Spis treści:

1. Wstęp

1.1 Badanie Rynku

Branża gier od lat cieszy się pozytywną renomą. W wyniku pandemii zyskała jeszcze więcej zwolenników, ponieważ wiele osób poszukiwało rozrywki w świecie wirtualnym. Szacuje się, że w 2020 roku dział gier wideo wart był 159,3 miliarda dolarów, co oznacza znaczny wzrost o 9,3% w stosunku do 2019 roku. Obecne prognozy przewidują, iż do 2023 roku wartość rynku gier wzrośnie aż do 200 miliardów dolarów. W 2020 roku na całym świecie w gry miało przyjemność grać 2,7 miliarda graczy. Przychody z niezależnych produkcji nie są duże, ponad 50% takich gier nie zarobiło więcej niż 4000$. Najpopularniejszymi gatunkami gier, które odnoszą sukcesy są strategie, symulacje i gry RPG. Największą obecną platformą dystrybucji cyfrowej jest *Steam*. Opublikowanie na niej własnej produkcji jest bardzo proste i wymaga wpłaty 100 dolarów za dowolny tytuł. Opłata ta może zostać zwrócona, gdy gra zarobi 1000 dolarów. Jedną z konkurencyjnych platform dla *Steam'a* jest *Epic Games*, który został uruchomiony w grudniu 2018 roku i od początku zdobył część rynku. W przeciwieństwie do *Steam'a*, *Epic Games* pobiera tylko 12% zarobku z gry zamiast 30%, jednak nie jest on otwartą platformą, na której każdy może zarejestrować i opublikować swoje gry. Wśród polskich użytkowników internetu jest aż 76% graczy. Najczęściej spędzamy od 31-60 minut dziennie delektując się daną produkcją. W grach wideo najlepsze jest to, że każda osoba w dowolnym wieku znajdzie coś dla siebie. W Polsce większość graczy korzysta ze smartfonów i tabletów, ponieważ są przenośne i umożliwiają szybki dostęp do świata wirtualnego. Jeśli chodzi o kupowanie gier, to w większości Polacy dokonują zakupu w sklepie stacjonarnym. Oczywiście branża gier posiada swoich światowych liderów, są nimi *Sony*, *Nintendo* oraz *Microsoft*. Wartość tych korporacji sięga ponad 11 miliardów dolarów. Jeśli chodzi o Polskę, to z roku na rok firmy coraz bardziej się rozwijają i wcale nie odbiegają od tych ,,czołowych gigantów''. Wśród polskich spółek najlepiej funkcjonują *CD Project*, *Techland* oraz *11 Bit Studios*.

1.2 Wybór narzędzi

Unity:

Zintegrowane środowisko Unity umożliwia nam kreowanie dwuwymiarowej gry. Jest ono wyposażone w wiele interaktywnych materiałów, takich jak animacje czy wizualizacje. Silnik Unity został napisany w języku programowania C, C++ (Runtime) i C# (Unity API). Jedną z wielu zalet tego środowiska jest to, iż obsługuje dużo platform sprzętowych, jak choćby Windows, Android czy Linux.

Visual Studio:

Integralną częścią Unity jest program Visual Studio, w którym możemy modyfikować różne skrypty, wybierać motywy, kolory, czcionki i inne ustawienia według naszych potrzeb. Środowisko te pozwala identyfikować problemy, gdy dany skrypt nie jest poprawny. Możliwe jest tworzenie różnego rodzaju metod skryptowych Unity. Korzystając z Visual Studio wykorzystywaliśmy język programowania C#, ponieważ jest on jedynym, obsługiwanym przez Unity.

Asperite:
Asperite pozwala nam na tworzenie animacji 2D i wszelkiego rodzaju grafiki do gier. Program ten jest przydatnym, bardzo intuicyjnym narzędziem dla początkujących twórców, który ułatwia pracę grafikom komputerowym. Asperite daje możliwość używania własnych animacji, grafik, czy zasobów w dowolnym projekcie, dzięki czemu możemy tworzyć kreatywne dzieła do celów osobistych oraz komercyjnych.

1.3 Gatunek Gry/Koncept gry

Produkt należy do gatunku platformowej gry akcji, która koncentruje się na mechanice ruchu postaci, walki i manipulacji otoczeniem. Gra akcji to gatunek wyróżniający się dużą dynamiką prezentowanych wydarzeń oraz elementami, które wymagają od gracza zręczności i szybkiej reakcji. Brak istnienia sztywnych kryteriów definiujących powoduje, iż gry tego gatunku mają ogromną różnorodność formy i treści.
Gra platformowa (pot. platformówka) jest z kolei odmianą gier zręcznościowych,
w których to postać sterowana przez gracza odbywa podróż przez świat gry, poruszając się po wielopoziomowych platformach (stąd nazwa gatunku), którymi mogą być np. chmury, skały, drzewa, piętra budynków itp. Poziomy często urozmaicone są pułapkami lub przeciwnikami, których trzeba ominąć bądź zlikwidować. Dodawane są również różne przedmioty do znalezienia takie jak skrzynki z przedmiotami, waluta, kamienie szlachetne itp. W klasycznej platformówce akcja gry ukazana jest z perspektywy trzeciej osoby (TPP)
i ze statycznym ujęciem kamery, obserwującej poczynania bohatera z boku. Wraz
z postępem technologicznym do gatunku gier platformowych zawitał również trzeci wymiar. W takim przypadku kamera najczęściej umieszczona jest za plecami głównego bohatera. Swoistą syntezę obu podejść stanowi widok 2.5D, który łączy trójwymiarową grafikę
z tradycyjnym ujęciem z boku.
Nasz projekt to platformowa gra akcji 2D w stylu pixel art, której rdzeń rozgrywki skupia się na poruszaniu się postacią po poziomie w celu jego ukończenia, mechanice walki i manipulacji otoczeniem.

