

Mapowanie lokalizacji gracza

Rdzeniem większości gier w świecie rzeczywistym jest mapa oparta na lokalizacji. Ta integracja pozycji gracza w świecie rzeczywistym odwzorowana w świecie wirtualnym rozszerza wirtualne elementy gry na świat rzeczywisty, pozwalając graczom odkrywać otaczający ich świat z nową perspektywą i poczuciem eksploracji. W tej części zrobimy pierwsze kroki, aby zrozumieć, jak zintegrować mapy z grą Unity. Jednak przed rozpoczęciem pracy z Unity omówimy podstawy GIS i GPS. Umożliwi nam to ustalenie kilku prostych definicji i tła tego, co może być złożonym tematem. Następnie wrócimy do Unity i do naszego projektu Foody GO dodamy mapę opartą na lokalizacji, podstawową postać i kamerę o swobodnym wyglądzie. Chociaż mamy do czynienia z zaawansowanymi koncepcjami, na razie zachowamy prostotę i unikniemy wchodzenia w jakikolwiek kod. Oczywiście bardziej zaawansowani czytelnicy z doświadczeniem w GIS mają możliwość eksplorowania kodu w wolnym czasie. Oto krótkie zestawienie tego, co omówimy:

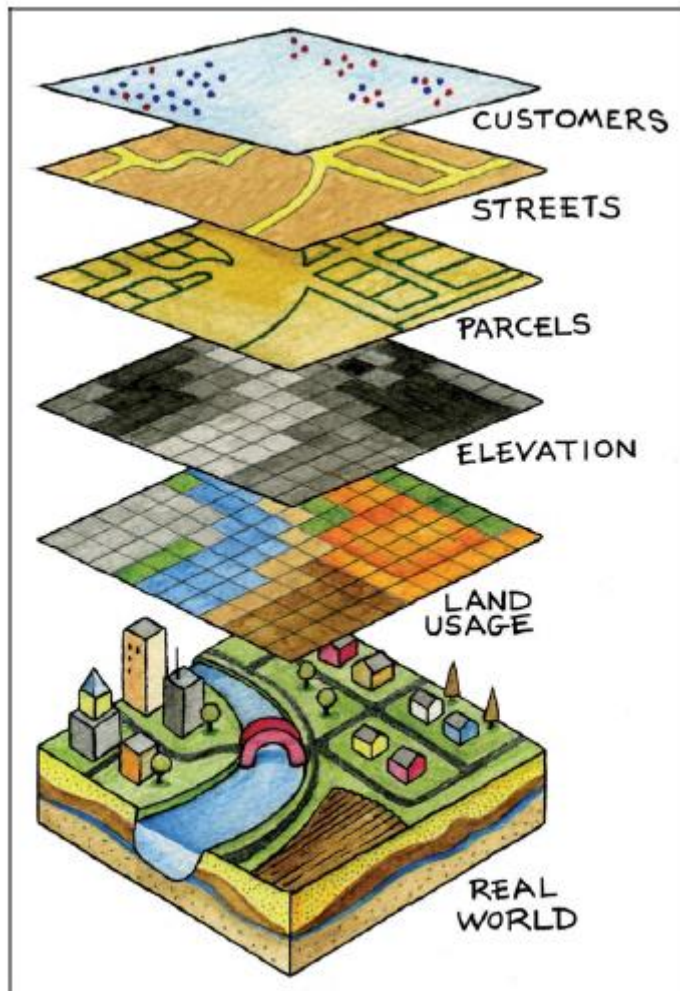
- * Terminologia GIS i podstawowe podstawy
- * Terminologia i podstawy GPS
- * Mapy Google
- * Importowanie aktywów
- * Konfigurowanie usług
- * Debugowanie za pomocą CUDLR

Podstawy GIS

GIS oznacza system informacji geograficznej, system, w którym dane geograficzne są gromadzone, przechowywane, analizowane, przetwarzane i wykorzystywane jako mapy. Chociaż ta definicja jest nadal widoczna, GIS zaczął oznaczać wszystko, od aplikacji, sprzętu, narzędzi, nauki i usług. Na przykład Mapy Google są najbardziej znanym przykładem obecnie używanego GIS. Jednak w tej książce będziemy również używać GIS w znaczeniu nauki i procesu konwersji danych geograficznych i mapowania.

