1. **Zastosowane algorytmy**
   1. Sztuczna inteligencja

Aby zacząć pracę nad szerokopojętym zachowaniem wrogów postanowiliśmy zastanowić się najpierw jak to robili profesjonaliści z dużym doświadczeniem. Dość oczywistym wyborem było zajrzenie do kodu źródłowego Quake 2. Jest on publicznie dostępny już od 22 grudnia 2001 roku. Poza tym kod pochodzący z id Software będący dziełem John’a Carmack’a jest znany jako jeden z najpiękniejszych kodów. Prawdą jest, że Quake 2 bardzo różni się od naszej gry z wielu powodów, z czego najważniejszym jest fakt, że są to dwa zupełnie odrębne gatunki. Jednak mimo innej perspektywy i typu rozgrywki od zachowania przeciwników oczekujemy niemal tego samego, czego oczekiwalibyśmy w wypadku gry FPS.

Przeglądając kod zauważyliśmy, że przeciwnicy są sterowani różnymi flagami oznaczającymi ich stan i w zależności od stanu wykonują różne czynności takie jak szukanie, gonienie i atakowanie bohatera, czy po prostu stanie w miejscu. Na ich stan mają wpływ bodźce z otoczenia od tak oczywistych jak ujrzenie głównego bohatera, po mniej oczywiste, jak na przykład zaobserwowanie innego zachowania sąsiednich sojuszników.

Podobnie też zostało to zrealizowane przez nas. Domyślnie wrogowie są w stanie spoczynku, który ulega zmianie jeśli wróg ten zostanie zaatakowany, zobaczy głównego bohatera, lub innego wroga, który goni, lub atakuje tytułowego Mariana. W zależności od spowodowało wyjście ze stanu spoczyku, dana jednostka może rozpocząć różne czynności. W wypadku zobaczenia Mariana będzie za nim podążać, lub atakować go jeśli jest w zasięgu ataku. Jeśli zobaczy sojuszniczą jednostkę znającą położenie głównego bohatera będzie podążać w jej kierunku. Natomiast jeśli zostanie zaatakowana będzie podążać w kierunku miejsca z którego zostały zadane obrażenia, chyba że miejsce to nie będzie widoczne z jej perspektywy – wtedy przejdzie w stan zaalarmowany, w którym będzie losowo przeszukiwać otoczenie.

Jednostki podejmują próbę ataku tylko jeśli są w zasięgu charakterystycznym dla rodzaju wrogów, który reprezentują. W podobny sposób ograniczony jest ich zasięg widzenia. Jednostka będąca za daleko by atakować, lecz wystarczająco blisko, aby widzieć Mariana będzie go gonić. Jeśli ten opuści pole widzenia, jednostka będzie podążać do miejsca, w którym był on widziany po raz ostatni. Gdy dojdzie w pobliże tego punkty wciąż nie wiedząc gdzie znajduje się Marian, będzie próbować iść jego śladami. Jednak aby ucieczka przed wrogami była możliwa algorytm, według którego poruszają się wrogowie idąc jego śladami jest napisany w taki sposób, by trop prowadzący do protagonisty był szybko gubiony. Tak też się przeważnie dzieje, po czym dany wróg przechodzi w stan zaalarmowany.

Aby sprawdzić czy dany wróg widzi Mariana sprawdzane jest czy promień od danego wroga do Mariana przecina jakiś obiekt, który mógłby zasłonić widok. Takim obiektem może być ściana, natomiast inni wrogowie, lub latające w powietrzu magiczne pociski nie są w tym wypadku brane pod uwagę. Oczywiście zanim kosztowne wydajnościowo obliczenia promieni zostaną zastosowane najpierw sprawdzona zostaje odległość między wrogą jednostą, a bohaterem. Jeśli jest za duża dalsze obliczenia są zbędne. Podobna technika jest stosowane w wielu innych przypadkach takich jak sprawdzenie czy wróg widzący Mariana może poinformować o tym swoich sąsiednich sojuszników.

Poza tym zaimplementowany jest też prosty mechanizm, dzięki któremu jednostki nie skupiają się grupami w jednym punkcie, a zamiast tego gdy jest ich dużo mają tendencję do tworzenia formacji przypominającej tłum.