1.4 Przegląd podobnych rozwiązań

Super meat boy - gra Super meat boy studia Team Meat jest doskonałym przykładem perfekcyjnie wykonanego sterowania w grze platformowej. Gracz ma pełną kontrolę nad postacią. Sterowanie jest intuicyjne i pozostawia dużo pola do doskonalenia swoich umiejętności, co jest niezwykle ważne w grach platformowych.

Blasphemous - gra studia The Game Kitchen posiada niesamowitą warstwę audiowizualną, która stanowi dla nas inspiracje. Poszczególne poziomy posiadają niespotykaną szczegółowość, a każda z animacji jest niezwykle naturalna dzięki czemu chce się tą grę ukończyć.

Castlevania symphony of the night - gra studia Konami jest niezwykłym przykładem dobrze wyważonej proporcji gameplay-u oraz fabuły. Pomimo ponad 20 lat od wydania, ta gra cały czas jest wzorem dla twórców gier z gatunku Roguelike.

2. Informacje Ogólne

2.1 Movement Postaci

Poruszanie się postaci uznaliśmy za priorytet i główny aspekt naszej produkcji. Chcieliśmy, aby gracz potrzebował chwili na przyzwyczajenie się do naszej postaci, ale nie odkrywał też wszystkich mechanik na początku przygody. Gracz w miarę postępu gry ma za zadanie nie tylko walczyć z przeciwnikami i znajdować ukryte skarby, ale też zgłębiać kombinacje możliwych do zastosowania technik poruszania się, które będą wymagane w dalszych częściach rozgrywki.

Podstawowe poruszanie się

Naszym celem było, aby gracz od samego początku wiedział, że każdy jego ruch jest i musi być precyzyjny, w takim wypadku zastosowaliśmy zwiększanie się naszej prędkości w czasie do docelowej wartości. W ten sposób postać przy pojedynczych kliknięciach klawiszy ruchu poruszała się niezwykle precyzyjnie, dzięki temu gracz miał nad nią pełną kontrolę.

Skok

Sytuacja z mechaniką skoku jest podobna tak jak z poruszaniem się, im dłużej gracz przytrzyma klawisz, tym wyżej nasz bohater podskoczy.

Kucanie

Gracz posiada możliwość także kucnięcia w celu uniknięcia różnych obrażeń, które byśmy otrzymali bez tego mechanizmu - powoduje on zmniejszenie rozmiarów naszego collidera, jednakże nie możemy się wtedy przemieszczać.

Zaawansowane poruszanie się

Gracz w trakcie rozgrywki dostaje możliwość użycia dodatkowych mechanik, które umożliwiają mu dostosowanie swojego stylu gry, oraz dostanie się w niedostępne dla niego wcześniej lokacje.

Dashing

Jest to ślizg naszej postaci w kierunku, który był wcześniej skierowany nasz bohater. Podczas tej mechaniki jesteśmy niewrażliwi na wszelkie ataki, oraz możemy dostać się w miejsca, które nie były wcześniej dla nas dostępne, ze względu na to że byliśmy zbyt “wysocy”, ponieważ ślizg powoduje także zmniejszenie się naszego collidera, podobnie jak w przypadku kucania. Ta umiejętność korzysta z paska wytrzymałości naszej postaci, więc nie jesteśmy w stanie używać jej w nieskończoność, po jednym użyciu musimy poczekać określoną ilość czasu, aby nasz bohater mógł się zregenerować.

Walljump

Poziomy w naszym zamyśle będą posiadać miejsca, w których będzie istniała konieczność odbijania się od ścian, aby dostać się do konkretnej lokacji. Z pomocą przyjdzie nam kolejna zaawansowana mechanika - Walljump. Bohater jest w stanie skoczyć na ścianę i się jej złapać, jednakże aby zapobiec trzymania się jej w nieskończoność, dodaliśmy efekt ześlizgiwania się z niej. Gracz będzie musiał wykorzystać zdobytą wiedzę podstawowego poruszania się i wyczucia czasu, aby dostać się metodą walljumpu w docelowe lokacje. Mechanika ta jest też przydatna przy walce z przeciwnikami, ponieważ w momencie wskoczenia na ścianę, można się od niej odbić i wylądować za plecami wroga i zdobyć w

ten sposób nad nim przewagę. W celu użycia tej umiejętności należy wskoczyć na ścianę, klawiszem ruchu nadać kierunek i wykonać skok ponownie.