Mapowanie

Kartografowie od tysięcy lat tworzą mapy otaczającego ich świata. Jednak dopiero niedawno, wraz z rozwojem komputerów, wydajne stało się wykorzystanie ich w komputerach i GIS do tworzenia dynamicznych map. W przeciwieństwie do tradycyjnych ręcznie rysowanych map papierowych, mapy renderowane dynamicznie składają się z warstw danych przestrzennych, które mogą opisywać drogi, ciekawe miejsca, parki, granice, krajobraz, wodę i nie tylko. Chociaż zwykły użytkownik Map Google ma niewielką kontrolę nad wyświetlanymi warstwami, są one dostępne. Poniżej znajduje się rysunek przedstawiający typowe warstwy, które mogą składać się na mapę drogową:



Mapy Google, Bing i inni dostawcy GIS zazwyczaj renderują mapy przy różnych poziomach powiększenia, a następnie tną je na statyczne kafelki obrazów. Serwer GIS następnie udostępnia te kafelki użytkownikowi. Działa to dobrze w przypadku wydajności, ale zapewnia kilka opcji dostosowywania lub stylizacji, oprócz dodawania kształtów, linii lub interesujących miejsc. Ta forma mapowania kafelków jest często określana jako mapowanie statyczne. W przypadku tej książki będziemy pracować z interfejsem API Map Google, który zapewnia nam dynamiczne mapy. Mapy dynamiczne dają prezenterowi lub programiście opcje stylizacji i/lub symbolizowania danych zgodnie z wymaganiami. Na przykład na niektórych mapach możesz chcieć zmienić parki, aby były wyświetlane na niebiesko zamiast na zielono. Jest to rodzaj elastyczności, jaką zapewniają mapy dynamiczne. Gdy w dalszej części dodamy Google Maps API do naszego projektu, zbadamy niestandardowe opcje stylów i symboli. Teraz, gdy rozumiemy podstawy mapowania GIS, przejdźmy do zrozumienia niektórych terminologii i pojęć. Poniżej znajduje się lista terminów, których używamy przy opisie lub tworzeniu map:

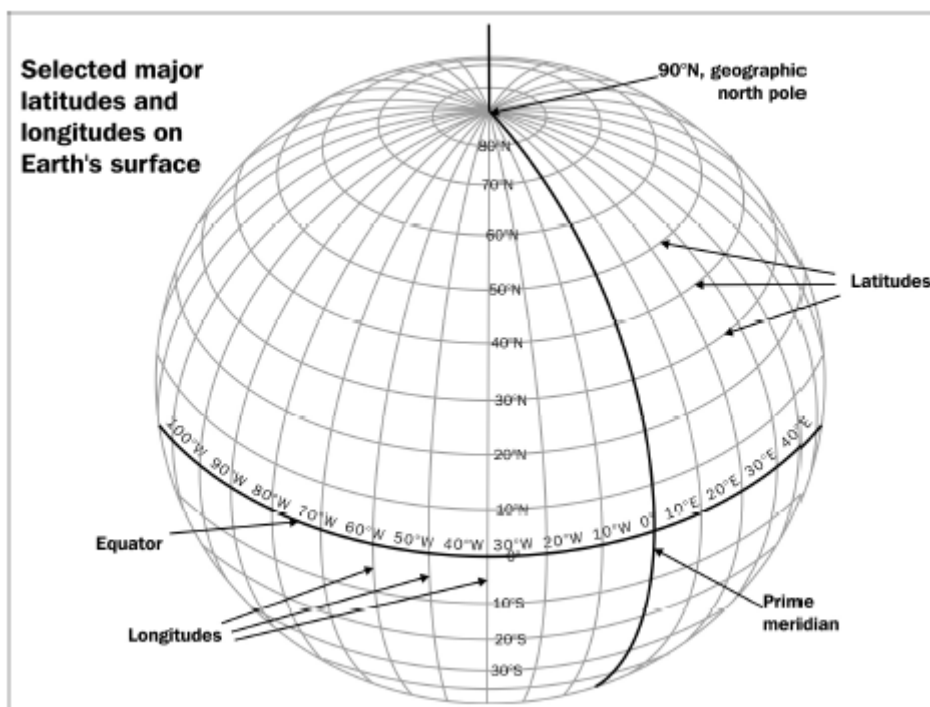
* Skala mapy: mapy geograficzne mogą przedstawiać wszystko, od Twojej okolicy po cały świat. Skala mapy jest często renderowana jako tekst lub grafika i zapewnia użytkownikowi zrozumienie zakresu, jaki obejmuje mapa.

