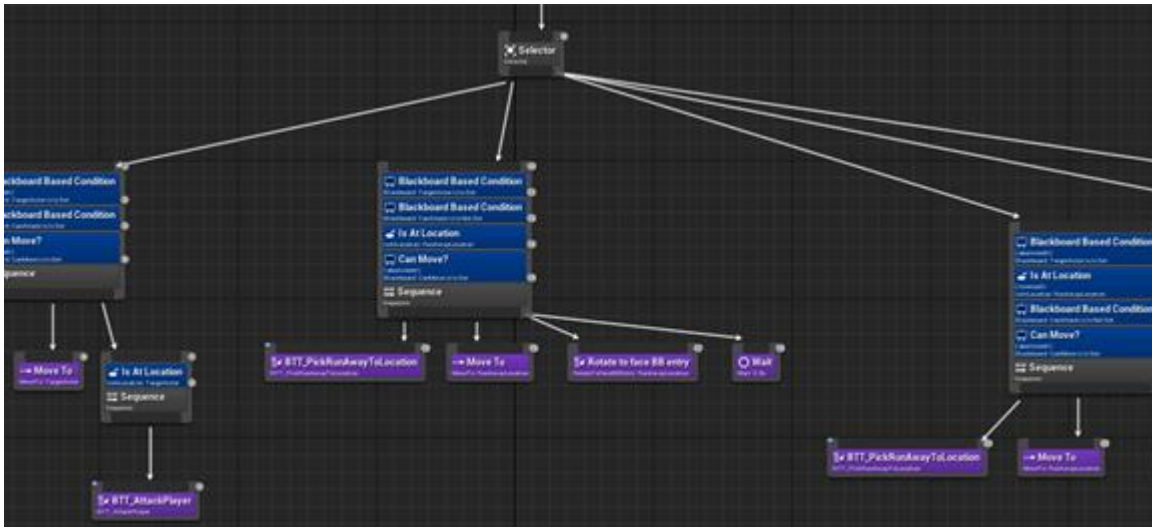
Komunikacja BLUEPRINT - AI CONTROLLER - BEHAVIOR TREE I BLACKBOARD

Blueprint postaci jest dzieckiem klasy Character, czyli specjalnej wersji klasy Pawn, zawierającej w sobie Movement Component. Tak samo jak klasa Pawn, może zostać "opanowany" przez gracza lub AI Controller oraz odbierać input od gracza. W przypadku przeciwników gracza (AI), ostatniej z tych funkcji nie używamy.

AI Controller odpowiada za zachowanie przeciwników na podstawie danych zebranych ze środowiska oraz blueprintu, który kontroluje, wykonując polecenia idące z Behavior Tree.

W Blueprincie każdego przeciwnika umieszczony jest komponent AI Perception służący do zbierania danych ze środowiska na podstawie aktywowanych zmysłów (na przykład: w momencie zauważenia gracza przeciwnik zapisuje sobie referencję do niego, próbuje go gonić, zaatakować lub uciekać od niego.)