2.2 Pułapki

W grze znajduje się kilka rodzajów pułapek, w ramach których istnieją ich słabsze lub silniejsze wersje. Każdy wariant charakteryzuje się odmiennymi statystykami dotyczącymi punktów obrażeń, opóźnienia w ich zadawaniu oraz czasu zatrzymania movementu naszej postaci. Do tej pory zaprojektowano poniższe rodzaje pułapek:

- **Kolce** (Pułapka zadaje niskie obrażenia gdy gracz na nią wpadnie. Zadaje tylko obrażenia i zatrzymuje gracza na krótką ilość czasu)
- **Rosiczka** (Pułapki zadające średnie obrażenia graczowi gdy na nie wpadnie. Posiadają krótkie okno czasowe w którym gracz może przejść bezpiecznie gdy rosiczka jest schowana w ziemi)
- **Kula** (Pułapka zadającę duże obrażenia i odrzucająca naszego gracza na pewny dystans. Posiada najwyższy czas blokady ruchu naszego gracza przy uderzeniu.)
- **Laser** (Pułapka zadająca średnie obrażenia. Lasery pojawiają się w konkretnych miejscach na krótki okres czasu, dając graczowi czas na przebiegnięcie przez niebezpieczny dla niego obszar)
- **Strzały** (Pułapki strzelające strzałami co jakiś czas i zadające niskie obrażenia naszemu graczowi. Strzały wystrzeliwane są w takich przerwach by dać graczowi możliwość przeskoczenia nad nimi by uniknąć obrażeń)

Tak prezentują się stworzone pułapki do tej pory. W trakcie dalszych prac nad grą powstaną kolejne pułapki np. wybuchające przedmioty, krople kwasu/trucizny, przeszkody teleportujące naszego gracza do tyłu lub tzw. znajdźki pułapki - czyli przedmioty przypominające inne przedmioty do podniesienia, z tym, że zadają graczowi obrażenia po podniesieniu.

2.3 Przeciwnicy

Obecnie w grze istnieją dwa rodzaje przeciwników: walczący w zwarciu używając broni białej i walczący na dystans przy użyciu łuku. Ich działanie bazuje na maszynie stanów.

Przeciwnik walczący w zwarciu

Stan początkowy to *Moving State*, którym jest przemieszczanie się przeciwnika w prawo i w lewo. Możliwe są dwa przejścia ze stanu początkowego:

- jeżeli przeciwnik napotka krawędź platformy lub ścianę, przejdzie do *Idle State*, w którym przez krótką chwilę (1-2 sekundy) zatrzyma się w miejscu wykonując animację bezczynności (*Idle*), po czym odwróci się i przejdzie ponownie do *Moving State*;
- jeżeli postać gracza wejdzie w zasięg wykrywania przeciwnika, przejdzie on do *Player Detected State*, w którym wykona się krótka animacja przyjęcia postawy bojowej przez przeciwnika. Przejście to jest również możliwe z *Idle State*.

Z *Player Detected State* możliwe są trzy przejścia:

- jeżeli postać gracza opuści zasięg wykrywania przeciwnika, nim ten zdąży wykonać jakąkolwiek inną akcję, wróg przejdzie do *Look For Player State*, w którym zacznie się obracać “szukając” postać gracza. Jeżeli nadal nie będzie on w zasięgu wykrywania, następuje przejście do *Moving State*. W innym wypadku mamy powrót do *Player Detected State*;

- jeżeli postać gracza nie jest w zasięgu ataku, przeciwnik przejdzie do *Charge Player State*, tzn. zacznie biec w jego kierunku. Jeśli postać gracza znajdzie się w zasięgu ataku, przeciwnik przejdzie do *Attack Player State*. Natomiast, gdy minie określona ilość czasu o od momentu przejścia stanu i jeśli postać gracza pozostanie w zasięgu wykrywania, przeciwnik przejdzie z powrotem do *Player Detected State*. W przypadku, kiedy postać gracza opuści zasięg wykrywania wroga, nastąpi przejście do *Look For Player State*;
- jeżeli postać gracza jest w zasięgu ataku przeciwnika, przejdzie on do *Attack Player State*. Wykona on animację ataku i jeśli gracz nie zdoła go uniknąć, jego postać otrzyma określoną liczbę obrażeń.

Z *Attack Player State* możliwe są trzy przejścia:

- jeżeli postać gracza po wykonanej animacji pozostanie w zasięgu ataku, następuje przejście własne;
- jeżeli postać gracza po wykonanej animacji opuści zasięg ataku przeciwnika, ale pozostanie w zasięgu wykrywania, nastąpi przejście do *Player Detected State*
- jeżeli postać gracza opuści zasięg ataku oraz zasięg wykrywania przeciwnika, nastąpi przejście do *Look for Player State*.

Jeżeli gracz zada przeciwnikowi pewną ilość obrażeń, to z każdego z wymienionych wyżej stanów przejdzie on do *Stun State*. Przez krótką chwilę wróg będzie stał w miejscu wykonując animację oszołomienia. Po zakończeniu animacji możliwe są dwa przejścia:

- jeżeli gracz jest w zasięgu wykrywania, nastąpi przejście do *Charge Player State*
- jeżeli gracz nie jest w zasięgu wykrywania, nastąpi przejście do *Look For Player State*.

W sytuacji, kiedy gracz zadał przeciwnikowi obrażenia większe od maksymalnej ilości jego punktów życia, nastąpi przejście do *Dead State*. Wykona się animacja śmierci, po czym przeciwnik zniknie.

Przeciwnik walczący na dystans

Stan początkowy to *Moving State*, którym jest przemieszczanie się przeciwnika w prawo i w lewo. Możliwe są dwa przejścia ze stanu początkowego:

- jeżeli przeciwnik napotka krawędź platformy lub ścianę, przejdzie do *Idle State*, w którym przez krótką chwilę (1-2 sekundy) zatrzyma się w miejscu wykonując animację bezczynności (*Idle*), po czym odwróci się i przejdzie ponownie do *Moving State*;
- jeżeli postać gracza wejdzie w zasięg wykrywania przeciwnika, przejdzie on do *Player Detected State*, w którym wykona się krótka animacja przyjęcia postawy bojowej przez przeciwnika. Przejście to jest również możliwe z *Idle State*.

