

Zakończenie gry

Pod koniec poprzedniej części podsumowaliśmy większość głównych funkcji, które planowaliśmy włączyć do gry. O ile moglibyśmy kontynuować pracę nad ostatnimi częściami nad każdym najmniejszym szczegółem gry, to stałoby się to dość powtarzalne i prawdopodobnie stracilibyśmy kilku czytelników. W końcu gra ma być jedynie zabawną demonstracją koncepcji budowania opartej na lokalizacji gry AR. Miejmy nadzieję, że zastanawiałeś się nad swoim własnym, unikalnym projektem gry i nad tym, jak można by go rozwinąć podczas pracy nad tą książką. Tak więc, zamiast budować na grze Foody GO, w tej części omówione zostaną następujące rzeczy:

- * Wybitne zadania rozwojowe
- * Brakujące umiejętności rozwojowe
- * Sprzątanie aktywów
- * Wydanie gry
- * Problemy z budowaniem gry opartej na lokalizacji
- * Gry wieloosobowe oparte na lokalizacji
- * Firebase jako platforma dla wielu graczy
- * Inne pomysły na gry oparte na lokalizacji
- * Przyszłość gatunku

Wybitne zadania rozwojowe

Po zakończeniu prac nad wersją demonstracyjną Foody Go pomocne będzie przejrzanie zaległych zadań, które deweloper może podjąć, aby ukończyć grę do komercyjnego wydania. To ćwiczenie z pewnością przyda się nowym programistom planującym w przyszłości wydanie pełnej gry komercyjnej. Być może słyszeliście o regule 80/20, która mówi, że wykonanie zadania będzie wymagało od 20% wysiłku do 80%, a kolejne 80% na wykonanie ostatnich 20%. Ta zasada odnosi się dobrze do większości zadań, ale szczególnie dobrze do tworzenia oprogramowania i gier. Jeśli weźmiesz pod uwagę, że gra demo jest ukończona w 80%, ukończenie całej gry będzie wymagało kolejnych czterech razy więcej wysiłku (6 rozdziałów rozwoju * 4 = 24 rozdziały). Może się to wydawać ogromną ilością pracy, biorąc pod uwagę wszystko, co zostało ukończone, ale podsumujmy według sceny, które elementy są wyjątkowe:

- * Scena mapy:
 - Efekty dźwiękowe lub muzyka
 - Skybox z chmurami i słońcem lub nocą w zależności od pory dnia
 - Ulepszone spawnowanie potworów, serwer centralny
 - Efekty cieniowania wizualnego
 - Style mapy
- * Złap scenę:
 - Efekty dźwiękowe i muzyka

- Efekty cieniowania wizualnego
- Kamera żyroskopowa
- Możliwość włączenia/wyłączenia AR (kamera działająca w tle)
- Przełączanie kulek zamrażających z ekwipunku
- Uciekaj z miejsca zdarzenia!
- Zrób zdjęcie aparatem urządzenia
- * Scena z inwentarzem (dom):
- Ekwipunek przedmiotów
- Szczegóły potwora
- Szczegóły postaci, statystyki i poziom
- Indeks potworów, jeśli rozwinie się więcej potworów lub innych stworzeń
- * Miejsca sceny:
- Efekty dźwiękowe i muzyka
- Efekty cieniowania wizualnego
- Kamera żyroskopowa
- Możliwość włączenia/wyłączenia AR lub tła Street View
- Miejsce śledzi szefa kuchni w danej lokalizacji
- Animacje potworów
- Ulepszony interfejs użytkownika
- * Scena powitalna:
- Dźwięk lub muzyka
- Ładowanie efektów
- Obrazy
- * Scena gry:
- Możliwość wyboru/dostosowania postaci (mężczyzna/kobieta)
- Różnorodne potwory, rozwój postaci
- Usługa dźwiękowa lub menedżer
- Opcja Massively Multiplayer Online (MMO)
- Oczyszczanie zasobów
- Poprawki błędów (wszystkie sceny)

Oprócz przedmiotu dla wielu graczy, o którym powiemy później, na liście znajdują się liczne przedmioty, które nie są zbyt skomplikowane, ale nadal wymagają kilku dodatkowych godzin wysiłku, aby je ukończyć. Efekty wizualne (shadery), style i poprawki błędów są szczególnie problematyczne i mogą dosłownie stać się czarną dziurą w czasie tworzenia. Wiadomo, że deweloperzy spędzają kilka miesięcy lub dłużej pracując nad konkretnymi shaderami, aby uzyskać konkretny wygląd odpowiedni dla ich gry. Mamy nadzieję, że teraz lepiej docenisz, jak zasada 80/20 z pewnością będzie miała zastosowanie do naszej gry demonstracyjnej. Oczywiście możesz wziąć grę demo, którą mamy tutaj i ukończyć ją jako własną, z własnymi funkcjami i mechaniką gry, jako grę AR opartą na lokalizacji. Jednak ważne jest, aby mieć na uwadze wysiłek, jaki wkładasz w rozwój lub ukończenie gry, zwłaszcza jeśli jest to Twoja pierwsza gra. Upewnij się, że ustaliłeś sobie datę premiery, a następnie postaraj się opublikować grę jako wersję alfa/beta/wydanie w tym dniu. Dzięki temu Twoja gra będzie docierać do graczy w odpowiednim czasie i otrzymasz natychmiastową informację zwrotną w jakiejś formie. Podążanie za tą strategią nie tylko sprawia, że jesteś bardziej świadomy terminów, ale także wzmacnia twoje oszacowanie wysiłku w odniesieniu do nagrody (informacje zwrotne). Jako przykład tego, jak łatwo jest spędzać czas na pracy z efektami wizualnymi lub odkrywaniu ich, wykonaj następujące ćwiczenie:

1. Otwórz dowolną scenę w Unity, z gry demo, którą właśnie zbudowaliśmy, lub inną wybraną przez siebie scenę.
 2. Z menu wybierz Zasoby | Importuj pakiet | Efekty. Spowoduje to zaimportowanie pakietu Unity Standard Assets Effects. Kontynuuj importowanie tak jak wcześniej.
 3. Zlokalizuj główną kamerę w swojej scenie. Jeśli nie masz pewności, gdzie to jest, użyj funkcji wyszukiwania w górnej części okna Hierarchia. Po prostu zacznij wpisywać kamerę główną lub kamerę w obszarze wyszukiwania, a kamera sceny zostanie pokazana w hierarchii.
- Wskazówka** : filtry i inne elementy wizualne łatwo dają się ponieść emocjom. Pamiętaj tylko, że te filtry mogą mieć bezpośredni i znaczący wpływ na wydajność, pamięć i rozgrywkę. Używaj ich w ukończonej grze z rozwagą.
4. Wybierz kamerę, a następnie z menu wybierz Komponent | Kwitnienie i blask | Kwiat. Spowoduje to dodanie filtra Bloom do aparatu, a okno gry przyjmie jaśniejszy odcień.
 5. Możesz kontynuować dodawanie różnych efektów (filtrów) do aparatu, co jakiś czas uruchamiając scenę, aby zobaczyć, jaki efekt ma filtr. Jest prawdopodobne, że po kilku krótkich godzinach zdasz sobie sprawę, że istnieje wiele możliwości wyboru i opcji.
 6. Spróbuj nie tylko dodać różne efekty, ale także spróbuj zmienić kolejność komponentów efektów w samym aparacie.

Listy zaległych zadań wskazywały na niektóre zestawy umiejętności, które ledwo omówiliśmy lub w ogóle nie. Chociaż było to celowe, ponieważ te umiejętności nie były krytyczne dla elementów funkcjonalnych gier, które tworzyliśmy, nadal są kluczowe dla każdej gry. Przyjrzymy się tym umiejętnościom, które, chociaż przegapiliśmy, będą kluczowe dla twoich przyszłych wysiłków rozwojowych, w następnej sekcji.

Brakujące umiejętności rozwojowe

Mimo że przegapiliśmy lub ledwo dotknęliśmy niektórych z tych obszarów rozwoju naszej gry demonstracyjnej, rzeczywiście będą one kluczowe dla ukończenia oszałamiającej reklamy lub po prostu

fajnej gry z tego gatunku. Poniżej znajduje się lista umiejętności rozwojowych według priorytetów, wraz z podlistą zasobów, na które powinieneś zwrócić uwagę:

* Shadery (efekty wizualne i oświetlenie): To obszerny temat i prawdopodobnie nigdy nie będziesz musiał pisać shadera, ale z pewnością będziesz musiał nauczyć się, jak skutecznie z nich korzystać. Shadery służą do rozwoju gry, tak jak powietrze do oddychania; niezbędne i wszędzie. Nauka tworzenia shaderów to zaawansowana umiejętność, ale nawet niewielka wiedza może zająć daleko. Nawet jeśli nie planujesz pisać własnych shaderów, warto przyrzeć się następującym zasobom:

- WikiBooks: (https://en.wikibooks.org/wiki/Cg_Programming/Unity), ma dobre wprowadzenie do programowania shaderów i dobry zestaw zasobów. Strona może być przytłaczająca dla początkujących lub tych, którzy lubią podążać ścieżką nauki, na przykład w dobrej książce.