* Poziom powiększenia: ma odwrotną zależność ze skalą mapy. Poziom powiększenia zaczyna się od 1, co reprezentuje globalny widok świata, podczas gdy poziom powiększenia 17 reprezentuje mapę

przedstawiającą Twoją okolicę. W naszej grze użyjemy małej skali mapy, aby umożliwić graczowi łatwe zidentyfikowanie otaczających ich punktów orientacyjnych, co będzie odpowiadać poziomowi powiększenia 17 lub 18.

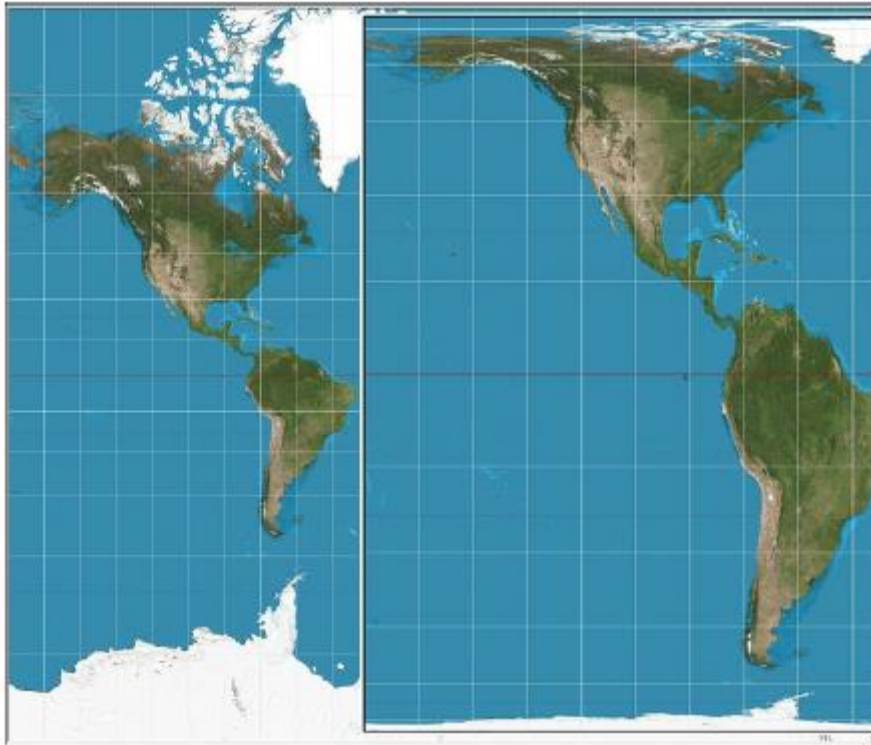
* Układ współrzędnych: różne układy współrzędnych były używane w historii do geograficznego lokalizowania interesujących miejsc. Wiele osób zna współrzędne szerokości i długości geograficznej, ale mogą nie zdawać sobie sprawy, że istnieje wiele różnych układów współrzędnych. W rzeczywistości wspólna szerokość i długość geograficzna, które uważamy za standardowe, również ma wiele odmian. Ponieważ będziemy pracować z Google Maps, we wszystkim, co robimy, będziemy używać współrzędnych WGS 84. Należy pamiętać, że jeśli spróbujesz zaimportować dane z innego GIS, mogą wystąpić różnice, które być może trzeba będzie przekonwertować.