Z Player Detected State możliwe są trzy przejścia:

- jeżeli postać gracza znajduje się w maksymalnym zasięgu wykrywania i animacja *Player Detected State* zakończyła się, to następuje przejście w *Ranged Attack State,* czyli wróg strzela z łuku. Jeśli po wykonanej akcji postać gracza nadal znajduje się w maksymalnym zasięgu wykrywania, przeciwnik przechodzi ponownie do *Player Detected State*. W innym wypadku przeciwnik przechodzi do *Look For Player State*,
- jeżeli postać gracza znajduje się w bliskim zasięgu ataku, przeciwnik przejdzie do *Dodge State,* tzn. odskoczy od gracza na określoną odległość, gdy dzieje się to pierwszy raz w tym cyklu. Jeśli postać gracza wejdzie w zasięg bliskiego ataku, to przeciwnik przechodzi do *Melee Attack State,* czyli atakuje wręcz. Natomiast, gdy postać gracza wejdzie w zasięg dalekiego ataku, przeciwnik przechodzi do *Ranged*

Attack State. W momencie, gdy postać gracza nie znajduje się w maksymalnym zasięgu wykrywania, przeciwnik przechodzi do *Look For Player*,
- jeżeli postać gracza znajduje się blisko wroga i nastąpiło wcześniej przejście do *Dodge State*, przeciwnik przechodzi do *Melee Attack State*. Jeśli postać gracza nie jest w maksymalnym zasięgu wykrywania, to wróg przechodzi do *Look For Player State,* w innym wypadku następuje przejście do *Player Detected State*.

Z Look For Player State możliwe są dwa przejścia. Jeżeli postać gracza znajduje się w zasięgu wykrywania, to przeciwnik przechodzi do *Player Detected State*, w przeciwnym wypadku przechodzi do *Moving State*.
Jeżeli gracz zada przeciwnikowi pewną ilość obrażeń, to z każdego z wymienionych wyżej stanów przejdzie on do *Stun State*. Przez krótką chwilę wróg będzie stał w miejscu wykonując animację oszołomienia. Po zakończeniu animacji możliwe są dwa przejścia:
- jeżeli gracz jest w zasięgu wykrywania i nie jest w zasięgu bliskiego ataku, to nastąpi przejście do *Ranged Attack State,* w innym wypadku przeciwnik przejdzie do *Dodge State,*
- jeżeli gracz nie jest w zasięgu wykrywania, nastąpi przejście do *Look For Player State*.

W sytuacji, kiedy gracz zadał przeciwnikowi obrażenia większe od maksymalnej ilości jego punktów życia, nastąpi przejście do *Dead State*. Wykona się animacja śmierci, po czym przeciwnik zniknie.

2.4 Fabuła

Głównym bohaterem historii jest podróżnik, który przy pomocy magicznego artefaktu pokonuje granice międzywymiarowe i odwiedza różne światy/rzeczywistości. Na początku gry zostaje on zaatakowany przez tajemniczą kreaturę, która pragnie posiąść ów artefakt. W trakcie walki zostaje on jednak przez przypadek zniszczony, co skutkuje powstaniem tunelu między-wymiarowego, który wciąga nieprzytomnego wędrowca i potwora.
Po przebudzeniu bohater odkrywa, iż przeniósł się do innego świata. Pojawia się wtedy duch, który zaklęty był w artefakcie przez jego twórcę i jest on źródłem mocy artefaktu. Duch ten jest połączony z duszą bohatera i będzie towarzyszył graczowi przez całą grę pełniąc rolę przewodnika. Gracz dowiaduję się wtedy, iż ma za zadanie odnalezienie wszystkich części artefaktu rozsianych po świecie, który zdeformowany został przez magię artefaktu spaczoną demoniczną mocą demona, którego należy pokonać na końcu gry.
Na świat gry składa się kilka lokacji, które są alternatywnymi rzeczywistościami powstałymi z fragmentów świata, który się rozpadł przez spaczoną moc artefaktu. Każda lokacja ma pewną ilość poziomów. Na końcu każdej znajduje się boss (unikatowy i szczególnie wymagający przeciwnik w grach komputerowych, którego pokonanie zazwyczaj oznacza zwieńczenie pewnego etapu gry), po pokonaniu którego odzyskujemy jeden z fragmentów artefaktu. Pełną historię artefaktu i demona poznajemy wraz ze znalezieniem kolejnych jego fragmentów. Powoduje to stopniowe odzyskiwanie pamięci ducha zaklętego w artefakcie. Na końcu gry gracz wraca do centralnej lokacji, gdzie czeka demon. Zaczyna się walka z ostatecznym bossem. Po jego pokonaniu demoniczna moc spaczająca rzeczywistość, w której się wędrowiec znajduje, znika. Świat powraca do swojej pierwotnej formy, a bohater spokojnie wyrusza w kolejną podróż.