* Efekty cząsteczkowe: Ponownie, jest to umiejętność, bez której prawdopodobnie możesz się obejść, ale nadejdzie czas, kiedy będziesz chciał uzyskać dodatkowy efekt cząsteczkowy, który będzie wymagał wiedzy na temat budowania lub modyfikowania niestandardowego efektu cząsteczkowego. Nawet posiadanie wiedzy o tym, jak różne ustawienia w systemie cząstek Unity mogą być pomocne podczas dostosowywania systemu cząstek do swoich potrzeb. Oto lista zasobów, które mogą okazać się pomocne w poszerzeniu wiedzy o systemie cząstek Unity:

* Unity Learning (<https://unity3d.com/learn/tutorials/topics/graphics/particle-system>), to świetne wprowadzenie do korzystania z systemu efektów cząsteczkowych

* Multiplayer (sieć): Oczywiście, jeśli chcesz przekonwertować grę demonstracyjną na grę Massively Multiplayer Online (MMO), ta umiejętność będzie niezbędna. Niestety, wymagane konkretne umiejętności i wiedza w tej dziedzinie często będą zależeć od infrastruktury sieciowej, od której zależy twoja gra. Dowiemy się, dlaczego tak jest, i porozmawiamy o różnych rozwiązaniach sieciowych i zasobach w dalszej części.

* Animacja (Mecanim): Wielu programistów automatycznie zakłada animację, a Mecanim służy tylko do animowania postaci lub humanoidów. Chociaż to z pewnością prawda, dzięki animacjom Mecanim możesz osiągnąć kilka rzeczy, w tym oczywiście animację postaci. Oto kilka świetnych zasobów dotyczących animacji w Unity:

Unity Learning (https://unity3d.com/learn/tutorials/topics/animation/animate_anything_mecanim) to dobry szybki samouczek dotyczący korzystania z Mecanim.

* Dźwięk: To jedna z tych często pomijanych umiejętności, które są tak niezbędne w dobrych grach. Oczywiście innym problemem jest to, że jest bardzo mało informacji na temat najlepszych praktyk dodawania dźwięku i muzyki do gry Unity.

* Teksturowanie: to sztuka nakładania obrazów 2D na model 3D. Jest to umiejętność, która wykracza poza programowanie, ale jeśli jesteś małym niezależnym deweloperem pracującym nad własną grą, pewna wiedza w tej dziedzinie będzie korzystna. Ta umiejętność dobrze wpisuje się w modelowanie 3D lub stanowi jego część. Jest to również rozszerzenie shaderów i programowania shaderów.

* Modelowanie 3D (rozwój postaci): Jest to kolejna rzecz, bez której twórcy gier zręcznościowych mogą się obejść i nadal dobrze sobie radzą. Z pewnością jest wystarczająco dużo treści 3D. Jednak nauka samego procesu modelowania obiektu 3D lub postaci może być kolejnym świetnym dodatkiem do zestawu umiejętności. Nie wspominając o tym, że jeśli jesteś niezależnym programistą, w pewnym momencie otworzysz program do modelowania 3D, aby zbudować niestandardowy model. Oczywiście niektóre umiejętności, których nauczysz się w tym obszarze, będą specyficzne dla używanego

oprogramowania. Oto kilka pakietów specyficznych dla oprogramowania z samouczkami, które pomogą Ci zacząć:

- iClone Character Creator (Reallusion), <https://www.reallusion.com/iclone/game/> , to doskonały pakiet oprogramowania z dużym wyborem filmów instruktażowych, które pozwolą Ci łatwo tworzyć niestandardowe postacie. Jeśli zapomniałeś, postacie użyte w tej grze to postacie iClone dostarczone bezpłatnie od Reallusion.

- Blender, <https://www.blender.org/>, to standardowe bezpłatne narzędzie do modelowania treści 3D. Zapoznanie się z tym narzędziem wymaga inwestycji w czasie.

Oczywiście, jeśli nie masz czasu na doskonalenie umiejętności, zawsze możesz poszukać zasobu w Unity Asset Store, aby uzyskać pełną funkcjonalność. W następnej sekcji omówimy kilka wskazówek dotyczących wyboru i używania zasobów.

Sprzątanie majątku

Kiedy po raz pierwszy zaczynasz rozwijać się w Unity, Asset Store może być zarówno błogosławieństwem, jak i przekleństwem. W sklepie dostępnych jest kilka doskonałych zasobów, które mogą pomóc w tworzeniu gry. Jednak często istnieje cena związana z użyciem zbyt wielu zasobów i zasobów, które mają dodatkowy bagaż, takich jak sceny demonstracyjne lub inne pakiety. Pakiety zasobów omówiliśmy wcześniej, ale oto kilka wskazówek, które pomogą Ci zdecydować, jak i kiedy kupować i/lub pobierać zasoby:

* Zadaj sobie pytanie, dlaczego potrzebujesz zasobu:

- Napisanie własnego zajęłoby zbyt dużo czasu: jeśli uważasz, że większość aktywów jest wyceniana na mniej niż 100 USD, może to być ważny powód do zakupu, zwłaszcza jeśli zasób wykonuje wiele rzeczy, których potrzebujesz i cenisz swój czas. Upewnij się tylko, że potrzebujesz zasobu, nawet jeśli jest bezpłatny, postępując zgodnie z innymi wytycznymi z tej listy.

- Dostarcza treści wykraczające poza twoją wiedzę: Prawdopodobnie najlepszym powodem, aby wybrać zasób, jest nowość w Unity i/lub tworzeniu gier. Upewnij się, że postępujesz zgodnie z innymi wskazówkami z tej listy, aby potwierdzić, że chwycenie go ma sens.

- Jest w sprzedaży: unikaj tej pułapki; to powinien być drugorzędny powód, który być może podejmuje decyzję za Ciebie.

- Ma świetne recenzje: recenzje są świetne, ale rozumiem główny powód, dla którego potrzebujesz tego zasobu. Nawet jeśli zasób jest narzędziem do edycji, pamiętaj, że nadal zajmuje miejsce w twoim projekcie.

- Używają go inne fajne gry: To kolejna pułapka; chyba że próbujesz dokładnie dopasować funkcjonalność i infrastrukturę, upewnij się, że inne powody mają sens.

* Określ, jakie wsparcie istnieje dla zasobu:

- Wersja zasobów jest zgodna z wersją Unity. Jest to konieczne w przypadku nowicjuszy, ponieważ radzenie sobie z konfliktami wersji może być koszmarem. Dla bardziej doświadczonych programistów może to nie stanowić problemu.

- Deweloper obsługuje aktualizacje zasobu. Sprawdź stronę zasobu i upewnij się, że deweloper obsługuje zasób. W niektórych przypadkach może to stanowić mniejszy problem, na przykład w przypadku aktywów artystycznych. W przypadku zasobów z kodem źródłowym staje się to mniejszym

problemem. Jednak wysokowydajne lub ściśle zintegrowane zasoby, takie jak woda, teren i animacja, to coś, co programista powinien często aktualizować.

- Recenzje są pochlebne i aktualne. Upewnij się, że zasób ma recenzje i są one aktualne i oczywiście pozytywne. Ponownie, jest to mniejszy problem w przypadku bezpłatnych zasobów, ponieważ programiści zazwyczaj nie komentują bezpłatnych zasobów. W przypadku innych zasobów może to mieć krytyczne znaczenie. Jeśli jest jakaś wskazówka problemów z zasobem, często lepiej jest tego unikać.

- Zasób ma wątek na forum Unity. Często jest to znakomita wskazówka, że programista reaguje na komentarze i poprawki błędów. Przeczytaj posty i sprawdź daty oraz rodzaj pytań. Często daje to świetny wgląd w to, jak działa zasób i czy naprawdę będzie działać w twoim projekcie.

- Zasób zapewnia bezpłatną dokumentację. Jest to często wiadome teraz, gdy większość aktywów zapewnia bezpośredni link do ich dokumentacji, niezależnie od tego, czy kupujesz, czy nie. Przeczytaj pokrótce i ustal, czy możesz postępować zgodnie z dokumentacją i czy jest to coś, co będzie pasować do Twojego projektu.