U przeciwników występują trzy rodzaje ataków. Jeden z nich to atak wręcz, którego działanie jest trywialne. Kiedy przeciwnik walczący wręcz jest w zasięgu ataku zadaje obrażenia co określony czas. Kolejnym, również trywalnym sposobem ataku są ataki samobójcze stosowane przez jeden z rodzajów wrogów, które na zetknięciu się z głównym bohaterem eksplodują zadając obrażenia. Ostatnim i najciekawszym pod względem algorytmicznym sposobem ataku przeciwników jest atak dystansowy. Jest on konfigurowalny kilkoma parametrami. Prócz uniwersalnych takich jak częstotliwość ataków, czy wartość obrażeń są też parametry definiujące prędkość magicznych pocisków, które wykorzystuje atak dystansowy, jak również sposób określenia kierunku w którym mają one lecieć. Przeciwnicy w przybliżony sposób przewidują gdzie należy wystrzelić pocisk by trafić w Mariana na podstawie jego prędkości i kierunku poruszania się. Wspomniane powyżej parametry definiują z jakim prawdopodobieństwem ma następować to przewidywanie i jaki wpływ na ostateczny kierunek lotu pocisków ma aktualna pozycja Mariana, oraz ta która została przewidzana. Poza tym jest jeszcze parametr określający rozrzut pocisków wycelowanych w Mariana.

Opisane powyżej algorytmy dają pozytywne wrażenia z gry, ponieważ jednostki zachowują się dość naturalnie reagując na zdarzenia w grze i informując się nawzajem co powoduje ich grupowanie się, gdy gra jest na bardziej zaawansowanym etapie i zagęszczenie wrogów jest większe. Ich ataki cechują się wysoką różnorodnością, dzięki czemu gracz, aby uniknąć obrażeń musi skorzystać z wielu różnych technik, w zależności od szybko zmieniającej się sytuacji.

* 1. Fizyka poruszania się w mini grze

Gra The Mighty Marian zaczyna się mini grą, której najważniejszym elementem jest niestandardowy sposób poruszania się inspirowany tym znanym z Quake 3 Arena. Jego założenia mogą wydawać się dziwne, jednak po nabraniu wprawy są bardzo przyjemnym elementem urozmaicającym grę. Charakteryzuje się tym, że mimo limitu prędkości, po osiągnięciu którego przestaje się przyśpieszać w kierunku, w którym ta prędkość jest skierowana, można wciąż przyśpieszać w innym kierunku. Dodatkowo ważnym jest fakt, że tarcie podczas poruszania się w powietrzu nie występuje, a skacząc tuż po zetknięciu z ziemią pomija się tarcie z ziemią. W efekcie możliwe jest ciągłe przyśpieszanie jeśli grasz skacze i rytmicznie zmienia kierunek wektora przyśpieszenia, co osiąga się zgrywając ruchy myszą z kierunkiem wybieranym klawiaturą. Technika ta jest znana jako „strafe jumping”.

Nasza implementacja znacznie różni się od tej z gry id Software, jednak wrażenia z poruszania się są dość podobne. W naszym projekcie zgranie ruchów myszy i klawiatury jest nieco łatwiejsze. Jeśli kierunek odczytany z wejścia myszy i klawiatury jest taki sam, możliwe jest uzyskanie dodatkowego przyśpieszenia w locie, jednak to bonusowe przyśpieszenie działa tylko przez pierwszy ułamek sekundy po wyskoku, aby tak ograniczyć jego działanie do racjonalnych wartości. Siła tarcia jest znacznie zwiększana jeśli gracz nie wybiera kierunku poruszania się klawiaturą, tak aby ten był w stanie szybko zahamować. Skok zostanie wykonany automatycznie po dotknięciu ziemi jeśli przycisk skoku został naciśnięty tuż przed tym momentem. Ma to na celu uniknięcie spowolnienia poprzez tarcie. Podobnie jeśli skoczymy tuż po zetknięciu z ziemią również nie stracimy prędkości, ponieważ tarcie zaczyna działać dopiero po krótkiej chwili od wylądowania. Aby odróżnić dotykanie ziemi od dotykania ścian w dolnej części postaci został dodany dodatkowy „collider”, a więc wykrywacz zderzeń, który ustawiony jest jako tak zwany „trigger”, dzięki czemu nie wpływa na obliczenia fizyki a jedynie informuje o tym, że jakiś obiekt zawiera się w jego przestrzeni.