2.5 Interakcja z otoczeniem i atak

System interakcji pozwala graczowi oddziaływać na świat gry. Na chwilę obecną obiekty, z którymi gracz może wejść w interakcje to:

1. Skrzynia - gracz może ją otworzyć i zamknąć
2. Dźwignia - pociągnięcie jej wywołuje określony efekt np. otwiera drzwi
3. Drzwi – przejście przez nie przenosi gracza w określone miejsce

System działa przy użyciu raycast'ów, wokół naszej postaci rysowany jest prostokąt, który służy jako ich zasięg. Jeżeli odpowiednie obiekty (posiadające jako komponent, skrypt który dziedziczy z klasy Interactable) znajdą się w zasięgu Player'a, dodawane są do tablicy typu RaycasHit2D. Następnie sprawdzana jest długość tej tablicy, jeśli jest większa od 0 to wykonuje się pętla foreach, która wywołuje funkcję Interact() danego obiektu.

Gracz ma też możliwość walki z przeciwnikami, poprzez wykonywanie ataków. Po naciśnięciu odpowiedniego przycisku uruchamia się animacja ataku, a obiekty w zasięgu Player'a, które mogą zostać zaatakowane otrzymują obrażenia

Dodatkowe stworzyliśmy również system destrukcji, który pozwala na niszczenie wybranych obiektów. Po tym jak wykryją, że gracz je zaatakował ulegają zniszczeniu.

2.6 User Interface

Obecnie nasz interface podczas rozgrywki składa się z paska życia, paska kondycji i aktualną ilość „Gem'ów”, czyli waluty w naszej grze. Naciśnięcie klawisza „esc” zatrzymuje grę i uruchamia menu, w którym możemy: wznowić rozgrywkę, przejść do menu głównego, wyjść z gry i zresetować poziom. Menu główne pozwala na wczytanie poziomu, wejście w opcje i wyjście z gry.

3. Level Design

3.1 Pixel Art

W naszym projekcie zdecydowaliśmy wykorzystać grafikę 2d w technice pixel-art w rzucie nie izometrycznym. Pixel art jest sposobem tworzenia grafiki rastrowej w oprogramowaniu pozwalającym wykorzystać pojedyncze piksele. Zwykle, grafiki pixel-artowe tworzone na potrzeby gier składają się z ograniczonej liczby kolorów w palecie. Zastosowanie ograniczonej palety barw ma szereg zastosowań:

- zastosowanie artystyczne - Grafiki tworzone w technice pixel artowej mają charakterystyczny klimat, dla odbiorców mają nostalgiczny charakter z uwagi na stosunkowo niską rozdzielczość(przypomina odbiorcom gry z epoki 8 i 16 bitowych konsol)
- zastosowanie praktyczne - Tworzenie grafik pixel artowych jest mniej czasochłonne oraz mniej kosztowne od tworzenie grafik 3d, dzięki czemu wiele niezależnych twórców sięga po tą technikę tworzenia grafik.

3.2 Assety

Nasz projekt składa się w większości z autorskich assetów tworzonych wyłącznie na potrzeby naszej gry. Do tworzenia grafik oraz animacji wykorzystujemy program Aseprite

wraz z szeregiem dodatkowych skryptów w języku lua.Dodatkowo, w procesie tworzenia animacji wykorzystujemy:

- **Bitmapflow** - do tworzenie klatek pośrednich animacji, dzięki czemu graficy mogą rysować grafiki w mniejszej ilości klatek animacji.
- **Pixel fx designer** - do tworzenia proceduralnych efektów cząsteczkowych
- **SmearFX** - w celu dodania tzw.”Smear Frames”, czyli klatek animacji w których obraz jest celowo zniekształcony, dla uzyskania wrażenia ruchu.
- **Juice fx** - do tworzenia powtarzalnych animacji np.niszczenia obiektu, pojawienia się obiektu itp.
- **Tilesetter** - do generowania Rule Tile, czyli łączenia poszczególnych grafik poziomu w taki sposób, aby łatwo i wygodnie tworzyć design levelu bezpośrednio w silniku Unity.

3.3 Oświetlenie

W celu dodania oświetlenia potrzebowaliśmy dodania osobnej paczki Universal RP. Jest to oskryptowany rendering stworzony przez Unity. URP zapewnia przyjazne dla artystów przepływy pracy, które pozwalają szybko i łatwo tworzyć zoptymalizowaną grafikę na wielu platformach, od urządzeń mobilnych po wysokiej klasy konsole i komputery PC.

Oświetlenie globalne

Na początku pracy z oświetleniem potrzebowaliśmy światła podstawowego, który definiuje oświetlenie całego poziomu. Dzięki niemu można dostosować klimat lokacji, jak i odwzorować rzeczywisty stan światła, który by panował w np. lochach.

Oświetlenie punktowe

Jest najczęstszym typem oświetlenia, który wykorzystujemy w naszej produkcji. Właściwości jakie możemy w nim edytować to na przykład:

- Promień zewnętrzny światła
- Promień wewnętrzny światła
- Natężenie
- Barwa
- Intensywność opadania

Wartym uwagi jest też możliwość ustawienia warstwy, z którą ma kolidować konkretne światło, co jest bardzo pomocne w kwestii mieszania się światła globalnego i punktowego.
Oświetlenie punktowe jest wykorzystywane w:

- Pochodniach
- Oknach
- Postaci
- Niektórych przeszkodach

3.4 Audio Design

W naszym projekcie, do tworzenia muzyki oraz efektów dźwiękowych jako Digital Audio Workstation wykorzystujemy program Fl-Studio 20.

Używane syntezatory:

- Zebralette
- Fl Keys
- Magical8bitplug
- 3xOsc

Używane SoundFonty:

- SGM
- EarthBound

4. Testy

4.1 Założenia Testów

Głównym założeniem testów jest sprawdzenie jakości rozgrywki i części wizualnej produktu, a także przetestowanie zaimplementowanych mechanik.