* Przejrzyj zawartość pakietu zasobów:

- Zasób zawiera foldery wtyczek. Zazwyczaj oznacza to, że zasób będzie zawierał skompilowane biblioteki, które mogą wymagać specjalnej konfiguracji podczas wdrażania gry. Oczywiście może to być problem i coś, o czym należy pamiętać. Niektórzy programiści mogą wyświetlać podgląd tylko zasobów, w których mają dostęp do całego kodu źródłowego. Jednak czasami może to być kompromis, ponieważ zasób może działać znacznie lepiej jako skompilowana wtyczka.

- Zasób zawiera dodatkową zawartość. Zasób, który zawiera dodatkowe pliki demonstracyjne, może być świetnym zasobem edukacyjnym. Niektóre zasoby mogą dodawać pakiety Unity Standard Asset lub inne treści, które mogą kolidować z istniejącymi zasobami. Może to nie być powód, aby unikać używania zasobu, ale warto wiedzieć, zanim wciągniesz zasób do swojego projektu. Często korzystniejsze jest również wprowadzenie zasobu z dużą ilością wersji demonstracyjnych do projektu testowego, w którym można rozłożyć zasób tylko do podstawowych elementów, a następnie wyeksportować go do wykorzystania w projekcie.

- Struktura zawartości aktywów jest dobrze zorganizowana. Staraj się unikać zasobów, które ładują zawartość do wielu folderów głównych. Może to być nie tylko przeszkodą w zarządzaniu aktywami, ale często jest oznaką, że rzeczy, których nie trzeba ładować, mogą być ładowane". Zawartość zasobu jest przeznaczona dla Twojej platformy wdrożeniowej. Ten jest duży; nie oczekuj, że na urządzeniu mobilnym będziesz używać treści zaprojektowanych dla komputerów stacjonarnych. Jeśli zasób nie zawiera treści zaprojektowanej dla Twojej platformy, szukaj dalej.

- Skrypty zasobów są w preferowanym przez Ciebie języku. Nie zawsze jest to dane i nie chcesz, aby była to niespodzianka. Jeśli wolisz na przykład C#, upewnij się, że wszystkie pliki skryptów zawartości kończą się rozszerzeniem .cs.

* Porównaj alternatywy:

- Zasób ma dużą konkurencję. Ogólnie rzecz biorąc, jest to dobra rzecz dla dewelopera, ponieważ często oznacza, że składnik aktywów będzie dobrze wyceniony w stosunku do pełnionej funkcji. Oznacza to również najprawdopodobniej, że samodzielne budowanie aktywów nie byłoby warte czasu. Jednak konkurencyjny rynek może sprawić, że określenie, który zasób jest odpowiedni dla twojego projektu, może być przytłaczające, ponieważ mogą istnieć nakładające się cechy i funkcje.

- Zasób nie ma konkurencji. W sklepie jest kilka rzeczy, które są tak dobrze wykonane; wymagają tak zaawansowanej wiedzy o Unity i tworzeniu gier, że nikt nie chce z nimi konkurować. Alternatywnie, aktywnym może być nowa koncepcja rozwoju, która zostanie wydana jako pierwsza. Dopóki przestrzegasz pozostałych wskazówek, możesz określić, czy zasób jest dobrym wyborem.

* Inne uwagi:

- Zasób jest zgodny z platformami docelowymi. Upewnij się, że zasób będzie działał z Twoimi platformami wdrożeniowymi. W wielu przypadkach może to nie być możliwe.

- Zasób ma bezpłatną wersję. Za każdym razem, gdy zobaczysz bezpłatną wersję zasobu, z pewnością ułatwi to podjęcie decyzji. Oczywiście nadal musisz pobrać zasób i wykonać całą konfigurację i konfigurację dla swojego projektu. Następnie, jeśli okaże się, że zasób nie działa, przejdź przez problem z późniejszym usunięciem zawartości. Uważaj na darmowe wersje, w końcu mogą być po prostu stratą czasu.

- Atutem jest pakiet startowy gry lub duży framework. Pakiety startowe i frameworki mogą być świetne, ale musisz być szczególnie ostrożny, gdy idziesz tą drogą. Oczywiście ta książka sama w sobie rozwija grę AR opartą na lokalizacji. Jednak czytając ten tekst, dobrze znasz wszystkie niuanse projektu. Może tak nie być, jeśli pobrałeś całkowicie zagraniczny zestaw startowy. Jeśli planujesz zbudować własną grę na początek, odrób swoją pracę domową i upewnij się, że rozumiesz niuanse zestawu.

Teraz, gdy przejrzeliśmy obszerny zestaw wytycznych dotyczących gromadzenia zasobów dla Twojego projektu, chcemy również omówić sposób czyszczenia i usuwania dodatkowej zawartości zasobów, której projekt nie jest potrzebny. Poniżej znajduje się lista opcji zarządzania zasobami wraz z praktyczną wskazówką, jak łatwo przeprowadzić podstawowe porządkowanie projektu:

* Blokuj niepotrzebne treści podczas importu. Jeśli dobrze odrobiłeś pracę domową, powinieneś mieć dobry pomysł na wszystkie przedmioty, które zasób będzie chciał zaimportować. Dobrą praktyką jest również zawsze importowanie projektu do projektu pustego lub testowego. To powinno dać ci wyobrażenie o tym, jakiej zawartości potrzebujesz i czego możesz użyć do demonstracji. Następnie, po dokonaniu faktycznego importu do działającego projektu, możesz odznaczyć zawartość, której nie potrzebujesz.

* A+ Assets Explorer, płatna wersja tego narzędzia jest rozsądna i zapewnia dobry zestaw narzędzi dla projektów z dużą ilością zasobów, które mogą obejmować większość projektów. To narzędzie nie wyczyści zasobów, po prostu powie Ci, gdzie są Twoje problemy.

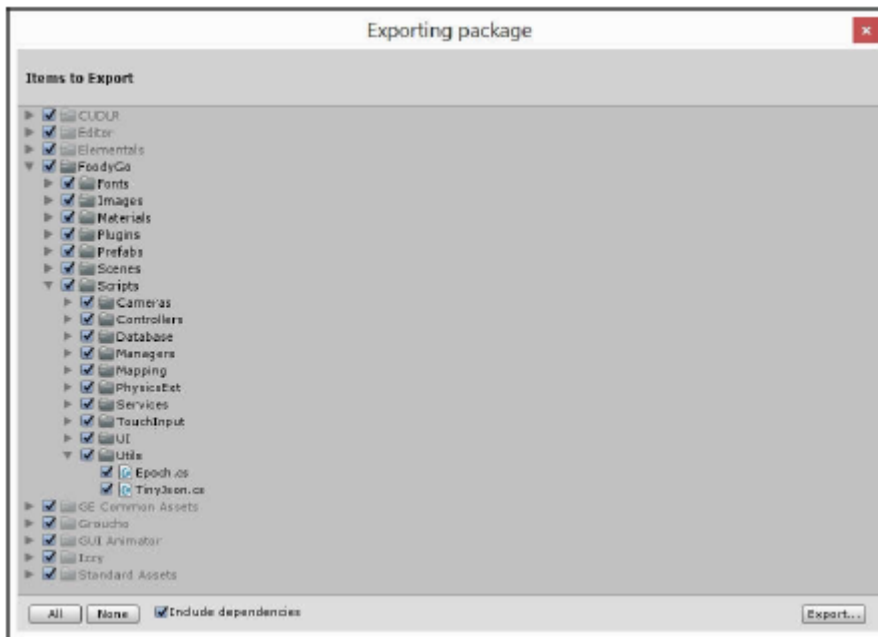
* Nie przenoś zawartości importowanych zasobów. Nigdy nie przenoś zawartości zasobu, która może zostać później zaktualizowana przez programistę zasobu. Przenoszenie treści spowoduje tylko późniejsze problemy; podczas aktualizacji pliki mogą zostać zduplikowane lub upuszczone w niewłaściwych folderach. Z drugiej strony, jeśli wiesz, że zawartość nie zostanie uaktualniona, przenoszenie jej nie stanowiłoby problemu. Oczywiście ta zasada nie dotyczy również treści, które możesz zaimportować ręcznie.

* Wykonaj pełny eksport zasobów projektu. Gdy projekt osiągnie etap, w którym uważasz, że nadszedł czas na czyszczenie lub po prostu chcesz usunąć niepotrzebne zasoby, wykonaj pełne czyszczenie eksportu zasobów w następujący sposób:

1. Otwórz swoją grę w edytorze Unity i upewnij się, że bieżąca scena i projekt zostały zapisane.
2. Upewnij się, że cała niestandardowa zawartość gry (skrypty, prefabrykaty, sceny, materiały i obrazy) znajduje się w głównym folderze projektu.

Wskazówka: jeśli użyjemy projektu FoodyGo jako przykładu, utworzysz nowy folder Scenes w folderze FoodyGo i przeciągnij wszystkie sceny z folderu Asset do nowego folderu Assets/FoodyGo/Scenes. Następnie cała zawartość niestandardowa będzie znajdować się w folderze Assets/FoodyGo.