Ogólnie rzecz biorąc skok odbywa się poprzez zmianę prędkości w pionie, nie jest to jednak tak trywialne jak można by się spodziewać, ponieważ zmiany te muszą być różne w wypadku gdy postać stoi w miejscu, porusza się pod górkę, lub odwrotnie. Za opadanie odpowiada napisana przez nas grawitacja, która po prostu zmniejsza prędkość postaci w górę, gdy ta nie dotyka podłoża.

Dość istotnym elementem jest też fakt, że w locie sterowność jest znacznie ograniczona i nie da się zmienić kierunku poruszania się tak łatwo jak podczas chodzenia po ziemi. Mimo to korzystanie z techniki zwanej „strafe jump” opisanej powyżej początkowo powodowało dość radykalne zmiany kierunku poruszania się w chwili kontaktu z ziemią, w związku z czym algorytm sterujący poruszaniem został wzbogacony o kolejne warunki ogarniczające sterowność w sytuacji, gdy lądujemy na ziemi tylko po to, by szybko znów podskoczyć. W efekcie możliwe jest rozwijanie coraz większych prędkości w stabilnym kierunku, natomiast jeśli nie to jest naszym celem również nic nie przeszkadza nam w doborze dowolnego toru ruchu.

Niektóre z opisanych powyżej funkcjonalności dałoby się osiągnąć za pomocą gotowych mechanizmów fizycznych takich jak grawitacja, czy tarcie. Jednak takie rozwiązanie ogranicza wolność w modyfikowaniu. W grze ponad realizm są istotne wrażenia z gry, zatem nie wszystko można zapisać używając praw fizyki imitujących rzeczywistość.

Kolejną techniką nietypowego poruszania się we wspomnianej grze, która była inspiracją do stworzenia opisywanej w tym punkcie mini gry jest tak zwany „rocket jumping”. Istotą tej techniki jest wykorzystanie siły odrzutu eksplozji rakiety do zyskania dodatkowego przyśpieszenia. Zwykle w tym celu strzela się rakietą pod własne nogi, aby dzięki temu móc skoczyć wyżej i dalej.

W naszej grze główną różnicą jest fakt, że zamiast rakiety jest wybuchowa kula ognia, która jednak w przeciwieństwie do rakiety nie zadaje obrażeń głównemu bohaterowi. Efekt działania jest bardzo podobny, czyli magiczna kula ognia po wystrzeleniu porusza się ze stałą prędkością w linii prostej, a na kontakcie z obiektami eksploduje powodując odrzut. Aby jednak uniknąć problemów z niezbyt dokładnym obliczaniem kolizji szybko poruszających się obiektów w momencie wystrzelenia ognistego pocisku jest symulowany promień. Jeśli przetnie on jakiś obiekt zostanie zapisana pozycja tego przecięcia, oraz czas w którym magiczna kula znajdzie się w tym miejscu. Czas ten jest obliczany na podstawie prędkości, aktualnego czasu, oraz odległości między punktem przecięcia, a źródłem promienia, czyli miejscem z którego kula ognia zaczyna swoją drogę. Następnie w momencie wyznaczonym obliczeniami opisanymi powyżej Marianowi zostanie nadana prędkość w wyniku odrzutu, jeśli był on wystarczająco blisko eksplozji. Kierunek tego impulsowego przyśpieszenia też oczywiście jest konkretnie obliczany na podstawie pozycji bohatera, oraz eksplozji.

Przedstawiony model fizyki nadaje atrakcyjność mini grze i z pewnością jest przyjemnym elementem dla bardziej wprawionych graczy, natomiast w wypadku mniej doświadczonych osób może sprawiać wrażenie trudnego do nauczenia, w związku z czym mini gra jest traktowana tylko jako startowy bonus i dobre wyniki w niej nie są konieczne do czerpania radości z całej gry.

* 1. **Dźwięk**
     1. Muzyka
     2. Efekty dźwiękowe
  2. **Ataki i umiejętności**
     1. Mechanika
     2. Efekty wizualne
  3. **Grafika**
     1. Shadery i materiały
     2. Modele i tekstury
  4. **Mechanizmy urozmaicające grę**
     1. Mechanizm nagłej śmierci

1. **Testowanie**
2. **Podsumowanie**