Założeniami przeprowadzonych testów były:

- wyłapanie złych mechanik, błędów czy też innych niezauważonych wcześniej problemów,
- zebranie informacji, które pozwoliłyby ulepszyć produkcję,
- sprawdzenie czy koncept projektu trafia w ogólne gusta graczy.

4.2 Przebieg Testów

Aby wykryć potencjalne błędy i bugi potrzebowaliśmy w miarę możliwości dużej liczby testerów. Do pomocy poprosiliśmy zaprzyjaźnionego streamera, aby na swojej transmisji przetestował naszą produkcję. Dzięki temu mieliśmy znacznie zwiększoną szansę na poznanie opinii szerszej publiki, jak i zapoznanie się z odbiorem naszego projektu w aktualnym stadium rozwoju.

Rezultaty testów przeszły nasze oczekiwania i już po jednym dniu udostępnionej pre-alphy, dostaliśmy kilkadziesiąt recenzji na takie tematy jak:

- Fabuła
- Rozgrywka
- Oprawa audiowizualna
- Poziom trudności
- Bugi i glitche

Dzięki tym testom mogliśmy dużo sprawniej i szybciej rozwijać projekt, co za tym idzie wynieść jakość produktu na wyższy poziom.

4.3 Wpadki/Bugi/Błędy logiczne

W projekcie wystąpiły liczne błędy, o których dowiedzieliśmy się od osób, testujących naszą grę.

Napotkane błędy:

- mogliśmy wielokrotnie zniszczyć jedną skrzynkę,
- wystąpiły problemy z animacją naszej postaci przy ścianach,
- na mniejszych platformach bohater często się ześlizgiwał, co było irytujące,

- po otrzymaniu obrażeń od kuli na łańcuchu, mogliśmy nadal poruszać się postacią w powietrzu i skakać w stanie animacji ,,damage/hurt'',
- nasz bohater ślizgał się przy kucaniu i atakowaniu,
- przeciwnicy zostali źle skonstruowani, walka z nimi była bardzo losowa,

Nietrafione pomysły:

- nasz bohater posiadał zbyt szybką animację, gdy stał w miejscu,
- połączenie grafiki pixel art z systemem particle, którego cząstki nie były przystosowane do tej grafiki wyglądało nieestetycznie,
- nieintuicyjny układ sterowania postaci,
- animacja po otrzymaniu obrażeń przez naszego bohatera była nieczytelna,
- zła konstrukcja planszy, gracz nie miał pomysłu w którą stronę się udać,
- gdy przeciwnik otrzymywał obrażenia, nie było żadnej animacji która to potwierdzała

4.4 Ogólne Wrażenia

Spośród kilkudziesięciu recenzji testerów wybraliśmy najczęstsze wypowiedzi i opinie na temat projektu:

- Zamysł wykonania gry spełnia oczekiwania
- Fabuła wydaje się ciekawa i pozostawia chęć poznawania dalszej części historii
- Atrakcyjny poziom trudności, gra łatwa do nauczenia, ciężka do pełnego opanowania
- Przejście testowanego poziomu powodował poczucie satysfakcji
- Czas przejścia poziomu wahał się od 10 minut (gracz zaawansowany) do 40-45 minut (gracz niedzielny)
- Gracze byli zadowoleni z możliwości wyboru grania na klawiaturze, lub na kontrolerze

5. Kosztorys

5.1 Pracownicy

Niniejsze kwoty przedstawiają koszt zatrudnienia konkretnych pracowników na dany okres powstawania projektu.

Junior Deweloperzy:

Zakładamy, że czas powstawania projektu to 6 miesięcy, kiedy to nasi developerzy będą pracować nad grą. Patrzą na średnie zarobki w Polsce jako Junior Deweloperzy wychodzi nam poniższy koszt:

Sześciu Junior Deweloperów * 4000 zł * 6 miesięcy = 144 000 zł

2D Artist/Audio:

Artysta od ścieżki audiowizualnej będzie pracował równomiernie z Deweloperami zaopatrując ich w nowe assety, które mogą zostać natychmiast zaimplementowane do projektu. Okres pracy Artysty będzie podobny co Junior Deweloperów.

Jeden Artysta od ścieżki Audiowizualnej * 5500 zł x 6 miesięcy = 33 000 zł

Testerzy:
Testerzy będą testować naszą grę gdy będzie ona już nadawać się do fazy testów. Okres intensywnego testowania naszej produkcji przypada na dwa ostatnie miesięcy prac nad samą grą. Do testowanie naszej gry zatrudnilibyśmy dwóch testerów, za średnią płacę jak na to stanowisko.

Dwóch Testerów * 3300 zł x 2 miesiące = 6600 zł

Za utrzymanie naszych pracowników przez cały planowany okres pracy nad grą zapłacilibyśmy **183 600 zł**

5.2 Licencja na silnik

Wybierając silnik Unity, musimy spojrzeć na licencję samego silnika. Na podstawie funduszy bądź zysków jakie pozyskamy ze sprzedaży naszego produktu opłaty licencyjne mogą się różnić. Dysproporcja cenowa pomiędzy konkretnymi licencjami jest następująca:

Jeśli przychody lub finansowanie mieszczą się poniżej 100 000 USD w ciągu ostatnich 12 miesięcy - Licencja Personal Free, za którą nie musimy umieszczać żadnych opłat

Jeśli przychody lub finansowanie mieszczą się poniżej 200 000 USD w ciągu ostatnich 12 miesięcy - Licencja Plus $399 za siedzenie (W naszym przypadku $2394, ze względu na ilość programistów).Jest to kwota którą należy płacić co rok.
Jeśli przychody lub finansowanie mieszczą się powyżej 200 000 USD w ciągu ostatnich 12 miesięcy - Licencja Pro $1,800 za siedzenie. (W naszym przypadku $10800, ze względu na ilość programistów).Jest to kwota którą należy płacić co rok.