3. Wybierz folder zawartości niestandardowej najwyższego poziomu i wybierz Zasoby | Eksportuj pakiet... z menu, które otworzy okno dialogowe Eksportowanie pakietu. Upewnij się, że pole wyboru Uwzględnij zależności jest zaznaczone na dole formularza i jak pokazano w przykładowym oknie dialogowym



4. Kliknij przycisk Eksportuj... w oknie dialogowym, aby otworzyć okno dialogowe Eksportuj pakiet.... Wybierz dobrą lokalizację, aby zapisać pakiet, nazwij go czymś niezapomnianym i kliknij Zapisz.

5. Otwórz nową instancję Unity z czystym pustym projektem. Teraz z menu wybierz Zasoby | Importuj pakiet | Pakiet niestandardowy... i użyj otwartego okna dialogowego, aby wybrać właśnie zapisany plik. Następnie kliknij Otwórz, aby rozpocząć import pakietu.

6. Po otwarciu okna dialogowego Importuj pakiet potwierdź, że znajduje się tam cała zawartość niestandardowa i część zależna. Jeśli wcześniej nie wyczyściłeś swojego projektu, zawartość zostanie wymiennie zmniejszona. Kliknij przycisk Importuj, aby zaimportować pakiet.

Sprawdź, czy pełna gra działa zgodnie z oczekiwaniami. Na tym etapie możesz spróbować ponownie wykonać krok 5, ale tym razem pomijając elementy, które na pewno nie będą potrzebne. Następnie przetestuj ponownie i spróbuj ponownie od kroku 5.

Uwaga : istnieją oczywiście inne narzędzia i metody ograniczania rozrostu zasobów. Ta metoda pozwala jednak testować zależności w izolacji i jednocześnie tworzy kopię zapasową Twojej gry.

Efektywne zarządzanie zasobami w Twoim projekcie jest koniecznością w każdej grze, która wyszła poza koncepcję. Czyszczenie i usuwanie niewykorzystanych zasobów to szczególnie ćwiczenie, które będziesz chciał wykonać przed wydaniem gry i zostało omówione w następnej sekcji.

Wydanie gry

Nawet jeśli tworzysz grę jako hobby, w celach rekreacyjnych lub edukacyjnych, powinieneś w pewnym momencie udostępnić ją innym. Otrzymywanie opinii na temat Twojej gry może być ekscytującym, upokarzającym, a nawet rozczarowującym doświadczeniem. Nie zniechęcaj się krytyką i zwracaj uwagę na dobre lub złe komentarze testera. Wystanie komuś zrzutu ekranu lub klipu filmowego z gry do sprawdzenia nie wystarczy. Poniżej znajduje się kolejny zestaw wskazówek, które zwiększą sukces Twojej premiery gry, nawet jeśli jest ona przeznaczona tylko dla znajomych:

- * Wdróż i przetestuj grę na swoich platformach docelowych. Może to być trudne dla platformy takiej jak Android, gdzie istnieje tak wiele odmian urządzenia. Jednak samo przetestowanie na jednym urządzeniu z Androidem może ujawnić wiele problemów, które napotkają Twoi gracze.

- * Testuj z małą grupą docelową, ważne jest, aby Twoi testerzy dopasowywali się do odbiorców, dla których wydajesz swoją grę. Jeśli nie masz takich testerów, to musisz wyjść i poszukać. Na szczęście istnieje wiele witryn i forów poświęconych tworzeniu gier, na których można znaleźć bardzo uczciwych kandydatów do testów.

- * Napraw błędy, zazwyczaj jest to moment, w którym wstępne opinie testerów zachęcają Cię do dalszego ukończenia gry lub całkowitego zapomnienia o projekcie. Pamiętaj tylko, że krytyka jest dobra i nie bój się rozważać sugestii. Oceń projekt uczciwie i zastanów się, czy musisz wrócić do rozwoju, aby naprawić jakiegokolwiek poważne lub krytyczne błędy.

Uwaga : Błędy krytyczne są zazwyczaj definiowane jako nieoczekiwane awarie gry lub utrata danych. Poważne błędy to coś, co przeszkadza w rozgrywce lub w zły sposób utrudnia graczowi.

- * Zwolnij do większej grupy docelowej. Jeśli to możliwe, zrób kolejną wersję dla większej grupy docelowej. Może to nie zawsze być praktyczne, aw niektórych przypadkach może być po prostu lepiej zrobić faktyczne wydanie do sklepu.

- * Prześlij do App Store. W zależności od platformy docelowej może to być duża przeszkoda, ale nie musi. Nawet jeśli wydasz grę w cenie podstawowej lub za darmo, będzie to wspaniałe doświadczenie.

- * Zautomatyzuj proces. Jeśli jesteś zadowolony ze swojego doświadczenia we wdrażaniu do sklepu z aplikacjami, w pewnym momencie będziesz chciał zaktualizować grę o nowe funkcje i poprawki błędów. Aby zminimalizować błędy i opóźnienia w wydawaniu aktualizacji, będziesz chciał zautomatyzować proces wydawania. Automatyzacja wydania nie tylko sprawi, że będziesz lepiej reagować na zmiany, ale także zwolni czas na poprawki błędów i nowe funkcje.

Teraz, gdy omówiliśmy umiejętności, zasoby i wytyczne dotyczące wydania, nadszedł czas, aby poznać brudne sekrety gier opartych na lokalizacji. W następnej sekcji porozmawiamy o trudnościach budowania gry opartej na lokalizacji.

Problemy z grami lokalizacyjnymi

Gry oparte na lokalizacji są stosunkowo nowym gatunkiem gier, spowodowanym powszechnym wykorzystaniem GPS w urządzeniach mobilnych. Pod wieloma względami ten gatunek wciąż jest nowatorski, a programiści wciąż znajdują sposób na to, co działa z graczami. Przyjrzyjmy się głównym problemom związanym z grami opartymi na lokalizacji i jakie opcje możesz mieć lub nie, aby je obejść:

- * Usługi mapowania GIS: Prawdopodobnie podstawą Twojej gry będzie wymóg wyświetlania lokalizacji na mapie. Poniżej znajduje się lista opcji, jak możesz to zrobić:

- Używaj własnych serwerów GIS: Chociaż jest to opcja i istnieje kilka dobrych platform GIS typu open source, pamiętaj, że nadal będziesz musiał obsługiwać serwery, a platformy GIS notorycznie zużywają procesor.

- Korzystaj ze statycznych map Google: jest to opcja, której z powodzeniem użyliśmy w tej książce. Należy jednak pamiętać, że korzystanie ze statycznego interfejsu API Map Google podlega ograniczeniom. Ograniczenie to 2500 żądań dziennie na adres IP.

- Korzystaj z Map Google: Mapy Google są obecnie bezpłatne na platformach Android lub iOS, pod warunkiem korzystania z dostarczonego pakietu SDK. Być może w przyszłości może to być również opcja dla programistów Unity po wyjęciu z pudełka.

- Użyj innej usługi: istnieje wiele innych bezpłatnych usług GIS. Może to być prawidłowa opcja, jeśli nie masz nic przeciwko stylom map. Alternatywnie, shader może być użyty do zmiany stylu wizualnego w sposób bardziej estetyczny dla twojej gry.

* Dane lokalizacyjne: Gry oparte na lokalizacji są z natury powiązane z danymi lokalizacyjnymi, czy to w świecie rzeczywistym, czy wirtualnym. Poniższa lista zawiera kilka opcji dostępu do danych o lokalizacji:

- Korzystaj z własnych serwerów GIS: Znowu jest to droga opcja, ale z pewnością ma swoje zalety. Możesz na przykład odradzać potwory lub stworzenia zgodnie z zaawansowanymi zasadami GIS. Wykonanie tego kroku będzie wymagało pewnej zaawansowanej wiedzy na temat GIS.

– Użyj Miejsc Google: opcja, której użyliśmy w tej książce, sprawdziła się w naszym małym przykładzie, ale niestety może nie być dobrze skalowana. Pamiętaj, że te limity użytkowania były dość restrykcyjne. Możesz jednak uzyskać licencję od Google, aby zwiększyć limity za odpowiednią cenę. Może to być więc opłacalna opcja, jeśli Twoja gra przynosi dochody.

- Użyj innej usługi: istnieją inne usługi danych oparte na lokalizacji (na przykład Foursquare), które zapewniają podobne lub, w niektórych przypadkach, lepsze dane w zależności od potrzeb i regionu. Może to być realna opcja i możesz wykorzystać niektóre umiejętności, które nauczyłeś się w tej książce, aby połączyć się z innymi usługami.