Poniżej znajduje się diagram przedstawiający główne linie zainteresowania WGS 84:



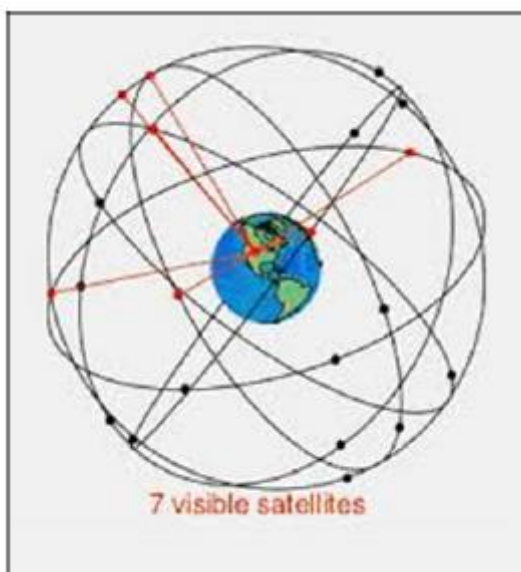
Długość geograficzna na zachód od południka zerowego jest mierzona jako ujemna. Szerokość geograficzna na południe od równika jest mierzona jako ujemna.

Projekcja mapy: Jednym z podstawowych problemów związanych z mapowaniem jest przedstawianie naszego trójwymiarowego świata w kształcie kuli w usłudze 2D. Wcześni kartografowie rozwiązali ten problem, rzucając światło z globu na cylinder papieru i śledząc kontur. Papier został następnie rozwinięty i widać było dwuwymiarowy widok kuli ziemskiej. Nadal używamy podobnej metody do renderowania obrazu naszego świata. Chociaż ten widok 2D jest zniekształcony, gdy znajdujesz się w pobliżu biegunów, stał się on standardem dla większości naszych potrzeb mapowania. Wystarczy powiedzieć, że istnieje wiele odmian odwzorowań map, które są wykorzystywane z różnych powodów. Od tego czasu opracowano prawdopodobnie lepsze metody projekcji, takie jak Gall-Peters, która lepiej wyjaśnia zniekształcenia na biegunach. Dla nas jednak będziemy trzymać się standardowego Google Web Mercator lub po prostu Web Mercatora Map Google. Poniżej znajduje się przykład mapy globalnej wykorzystującej projekcję Web Mercator po lewej stronie i projekcję Galla-Petersa po prawej:



Podstawy GPS

GPS to skrót od globalnego systemu pozycjonowania, sieci od 24 do 32 satelitów, które krążą wokół Ziemi dwa razy w ciągu godziny. Gdy te satelity krążą po orbicie, emitują zakodowane w czasie sygnały geograficzne jako latarnia morska dla każdego urządzenia GPS na Ziemi, które je widzi. Urządzenie GPS wykorzystuje następnie te sygnały do triangulacji swojej pozycji w dowolnym miejscu na planecie. Im więcej satelitów i sygnałów może odebrać urządzenie, tym dokładniejsza będzie lokalizacja. Poniższy schemat przedstawia urządzenie GPS odbierające sygnał z widocznych satelitów w sieci:



Oto lista terminów, na które możemy się natknąć, rozmawiając lub używając GPS na urządzeniu:

* **Datum:** Jest to termin używany w GPS do określenia systemu transformacji współrzędnych używanego do konwersji sygnałów satelitarnych na użyteczne współrzędne. Wszystkie urządzenia GPS używają standardu WGS 84, co jest dla nas wygodne, ponieważ nasze mapy również korzystają z WGS 84. Dedykowane urządzenie GPS obsługuje kilka różnych danych w zależności od potrzeb zaawansowanych użytkowników.

* **Szerokość/długość geograficzna:** Domyślnie urządzenie GPS zwraca współrzędne szerokości i długości geograficznej w układzie odniesienia WGS 84. Dla nas jest to łatwe, ponieważ nie wymagamy żadnych dodatkowych konwersji matematycznych, aby zmapować lokalizację urządzenia.

* **Wysokość:** reprezentuje wysokość urządzenia nad poziomem morza. Większość mobilnych urządzeń GPS, dla których będziemy budować naszą grę, obecnie nie obsługuje wysokości. Dlatego nie będziemy używać wysokości w grze, ale miejmy nadzieję, że będzie to wspierane w przyszłości.