Istnieje jeszcze licencje Enterprise, która za kwotę $2000 rocznie oferuje 10 siedzeń. Jest to idealna propozycja dla większych firm.

5.3 Oprogramowanie

Oprócz Unity do stworzenia naszej gry używamy Visual Studio 2019 w wersji Community. Na potrzeby tego projektu darmowa wersja w zupełności wystarczy. Inną kategorią oprogramowania, którego używamy są aplikacje do tworzenia grafiki. Do tej pory korzystaliśmy z:

1. Aseprite
2. Pixel FX Designer
3. JuiceFX
4. SmearFX

Całkowity koszt tych programów dla dwóch osób to 480 zł.

5.4 Sprzedaż produktu

Plan sprzedaży zakłada w pierwszej kolejności wypromowanie naszej gry i dotarcie do świadomości graczy. W tym celu najpierw zamierzamy udostępnić demo naszej produkcji na Itch.io. Jest to strona internetowa, na której użytkownicy mogą hostować i sprzedawać swoje gry niezależne. Wydanie gier w ten sposób jest powszechnie uznawane za dobre podejście dla nowych twórców.

Następnie planujemy przeznaczyć 10000 zł na cele promocyjne, takie jak zakup reklam na różnych platformach społecznościowych. Mimo niskiego można by się wydawać nakładu, liczmy na przebicie się za pomocą tzw. „marketingu wirusowego". Polega on na zainicjowaniu sytuacji, w której potencjalni klienci będą sami udostępniać między sobą informacje dotyczące naszej gry.

Ostatnim krokiem naszego planu jest wydanie gry na platformie Steam. Proces ten składa się z kilku kroków:

1. uiszczenie opłaty wydawniczej wynoszącej 100 $
2. wypełnienie dokumentacji cyfrowej
3. stworzenie strony produktu w sklepie i określenie ceny
4. oczekiwanie na pozytywną weryfikację

Proces ten może potrwać minimum 1 miesiąc.

Cena naszej gry będzie wynosić 10€. Po opłaceniu podatków i odliczeniu kwoty, którą Steam pobiera od każdej transakcji, zysk z jednej sprzedanej sztuki wynosi 24.15 zł. Koszt całego projektu wynosi 194 448 zł. Aby koszty projektu się zwróciły musimy sprzedać 8052 sztuk.

6. Dalszy Rozwój Projektu

6.1 Mechanika Spaczenia

Każdy przeciwnik i pułapki docelowo będą posiadały charakterystyczne dla nich cechy takie jak ilość punktów życia,zadawane obrażenia, szybkość ataku oraz szybkość poruszania, w zależności od rodzaju przeciwnika lub pułapki. Gracz włączając grę i pojawiając się na mapie 'uruchamia' timer odliczający czas według którego będzie skalowany poziom trudności. Im dłużej gracz będzie grał w naszą grę tym większe statystyki potworów oraz pułapek będą się stawać. Ich obrażenia, szybkość, czy punkty życia będą ulegać ciągłemu wzrostowi. Oprócz przeciwników z czasem otoczenie po jakim porusza się gracz będzie wyglądało na coraz bardziej 'spaczone', czy też niepokojące.
Nasz gracz będzie musiał nie tylko walczyć z przeciwnikami, ale także i z czasem. Główny bohater jednak nie jest bez szans. Za pomocą specjalnych przedmiotów, które gracz będzie mógł zdobyć tylko w konkretnych miejscach, po pokonaniu bardzo trudnych przeszkód/zagadek/przeciwników, będzie mógł przywrócić świat do normalnego stanu, tym samym zmieniając statystyki przeciwników oraz pułapek z powrotem do pierwotnego stanu. Podobne zmiany zajdą także w wyglądzie planszy.

Jednak gracz nie będzie mógł bezpiecznie resetować poziomu trudności w nieskończoność. Liczba przedmiotów jakie gracz będzie mógł do tego użyć będzie bardzo ograniczony, a samo używanie przedmiotów, za każdym razem powoduje, że poziom trudności będzie wzrastał jeszcze szybciej niż miało to miejsce wcześniej. Mechanika ta będzie wymagała od gracza bardzo rozsądnego i strategicznego używania powyższego przedmiotu.

6.2 Mechanika tworzenia własnych punktów kontrolnych

Gracz za pomocą specjalnych przedmiotów będzie mógł wyznaczyć własny punkt kontrolny do którego będzie mógł wrócić po utracie punktów życia. Przedmioty te podobnie jak te z mechaniki spaczenia będą ograniczone ilościowo, a ich zdobycie będzie się wiązało z dodatkowym ryzykiem jakie gracz będzie musiał podjąć w celu zdobycia tych przedmiotów. To gracz będzie decydował o tym kiedy będzie chciał użyć swojego punktu kontrolnego by ułatwić sobie najtrudniejsze etapy gry.