* Obsługa trybu wieloosobowego: w przeciwieństwie do innych gatunków, gra oparta na lokalizacji nie może po prostu oczekiwać korzystania z gotowych usług dla wielu graczy. Gry oparte na lokalizacji są ciągle, a gracze często grają przez dłuższy czas. Ponadto gracze powinni wchodzić ze sobą w interakcje tylko w określonej odległości fizycznej. Więcej szczegółów na temat sieci dla wielu graczy omówimy w następnej sekcji, ale oto lista opcji dla wielu graczy:

- Photon PUN: Photon to świetna usługa dla wielu graczy, łatwa do skonfigurowania i szybkiego uruchamiania. Jednak, podobnie jak inne usługi sieciowe dla wielu graczy, obsługa rozszerzonej zmiany stanu jest ograniczona. Oznacza to, że gracze, którzy połączą się po wyjeździe, mogą zostać przytłoczeni wiadomościami o aktualizacjach.

- Jedność UNET. System sieciowy Unity dla wielu graczy, UNET, to solidna i użyteczna platforma do gier peer-to-peer. W przypadku gier, które wymagają rozszerzonych filtrów stanowych i regionalnych, UNET z pewnością nie jest tym.

- Inne platformy dla wielu graczy: istnieje wiele innych opcji, które mogą działać. Kluczową rzeczą, o której należy pamiętać, jest zarządzanie stanem w sesjach i ograniczanie interakcji graczy do regionów geograficznych. Najlepsze opcje to platformy, które zapewniają serwer, który możesz dostosować do swoich potrzeb.

- Stwórz własny serwer: jeśli już wykonałeś poważny skok w dostarczaniu własnych usług danych GIS, jest to z pewnością opcja. Zajmiemy się tą koncepcją nieco dalej w następnej sekcji.

- Korzystaj z internetowej bazy danych w chmurze w czasie rzeczywistym: może to brzmieć jak gotowe i zwariowane rozwiązanie, ale jest to realna opcja i taka, którą poważnie omówimy w następnej sekcji dotyczącej sieci dla wielu graczy.

Mam nadzieję, że ta szczerza dyskusja na temat problemów z tworzeniem gry opartej na lokalizacji nie odstraszyła cię od kontynuowania. Jak wspomnieliśmy na początku książki, gry oparte na lokalizacji wiążą się z wieloma unikalnymi i trudnymi problemami, z obsługą trybu wieloosobowego na szczycie listy, ale w następnej sekcji omówimy niektóre strategie rozwoju gra wieloosobowa oparta na lokalizacji.

Gra wieloosobowa oparta na lokalizacji

Dodanie obsługi trybu wieloosobowego do gry opartej na lokalizacji, jak omówiono w poprzedniej sekcji, nie będzie łatwe. W rzeczywistości uniknęliśmy obsługi trybu wieloosobowego w naszej grze demonstracyjnej ze względu na dodatkową złożoność. Ponadto ważne było, aby zademonstrować, że nadal można zbudować funkcjonalną grę opartą na lokalizacji bez serwerów zaplecza i trybu wieloosobowego. W końcu jednak zastanawiasz się, jak dodać obsługę trybu wieloosobowego do swojej gry. W poprzedniej sekcji omówiliśmy kilka realnych opcji dodania obsługi trybu wieloosobowego: tworzenie własnego serwera, rozszerzanie istniejącej platformy i korzystanie z internetowej bazy danych w chmurze w czasie rzeczywistym. Te opcje mogą być atrakcyjne, ale zanim przejdziemy do konkretów, przyjrzyjmy się podstawowym problemom gier wieloosobowych opartych na lokalizacji:

* Gra jest ciągła: Nasza gra musi stale zapisywać stan dla wszystkich graczy na całym świecie. Jeśli nie doceniasz wielkości tego ostatniego zdania, poświęć chwilę i pomyśl o tym. Gdy gracze ponownie się połączą, cały otaczający ich świat musi zostać zaktualizowany niemal natychmiast, niezależnie od ich ostatniej lokalizacji. W zależności od ilości stanu, który oszczędzasz, może to być trudny problem. Dlatego w naszych lokalizacjach gier polegaliśmy na tych usługach zaplecza Google.

* Gracze muszą wchodzić w interakcje tylko lokalnie, jeśli w ogóle: Gry oparte na lokalizacji spotkały się z dużą krytyką i troską o umożliwienie graczom interakcji ze sobą, szczególnie w przypadku gier skierowanych do młodszych odbiorców. Dlatego generalnie będziesz chciał pozwolić swoim graczom na interakcję tylko za pośrednictwem miejsc, sklepów lub innych wirtualnych konstrukcji w regionie. Przykładem interakcji regionalnej może być np. ustawienie przynęty w danym miejscu. Wszyscy gracze mogliby czerpać korzyści z przynęty, ale nie wchodzić w bezpośrednią interakcję ze sobą.

* Stan gry wymaga filtrów regionalnych: gracze powinni mieć możliwość oglądania lub interakcji ze światem tylko w obrębie swojego obszaru mapy; nawet nasza zdolność do przeszukiwania usługi Miejsc Google z lokalizacji i promienia wymagała dalszego filtrowania. W idealnym przypadku stan gry, który gracz musi wyciągnąć, powinien obsługiwać jakąś formę wyszukiwania regionalnego. Może to być trudne, zwłaszcza jeśli masz ograniczone doświadczenie z GIS. Na szczęście wkrótce znajdziemy rozwiązanie tego problemu.

Teraz, gdy rozumiemy trudności, z jakimi boryka się każde z rozwiązań, przejrzymy listę funkcji, które każde rozwiązanie będzie musiało rozwiązać w poniższej tabeli:

Funkcje/wymagania: Rozwiń swój własny serwer: Rozszerz istniejący serwer dla wielu graczy: Baza danych w chmurze w czasie rzeczywistym

Bezpieczeństwo (dostęp): Trzeba opracować własną bazę danych dostępu i użytkowników. : Użyj lub dostosuj istniejący dostęp : Zapewnia solidną, bezpieczną platformę z wieloma opcjami dostępu dla użytkowników

Bezpieczeństwo (dane): Konieczność opracowania własnego mechanizmu bezpieczeństwa: Prawdopodobnie obsługuje ochronę danych i oszukiwanie graczy: Może być konieczne dostosowanie struktury bazy danych w celu zapewnienia bezpieczeństwa danych

Stan gry: Wymagaj bazy danych zaplecza, aby zachować stan: Prawdopodobnie już implementuje bazę danych: Jest bazą danych

Ciągły stan gry : Pełna kontrola nad bazą danych : Potrzebujesz serwera obsługującego gry MMO lub MMORPG : Pełna kontrola nad bazą danych

Izolowanie interakcji graczy według regionu: użyj geohash, aby izolować graczy według regionu: prawdopodobnie trzeba będzie dostosować świat, aby używał regionów i/lub geohash: użyj geohash, aby izolować graczy według regionu

Ogranicz aktualizacje stanu gry według regionu : Zarządzane przez geohash : Zarządzane przez niestandardowy geohash : Zarządzane przez geohash

Skalowalność. Dodawanie graczy : Konieczność zarządzania dodatkowymi serwerami i infrastrukturą : Konieczność zarządzania dodatkowymi serwerami i infrastrukturą, ewentualnie dodatkowe opłaty licencyjne : Rozwiązanie w chmurze umożliwiające rozbudowę

Kopie zapasowe danych: Konieczność obsługi tego samodzielnie: Prawdopodobnie konieczna jest obsługa samodzielnie: Prawdopodobnie obsługuje kopie zapasowe baz danych, ale tylko na pewnym poziomie

Infrastruktura (serwery): Potrzebujesz co najmniej jednego serwera działającego przez cały czas : Potrzebujesz co najmniej jednego serwera działającego przez cały czas : Obsługa chmury

Cena : Niedrogie lub bezpłatne, w zależności od dostępnej infrastruktury (serwery) : Cena oprogramowania do gry wieloosobowej plus koszty infrastruktury (serwer) : Prawdopodobnie na początek za darmo

Geohash to metoda hierarchicznego przedstawiania danych przestrzennych w siatce przy użyciu unikalnego identyfikatora, który jest sekwencją znaków. Jest to prawdopodobnie trudne do zrozumienia bez przykładu, więc poniższy diagram może pomóc w wizualnym wyjaśnieniu tego:

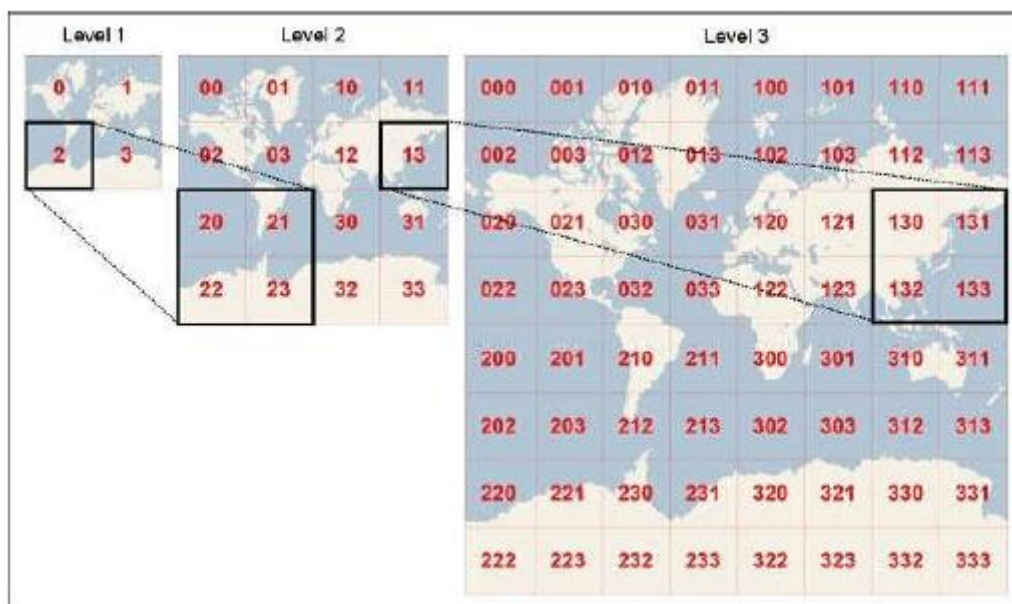


Diagram pokazujący, jak geohash dzieli świat na różne poziomy. Siatka pokazana na poprzednim schemacie jest bardziej reprezentatywna dla geohash kafelkowego, podczas gdy prawdopodobnie użyjemy bardziej standardowego Geohash-36, co oznacza po prostu, że każdy poziom jest podzielony na 36 sekcji. Sprawdź <http://geohash.gofreerange.com/>, jeśli chcesz zobaczyć, jaki jest geohash dla Twojego adresu lub miasta. Kiedy już możemy wyizolować lokalizację lub region tylko do jego reprezentacji geohash, pozwala nam to na bardzo szybkie i wydajne porządkowanie wiadomości według regionu. Na przykład, jeśli wiemy, że gracz jest w geohash 9q8y (obszar San Francisco, Kalifornia), możemy odizolować wszystkie wiadomości, które zaczęły się od tego geohash. Dlatego jeśli inny gracz na wyspie Alcatraz ma geohash 9q8zn i ponieważ pierwsze cztery znaki są różne, nie będą oglądać aktywności od drugiego. Poniższy zrzut ekranu z <http://geohash.gofreerange.com/> pokazuje to wizualnie:



Jeśli przyjrzyj się uważnie diagramowi mapy, możesz zobaczyć, jak siatka geohash jest dalej podzielona na Alcatraz. W rzeczywistości możesz zejść prawie do punktu lokalizacji (poziom geohash 12), ale prawdopodobnie nigdy nie będziesz musiał korzystać z tego poziomu. W tym przykładzie

używamy 4 poziomu geohash, ale to nie znaczy, że możesz używać wyższego lub niższego poziomu geohash do kontrolowania wiadomości regionalnych.

Uwaga: biblioteka geohash-36 została dołączona do końcowego kodu źródłowego FoodyGo w folderze C9A zawartym w pobranym kodzie źródłowym.

Jak widać w powyższej tabeli, ogólnie rzecz biorąc, wskazuje się na użycie bazy danych w chmurze w czasie rzeczywistym. Oczywiście nadal możesz mieć własne uprzedzenia do konkretnego rozwiązania, ale miejmy nadzieję, że lepiej zrozumiesz niektóre pułapki każdego z nich. Ta tabela stara się sprostać wysokim wymaganiom, ale rozwój oprogramowania jest zawsze o szczegółach. Na rynku dostępnych jest wiele baz danych działających w chmurze w czasie rzeczywistym, ale tylko kilka z nich obsługuje bezpośrednią integrację z Unity. Po pierwsze, jest Firebase od Google, która jest platformą aplikacji, która obsługuje bazę danych w czasie rzeczywistym i wiele innych funkcji, takich jak analityka, reklamy, raportowanie o awariach, hosting, przechowywanie i wiele innych. Po drugie, w chwili pisania tego tekstu Unity opracowuje bazę danych czasu rzeczywistego, która prawdopodobnie będzie obsługiwać wszystkie te same funkcje, co Firebase. Niestety wydaje się, że Unity zawsze jest o krok za swoimi konkurentami, a ta opcja jest obecnie w zamkniętej wersji alfa, co oznacza, że w następnej sekcji omówimy, jak można wykorzystać Firebase jako platformę dla wielu graczy.

Firestore jako platforma dla wielu graczy

Dla wielu programistów korzystanie z opartej na chmurze bazy danych czasu rzeczywistego może być rozwiązaniem, które nie jest warte rozważenia. Oczywiście zależy to również w dużej mierze od rodzaju gry, którą tworzysz. Prawdopodobnie nie chciałbyś używać bazy danych czasu rzeczywistego do FPS, ale na szczęście Unity ma do tego doskonałe opcje. Jednak w przypadku gry opartej na lokalizacji baza danych w czasie rzeczywistym jest idealna ze wszystkich powodów, o których wspomnieliśmy wcześniej.

Uwaga : nawet jeśli nie tworzysz gry opartej na lokalizacji i takiej, która nie wymaga aktualizacji o bardzo wysokiej wydajności, powinieneś rozważyć Firestore. Bezpłatna wersja Firestore w czasie rzeczywistym umożliwia 100 jednoczesnych użytkowników, a subskrypcja pierwszego poziomu zapewnia nieograniczoną liczbę jednoczesnych użytkowników. Inne platformy dla wielu graczy zazwyczaj zaczynają ładować po 20 jednoczesnych użytkownikach.

Nie dodamy Firestore jako platformy dla wielu graczy do przykładowej gry, którą opracowaliśmy w toku tej książki. To prawdopodobnie wymagałoby kilku rozdziałów lub nawet samej książki, aby omówić liczne szczegóły. Zamiast tego przyjrzymy się konfigurowaniu i używaniu jednej z próbek Firestore. Najpierw postępuj zgodnie z podanymi wskazówkami, aby pobrać przykładowy projekt bazy danych Firestore SDK:

1. Otwórz przeglądarkę internetową i przejdź do następującego adresu URL:

<https://github.com/firebase/quickstart-unity>.

2. Postępuj zgodnie ze wskazówkami na stronie GitHub, aby pobrać próbki SDK jako plik ZIP lub użyć Git do sklonowania repozytorium na komputer. Jeśli pobrałeś próbki jako plik ZIP, rozpakuj je do folderu, który możesz później zlokalizować.

3. Uruchom nową instancję Unity i kliknij przycisk Otwórz w górnej części okna wyboru projektu. Przejdź do folderu, do którego rozpakowałeś lub sklonowałeś próbki SDK. Następnie otwórz folder bazy danych i wybierz folder testapp jako folder projektu Unity. Kliknij przycisk Otwórz, aby załadować projekt.

4. Po otwarciu projektu zobaczysz błędy kompilacji. To normalne i nie ma się czym martwić. Wkrótce je rozwiążemy.

5. Otwórz MainScene, klikając dwukrotnie scenę MainScene w folderze Assets/TestApp w oknie projektu. Scena zostanie załadowana pusta.

Wskazówka: jeśli Unity nadal korzysta z układu mobilnego, możesz chcieć

do końca tego ćwiczenia zmienić układ z powrotem na domyślny. Z menu wybierz Okno | Układy | Domyślna.