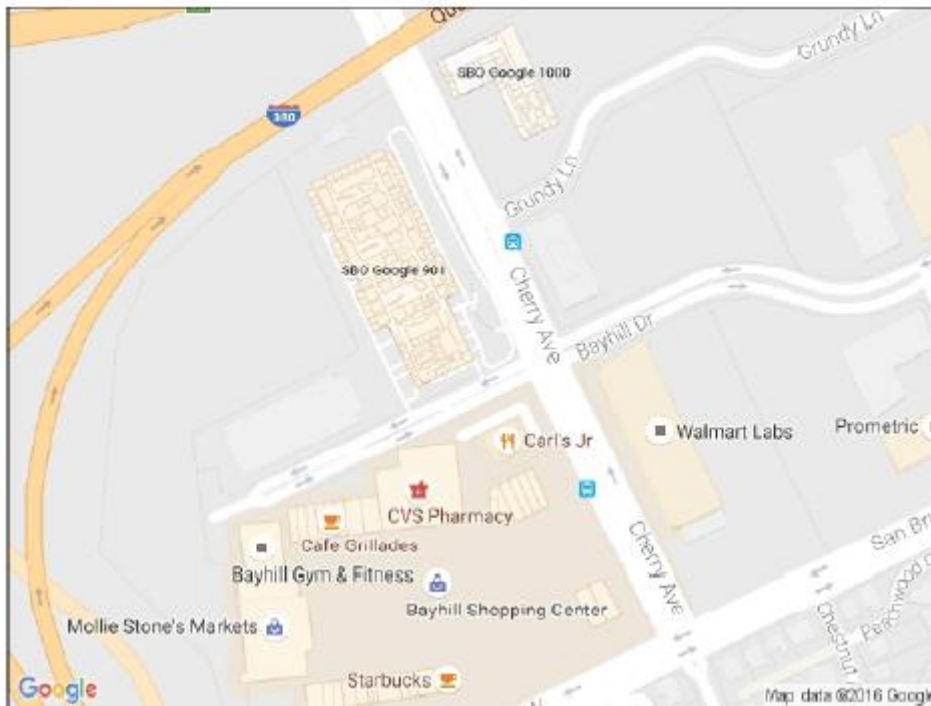
* **Dokładność:** jest raportowana przez urządzenie i reprezentuje zakres błędu obliczonego podczas określania lokalizacji. Im więcej satelitów urządzenie odbiera sygnały, tym lepsze będą obliczenia lokalizacji. Istnieje limit dokładności każdego urządzenia, a nawet tego, co jest możliwe w publicznej sieci GPS. Nowoczesny smartfon często podaje z dokładnością do 5-8 metrów. Jednak niektóre starsze smartfony mogą mieć nawet 75 metrów wysokości. Więcej czasu poświęcimy na omawianie dokładności GPS w dalszej części gdy zaczniemy umożliwiać graczowi interakcję z wirtualnymi obiektami.

Google Maps

Jak wcześniej wspomniano, w naszej usłudze mapowej będziemy używać Google Maps. W tej wersji gry użyjemy mapy statycznej, co oznacza, że nie potrzebujemy klucza programisty API ani nie musimy się martwić o jego użycie. Aby użyć API mapy statycznej, wywołujemy adres URL z szeregiem parametrów ciągu zapytania w żądaniu GET, tak jak wywołuje się typową usługę REST. Interfejs API Map Google zwraca następnie pojedynczy obraz pasujący do naszego żądania. Oto przykład żądania do usługi REST mapy statycznej Google Maps API:

```
https://maps.googleapis.com/maps/api/staticmap?center=37.62761,-  
122.42588&zoom=17&format=png&sensor=false&size=640x480&maptype=roadmap.
```

Spowoduje to wyrenderowanie następującego obrazu mapy:



Upewnij się, że przetestowałeś to samodzielnie, klikając łącze lub kopiując i wklejając ten adres URL do swojej ulubionej przeglądarki. Podzielmy ten adres URL na jego części składowe, abyśmy mogli zrozumieć elementy, które musimy zdefiniować podczas żądania obrazu mapy:

* <https://maps.googleapis.com/maps/api/staticmap>: reprezentuje podstawowy adres URL usługi Mapy Google. Jeśli wywołamy ten adres URL bez żadnych parametrów, otrzymamy błąd. Przyjrzyjmy się bardziej szczegółowo każdemu parametrowi i składni zapytania.

* `?`: Znak zapytania oznacza początek parametrów zapytania.

* `center=37.62761,-122.42588`: reprezentuje środek żądanej mapy we współrzędnych szerokości i długości geograficznej.

* `&`: Symbol ampersand oznacza początek nowego parametru zapytania.

* `zoom=17`: reprezentuje poziom powiększenia lub skalę, w jakiej chcemy renderować mapę. Pamiętaj z naszych podstaw GIS, że im wyższy poziom powiększenia, tym mniejsza skala mapy.