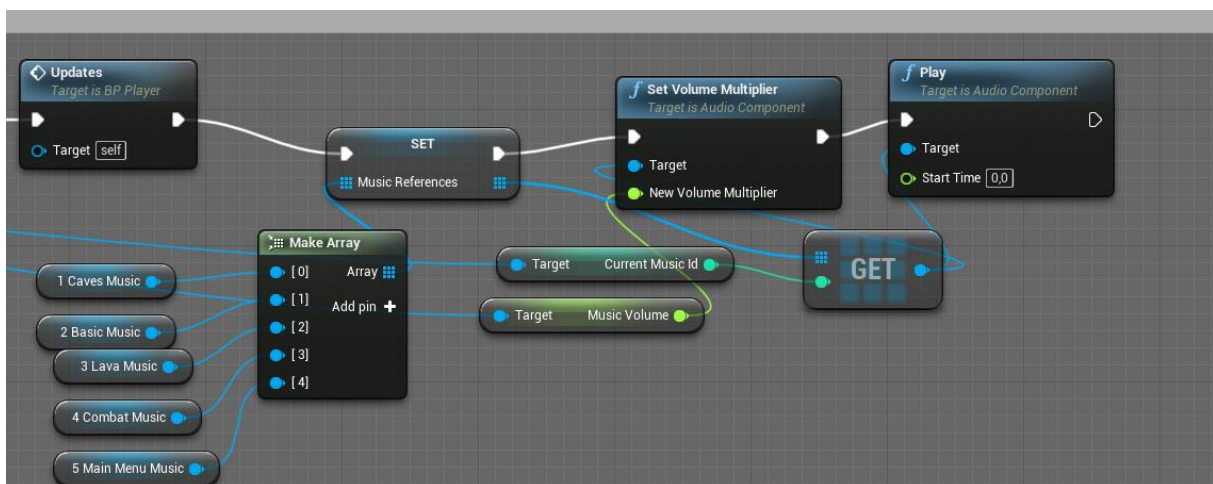
Behavior Tree jest zbudowane z poleceń takich jak "wybierz losowy punktu w zasięgu", "idź do" czy innych, niestandardowych jak na przykład "zaatakuj". Opiera się na danych pobranych z Blackboard'u, które z kolei są aktualizowane poprzez AI Controller. Przeciwnicy w Possessed Redeemer starają się dobiec do gracza, zaatakować go, a następnie zachowywać dystans dopóki nie będą mogły zaatakować ponownie.



12. System muzyki

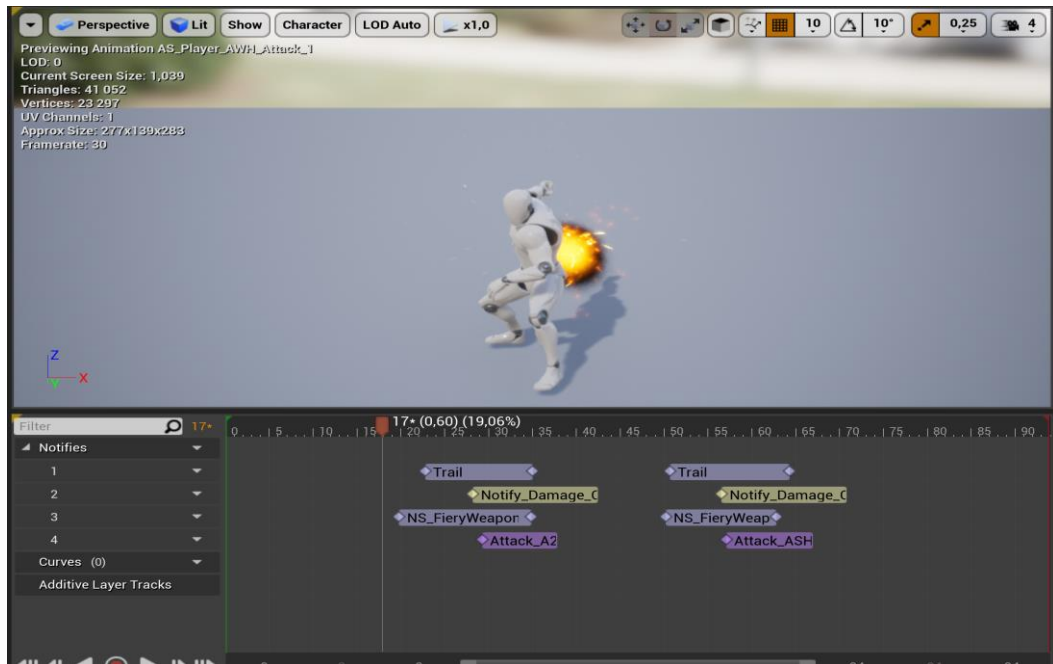
System muzyki zbudowany jest z 5 bazowych utworów:

- CombatMusic
- MainMenuMusic
- CavesMusic
- LavaMusic
- BasicMusic



Inicjalizacja muzyki w grze, w zależności od zapisanego w SaveGame indeksu odtwarza się odpowiedni utwór.

System dźwięków jest zbudowany z kilkadziesiątu odgłosów dołączonych do poszczególnych animacji.



Przykładowa animacja z wstawionym dźwiękiem.