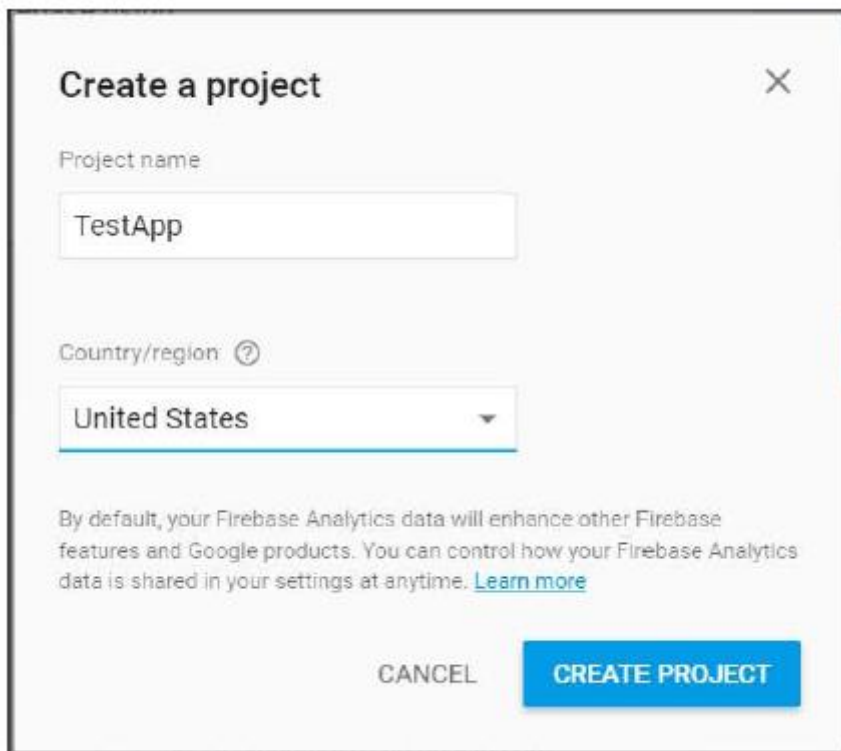
Po skonfigurowaniu przykładowego projektu bazy danych i gotowym do pracy, teraz musimy utworzyć konto Firebase i przeprowadzić pewną konfigurację:

1. Użyj przeglądarki internetowej, aby przejść do witryny Firebase pod adresem <https://firebase.google.com/>.

2. W górnej części tej strony będzie znajdować się duży przycisk z tekstem Rozpocznij za darmo. Kliknij ten przycisk i po wyświetleniu monitu zaloguj się na swoje konto Google, najlepiej to samo, którego wcześniej użyłeś do utworzenia klucza Google Places API.

3. Po zalogowaniu zostaniesz przeniesiony do Konsoli i poproszony o zaimportowanie lub utworzenie nowego projektu. Na razie po prostu stwórzmy nowy projekt.

4. Zostaniesz poproszony o nazwanie projektu i wybranie regionu. Nazwij swój projekt TestApp, a następnie wybierz region z listy, który pasuje do Twojej lokalizacji, jak pokazano:



Create a project [X]

Project name

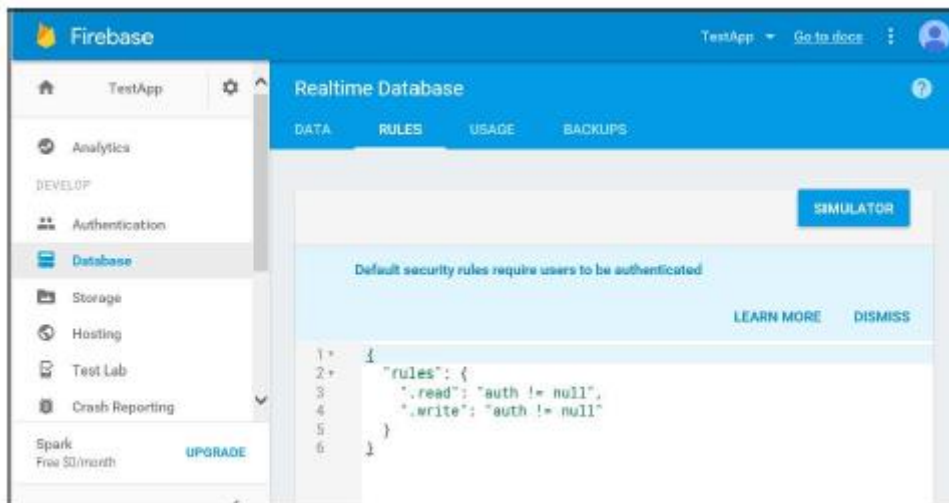
Country/region ?

By default, your Firebase Analytics data will enhance other Firebase features and Google products. You can control how your Firebase Analytics data is shared in your settings at anytime. [Learn more](#)

CANCEL CREATE PROJECT

5. Następnie kliknij przycisk Utwórz projekt, aby utworzyć nowy projekt. Po utworzeniu projektu zostaniesz przeniesiony do Konsoli Firebase projektu.

6. W panelu po lewej stronie wybierz element Baza danych, aby otworzyć panel Baza danych czasu rzeczywistego. Następnie w panelu wybierz zakładkę Reguły u góry. Poniższy zrzut ekranu jest pokazany w celach informacyjnych:



7. Na potrzeby tego przykładu tak naprawdę nie chcemy zajmować się konfiguracją zabezpieczeń. Oczywiście w rzeczywistej aplikacji lub grze bezpieczeństwo powinno być priorytetem. Możesz wyłączyć zabezpieczenia bazy danych odczytu/zapisu, ustawiając każdą z wartości na true, jak pokazano na poniższym zrzucie ekranu:



8. Po wprowadzeniu zmian w wartościach należy kliknąć przycisk Publikuj u góry panelu.

Wskazówka : Czy zauważyłeś, że wszystko w bazie danych wygląda jak JSON? Dzieje się tak, ponieważ baza danych czasu rzeczywistego jest w całości w formacie JSON.

9. Wróć do zakładki Dane. Zwróć uwagę na komentarz dotyczący domyślnych reguł bezpieczeństwa. Ponownie chcemy wyłączyć zabezpieczenia dla tego przykładowego testu, więc kliknij link Odrzuć obok powiadomienia. Spowoduje to wyłączenie uwierzytelniania pierwszego poziomu.

10. Na górze zakładki Dane będzie adres URL projektu, który powinien zawierać nazwę, którą ustawiłeś dla projektu (testapp). Skopiuj cały adres URL do schowka, zaznaczając tekst, a następnie naciskając Ctrl + C (polecenie + C na Macu). Pamiętaj, aby pozostawić konsolę Firebase otwartą w oknie przeglądarki.

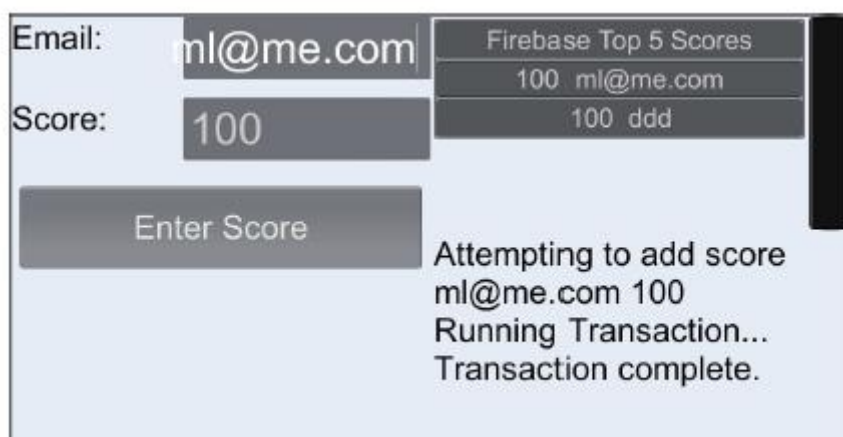
Wskazówka : Firebase obsługuje standardową i niestandardową autoryzację OAuth, Google, Facebook, Twitter i obsługę uwierzytelniania sfederowanego. Obsługuje również wiele poziomów reguł bazy danych, które można zastosować do poziomu węzła.

Teraz, po skonfigurowaniu bazy danych czasu rzeczywistego Firebase, wrócimy do Unity i skonfigurujemy projekt:

1. Kliknij dwukrotnie skrypt UIHandler znajdujący się w folderze Assets/TestApp w oknie projektu, aby otworzyć skrypt w wybranym edytorze.
2. Przewiń plik w dół i zlokalizuj metodę InitializeFirebase, jak pokazano w poniższym kodzie, do przeglądu:

```
void InitializeFirebase() {  
  
    FirebaseApp app =  
    FirebaseApp.DefaultInstance;  
    FirebaseApp.DefaultInstance;  
    app.SetEditorDatabaseUrl("https://YOURFIREBASEAPP.  
    firebaseio.com/");
```

3. Wybierz tekst adresu URL, który jest podświetlony, i wklej adres URL skopiowany wcześniej do schowka, naciskając Ctrl + V (polecenie + V na Macu). Przejrzyj poprzedni zestaw kroków, jeśli chcesz ponownie skopiować ten adres URL do schowka.
4. Zapisz plik i wróć do Unity. Poczekaaj, aż skrypt ponownie się skompiluje i kliknij Odtwórz, aby uruchomić aplikację testową w edytorze.
5. W oknie gry zostanie wyświetlony interfejs użytkownika. Wprowadź adres e-mail i punktację, a następnie kliknij przycisk Wprowadź punktację. Spróbuj tego kilka razy, aby wprowadzić liczbę punktów. Zwróć uwagę, jak wyniki wypełniają się na liście, jak pokazano na poniższym zrzucie ekranu:



6. Zatrzymaj i ponownie uruchom grę. Zwróć uwagę, w jaki sposób wyniki są automatycznie dodawane z powrotem. Po zakończeniu testowania pozostaw grę uruchomioną w edytorze.