* `format=png`: jest to typ obrazu, który chcemy zwrócić. Do naszego użytku preferowany jest format PNG.

* `sensor=false`: Oznacza to, że nie używaliśmy GPS do uzyskania naszej lokalizacji. Później, gdy zintegrujemy GPS urządzenia mobilnego, zostanie to ustawione na true.

* `size=640x480`: reprezentuje rozmiar żądanego obrazu w pikselach.

* `maptype=roadmap`: To prosi o typ mapy. Istnieją cztery rodzaje map, o które możesz poprosić:

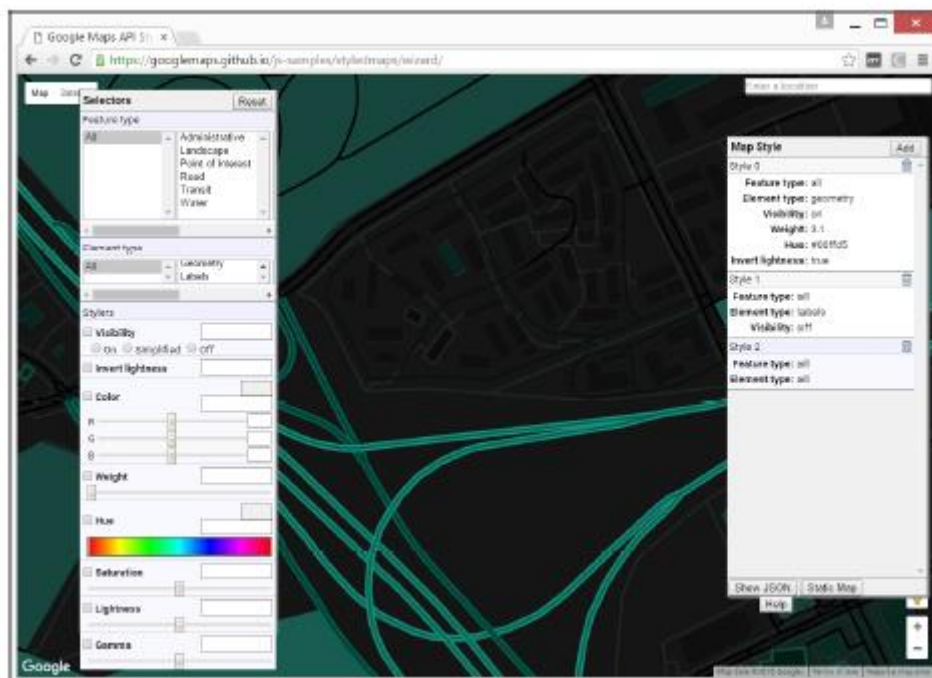
- `roadmap`: jest to mapa pokazująca ulice, tranzyt, obszary krajobrazowe, wodę i ciekawe miejsca.

- `satellite`: jest to mapa przedstawiająca rzeczywiste zdjęcia satelitarne.

- terrain: jest to mapa pokazująca wysokość zmieszana z mapą drogową.

- hybrid: jest to połączenie mapy drogowej z mapą satelitarną.

Na szczęście nie będziesz musiał generować tych adresów URL, ponieważ zrobią to już przygotowane pakiety skryptów. Jednak pomocne będzie zrozumienie, w jaki sposób te prośby o mapy są wysyłane, na wypadek, gdybyś chciał dostosować grę lub napotkać jakieś problemy. Jeśli pamiętasz, w naszej dyskusji na temat mapowania wspomnieliśmy, że mapy GIS są zawsze konstruowane w warstwach. Wspaniałą rzeczą w Mapach Google jest to, że możemy dynamicznie stylizować różne warstwy map w ramach żądania. To pozwala nam stylizować nasze mapy zgodnie z wybranym przez nas wyglądem i stylem gry. W naszej grze ustawiliśmy kilka prostych stylów, aby nadać naszej grze ciemniejszy wygląd. Poniższy zrzut ekranu przedstawia wybrane style tak, jak są one wyświetlane w kreatorze stylów Map Google:



Dodawanie mapy

Po naszym krótkim wprowadzeniu do mapowania za pomocą GIS i GPS, wróćmy do Unity i dodajmy mapę do naszej gry. W miarę tworzenia mapy ponownie przyjrzymy się niektórym z tych terminów GIS. Kontynuujmy to, co przerwaliśmy w poprzedniej części.