6.3 System Inventory

Inventory pozwoli nam założyć i zdjąć przedmioty z ekwipunku. Każdy przedmiot będzie posiadał dane statystyki, które oddziałują na statystyki gracza. Przed założeniem przedmiotu lub broni można zobaczyć jego atrybuty poprzez najechanie kursorem na daną rzecz. Do inventory planowane jest dodanie system craftingu, dzięki czemu gracz będzie mógł tworzyć coraz silniejsze przedmioty z wcześniej zdobytych rzeczy. Inventory będzie posiadać funkcję, która pozwala na używanie przedmiotów jednorazowych, takich jak mikstura życia.

6.4 Atrybuty Postaci

Do naszego projektu planujemy wprowadzić system atrybutów, który wzbogaci rozwój naszej postaci. Atrybuty postaci to dane opisujące w grach aspekty, posiadane przez wszystkie postacie występujące w grze. Są one wyrażane najczęściej w postaci liczby. Niektóre gry używają innych nazw na określenie cech postaci, np. cechy lub statystyki.

Większość gier używa atrybutów by opisywać zdolności fizyczne i umysłowe bohaterów, np. ich siłę lub mądrość. Wiele systemów zawiera także cechy społeczne jak na przykład charyzmę. Często atrybuty wpływają na posiadane przez bohatera umiejętności lub niektóre akcje, które może przeprowadzić w grze, np. sukces pewnych opcji dialogowych. W konsekwencji im wyższa jest wartość danego atrybutu, tym lepiej.

Planowane jest dodanie trzech atrybutów: vitality, strength, oraz agility. Vitality jest to statystyka, która zwiększa pasek zdrowia, dzięki czemu gracz dłużej wytrzymać w walce z przeciwnikami. Strength natomiast zwiększa obrażenia zadawane naszym oponentom. Agility zwiększa pancerz, który zmniejsza otrzymywane obrażenia, oraz zwiększa pasek wytrzymałości, co pozwala na częstsze wykonywanie specjalnych ruchów. Atrybuty można zwiększać za pomocą przedmiotów.

6.5 Rozwój świata/uniwersum gry

Planowane jest stworzenie kilku lokacji z wieloma poziomami. Każda z tych lokacji ma posiadać własną historię. Obecni mają być bohaterowie niezależni (ang. non-player character, non-playable character, skracane do NPC), którzy mają oferować różne misje do wykonania, związane z głównym wątkiem fabularnym lub rozmaitymi wątkami pobocznymi. NIektóre znalezione przez gracza przedmioty mają mieć ukrytą w opisach fabułę lokacji.

Dodani mają być kolejni przeciwnicy, w tym bossowie lokacji, a także nowe rodzaje pułapek. Planowane są również elementy takie jak oblodzone ściany, ruchome platformy, liny, których można się złapać, wybijające gracza w górę obiekty, iluzoryczne ściany, wybuchające

obiekty, przedmioty, które umożliwiają graczowi powrót po śmierci do danego miejsca (checkpoint), system wskazówek dla gracza w postaci np. tabliczek, przeszkody, które mają teleportować gracza wstecz.
Wraz z dodaniem systemu ekwipunku i atrybutów postaci pojawić się mają przedmioty (np. zbroje, bronie), w które będzie można wyposażyć postać gracza i które mają podnosić statystyki.
W planach jest również dodanie systemu umiejętności bohatera, które odblokowujemy wraz z odnajdywaniem kolejnych fragmentów artefaktu. Mogą mieć one z góry określone statystyki bądź skalować się z atrybutami postaci. Związane z tym ma być projektowanie poziomów w taki sposób, aby były obecne miejsca, które gracz może odwiedzić tylko przy pomocy umiejętności odblokowanych w dalszej części gry.

6.6 Sponsoring

Jeżeli zdecydujemy się stworzyć konto na stronie takiej jak np. patreon, planujemy umożliwić naszym patronom możliwość decyzji (każda za odpowiednią cenę) w sprawie powstających elementów gry np. imienny NPC w grze, zaprojektowanie bossa, autorski fragment fabuły, klimat/rodzaj nowej lokacji itp.

7. Podsumowanie

Podsumowując gra jest kreowana w oparciu o opinie społeczności. Sugerujemy się zdaniem innych osób, aby rozgrywka sprawiała przyjemność. Poziom trudności skonstruowaliśmy w taki sposób, aby usatysfakcjonować gracza, a fabuła zachęcała do zagłębiania się w nią. Jeśli chodzi o level design, to chcieliśmy sprawić, aby świat był immersyjny i ładny pod względem wizualnym. Zarówno opinia publiczna, jak i wiele testów pomogło nam zidentyfikować błędy, które udało nam się w szybkim tempie naprawić.
Od początku naszym głównym celem była mechanika poruszania się postaci. Gra nie mogłaby obejść się bez podstawowych funkcji takich jak: licznych pułapek, przeciwników, interakcji z otoczeniem, oraz przyjemnego systemu walki. W przyszłości chcemy dodać mechanikę spaczenia, system tworzenia punktów kontrolnych, ekwipunek i atrybuty postaci.
Mechanizmy te udoskonalą i usprawnią naszą produkcję, dzięki czemu stanie się bardziej przystępna dla gracza.
Dzięki kilkumiesięcznemu nabywaniu doświadczenia w realizacji projektu, poszerzyliśmy swoją wiedzę w tworzeniu gier komputerowych, poznaliśmy lepiej rynek branży gier. Praca zespołowa była dla nas przyjemna, nauczyliśmy się współpracy, poczucia odpowiedzialności, oraz reagowania na różne problemy, jakimi są projektowe błędy.