Wygląda więc na to, że prosta aplikacja zapisuje wyniki w bazie danych, ale teraz chcemy potwierdzić, że dane są zapisywane w bazie danych w chmurze i aktualizuje klientów w czasie rzeczywistym.

Postępuj zgodnie z podanymi instrukcjami, aby przetestować działanie bazy danych w czasie rzeczywistym:

1. Wróć do okna przeglądarki z konsolą Firebase.

Pierwszą rzeczą, którą zauważysz, jest to, że do bazy danych dodano nowe dziecko, zwane Leaders. Rozwiń następujące węzły i potwierdź, że wpisy pasują do przykładowych danych wprowadzonych w Unity, jak pokazano na poniższym zrzucie ekranu:



2. Edytuj wartości bezpośrednio w konsoli, a następnie przełącz się z powrotem do uruchomionej sesji Unity. Zwróć uwagę, jak wartości na liście wyników zmieniają się niemal natychmiast.

3. Spróbuj dodać lub usunąć wartości z listy Liderów w Konsoli. Zwróć uwagę, jak są one również automatycznie aktualizowane w kliencie Unity. Na tym polega moc bazy danych w czasie rzeczywistym i automatycznych aktualizacji klienta.

Na tym etapie, jeśli korzystałeś z innych rozwiązań sieciowych, możesz być tylko pod wrażeniem. W końcu Photon PUN i UNET obsługują bardzo podobne rzeczy od razu po wyjęciu z pudełka, a także zapewniają mechanizmy automatycznej aktualizacji obiektów transformacji i innych komponentów Unity. Jednak, jak już wcześniej wspomnieliśmy i omówiliśmy, te automatyczne transformacje mają niewielką wartość w grze opartej na lokalizacji, w której gracz zawsze znajduje się w centrum świata gry.

Uwaga: niektórzy czytelnicy mogą pomyśleć, że alternatywnym rozwiązaniem świata skoncentrowanego na graczu byłoby umieszczenie gracza w jego rzeczywistych współrzędnych świata w metrach. Chociaż może to działać, gdy odległości są małe (mniej niż 100 000 metrów dla 0,0), ilość błędów, które wystąpiłyby w większości innej części świata, byłaby ogromna. Oczywiście gry oparte na lokalizacji nie są jedynym miejscem, które dobrze nadaje się do tworzenia bazy danych w chmurze czasu rzeczywistego. Inne odpowiednie gatunki to plansza, budowanie imperium, strategia, puzzle, karty i inne. Mówiąc o innych rodzajach gier, co powiesz na sugestie dotyczące różnych rodzajów gier opartych na lokalizacji w poniższej sekcji?

Inne pomysły na gry oparte na lokalizacji

Chociaż gra demonstracyjna miała pokazać, jak została skonstruowana ta popularna gra, miejmy nadzieję, że zaintrygowała Cię do myślenia o zbudowaniu własnej gry. Oczywiście nie chciałbyś

kopiować popularnej gry, tak jak my, więc spójrzmy na kilka innych opcji możliwych gier opartych na lokalizacji, w następujący sposób:

* Strategia (budowanie imperium): Być może twoi gracze wcielą się w rolę króla, władcy lub potentata biznesowego z zamiarem budowania lub zabezpieczania zasobów wokół nich. Na przykład Parallel Kingdom, Parallel Mafia, Resources i Turf Wars.

* Paranormal (przetrwanie): Być może widzisz swoich graczy jako łowców duchów lub zombie, którzy albo muszą polować na zjawiska paranormalne, albo po prostu unikać bycia zjedzonym. Na przykład Zombies, Run i Spec Trek.

* Poszukiwanie skarbów: Być może twoi gracze są poszukiwaczami skarbów

którzy muszą szukać wskazówek i przeszukiwać obszary w poszukiwaniu ukrytych

skarb, czy to prawdziwy, czy wirtualny. Na przykład Geocaching, Zaploot.

* Szpieg (szpiegostwo): Twoi gracze pracują dla tajnej organizacji i są wysyłani na misje w okolicy, aby odkrywać, badać i zapobiegać sukcesom innych frakcji. Na przykład Ingress, CodeRunner.

* Tower Defense: Pozwól swoim graczom chronić obszary wokół ich sąsiedztwa przed zewnętrznymi wirtualnymi atakami w rzeczywistej grze typu tower-defense. Na przykład Geoglif.

* Polowanie (śledzenie): To nie jest gra dla młodszych graczy. Twoi gracze wcielają się w rolę myśliwego i ściganego. Łowcy muszą poruszać się w określonej odległości fizycznej, aby złapać ściganego gracza. Na przykład przesunięcie.

* Odgrywanie ról (RPG): Gracz wciela się w postać i może poruszać się po świecie rzeczywistym jako ta postać, wykonując wirtualne czyny, dołączając do grup i wchodząc w interakcje z innymi graczami. Na przykład Kingpin: Życie zbrodni.

* Kolekcjonerzy: To jest jak poszukiwanie skarbów, z tą różnicą, że gracze zwykle robią potem coś z zebranymi przedmiotami. Na przykład Pokemon Go oczywiście musiało być na liście.

Przyszłość gatunku

Biorąc pod uwagę popularność Pokemon Go i zmianę kulturową, która nastąpiła z jej powodu, można jedynie założyć, że gatunki oparte na lokalizacji i AR są obecnie głównymi platformami do gier. Wcześniej funkcje GPS były rzadko używane na urządzeniach mobilnych, a teraz stały się głównym nurtem. Ta zmiana nie tylko zachęci do lepszych możliwości GPS na urządzeniach mobilnych, ale to z kolei pozwoli aplikacjom i grom pobierać dokładniejsze dane o lokalizacji. Również zużycie energii GPS prawdopodobnie zostanie zmniejszone, a oba te czynniki będą miały pozytywny wpływ na twórców aplikacji lub gier opartych na lokalizacji. Ponieważ aplikacje i gry zaczną zużywać coraz więcej danych GPS i GIS z usług takich jak Google lub inne, można założyć, że ceny ulegną obniżeniu, a ograniczenia danych wzrosną. To może wydawać się sprzeczne z intuicją, ale Google jest w stanie świadczyć te usługi tanio dzięki śledzeniu użytkownika. Google będzie wtedy zazwyczaj wykorzystywać to śledzenie użytkownika do obsługi usług, takich jak analiza przepływu ruchu lub inne usługi. W przypadku Google im więcej śledzi użytkowników, tym lepsze są metryki danych. Dlatego Mapy Google są całkowicie bezpłatne dla aplikacji na Androida lub iOS. Biorąc pod uwagę te czynniki, zdecydowanie sugeruje, że gatunek oparty na lokalizacji stanie się bardziej popularny wśród graczy i programistów. Ciekawie będzie zobaczyć, jakie będą następne duże tytuły w tym gatunku.

Podsumowanie

Ta Część była podsumowaniem wysiłków rozwojowych, które podjęliśmy. Najpierw przyjrzelśmy się, jakie przedmioty musiałyby zostać ukończone, aby wysłać grę Foody Go. Następnie przyjrzelśmy się ważnym umiejętnościom, które byłyby cenne przy spędzaniu czasu na nauce. Następnie dokładnie przyjrzelśmy się wskazówkom dotyczącym nabywania zasobów Unity, które mogą uzupełnić brakujące zestawy umiejętności. Poniżej znajduje się krótkie ćwiczenie dotyczące zarządzania i usuwania niechcianych zasobów pozostających w projekcie. Od tego momentu omówiliśmy kilka wskazówek dotyczących wydania gry. Następnie zmieniliśmy biegi i staraliśmy się odkryć wszystkie problemy, z którymi boryka się programista podczas tworzenia gry opartej na lokalizacji; to zaprowadziło nas do dyskusji na temat wspierania wielu graczy w naszej grze. Jak jednak widzieliśmy, w grach opartych na lokalizacji pojawiło się wiele unikalnych problemów, które uniemożliwiają korzystanie z gotowych rozwiązań. Zamiast tego przyjrzelśmy się możliwości implementacji niestandardowych rozwiązań dla wielu graczy, a następnie poważnie przyjrzelśmy się działającej w czasie rzeczywistym bazie danych Google o nazwie Firebase Real-time. Przeprowadziliśmy krótkie ćwiczenie, aby skonfigurować i skonfigurować Firebase do użytku w Unity jako platforma dla wielu graczy. Stamtąd przyjrzelśmy się kilku innym pomysłom na gry oparte na lokalizacji, aby uzyskać inspirację. Na koniec przyjrzelśmy się, co może przynieść przyszłość gatunku opartego na lokalizacji.