Tworzenie kafelka mapy

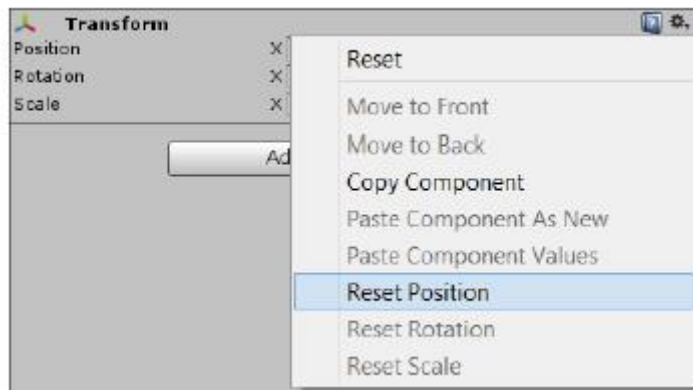
Postępuj zgodnie z instrukcjami tutaj, aby dodać mapę do gry:

1. Otwórz Unity i załaduj projekt FoodyGO, który stworzyliśmy w poprzedniej części aby załadować projekt.
2. Po otwarciu Unity powinieneś zobaczyć załadowany ekran powitalny. Jeśli nie jest załadowany, to w porządku, ponieważ będziemy tworzyć nową scenę mapy. Wybierz pozycję menu Plik | Nowa scena.
3. To utworzy nową pustą scenę w Unity tylko z główną kamerą i kierunkowym światłem. Zanim zapomnimy, uratujmy tę nową scenę. Wybierz pozycję menu Plik | Zapisz scenę jako... i w oknie dialogowym Zapisz scenę wprowadź nazwę Mapa jako nazwę pliku, a następnie kliknij Zapisz.

4. Wybierz scenę Mapa w oknie Hierarchia. Następnie wybierz pozycję menu GameObject | Utwórz pusty, aby utworzyć nowy pusty obiekt GameObject w scenie. Wybierz ten nowy obiekt gry i przejrzyj jego właściwości w oknie Inspektora.

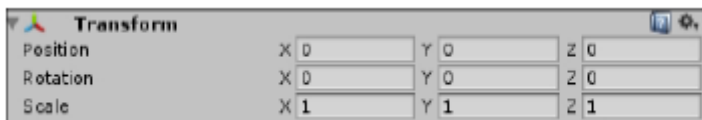
5. W oknie Inspektora zmień nazwę GameObject na Map_Tiles, edytując pole nazwy. Zresetuj pozycję transformacji obiektu, wybierając ikonę koła zębatego w komponencie Transformacja, a następnie wybierając opcję Zresetuj pozycję z menu rozwijanego.

Poniższy zrzut ekranu pokazuje, jak dokonać wyboru z menu rozwijanego:



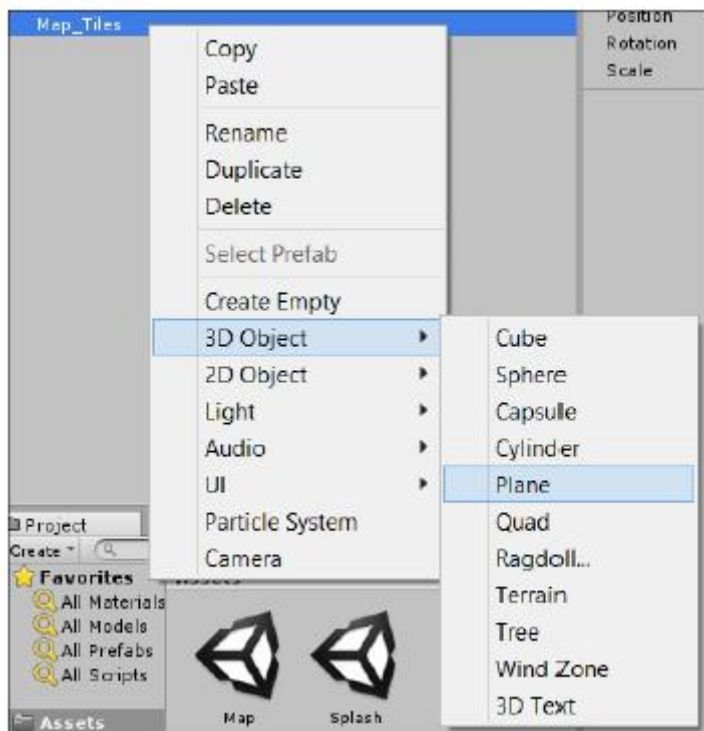
Resetowanie pozycji obiektu gry

1. Generalnie zresetujemy większość naszych obiektów w grze do lub w pobliżu zera, aby uprościć konwersje matematyczne GIS. Obiekt gry Map_Tiles powinien teraz mieć transformację zerową lub tożsamościową, jak pokazano na poniższym zrzucie ekranu:



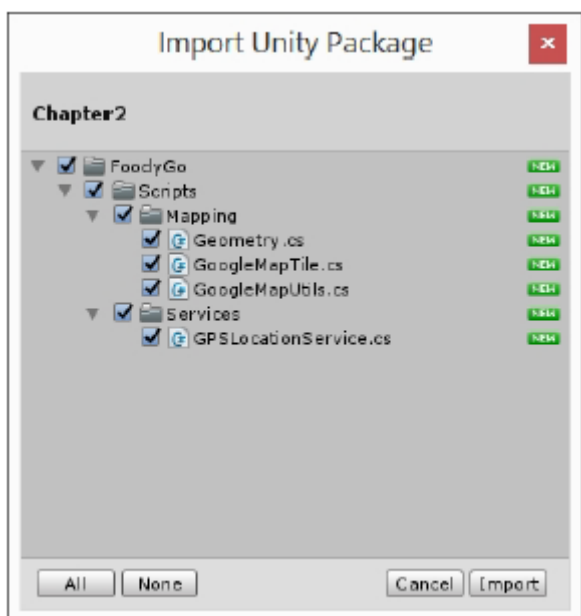
Obiekt gry z zerową transformacją

2. Po wybraniu obiektu gry Map_Tiles kliknij prawym przyciskiem myszy (naciśnij polecenie i kliknij na komputerze Mac), aby otworzyć menu kontekstowe i wybierz obiekt 3D | Samolot. Oto zrzut ekranu pokazujący, jak dodać samolot z menu kontekstowego:



Menu kontekstowe obiektu gry

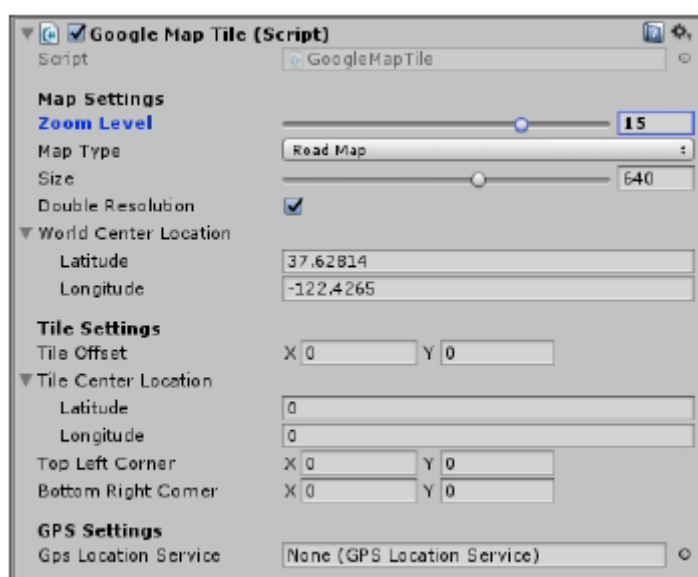
- Wybierz obiekt gry samolot i zmień jego nazwę na Map_Tile w oknie Inspektora. Upewnij się, że transformacja obiektu wynosi zero.
- Kliknij dwukrotnie płaszczyznę Map_Tile w oknie Hierarchy, aby skoncentrować się na obiekcie w oknie Scene. Jeśli obiekt nie jest widoczny, upewnij się, że przycisk 2D w oknie Scena jest wyłączony.
- W oknie Inspektora edytuj właściwości komponentu Transform, a następnie edytuj skalę X i Z na 10. Zwróć uwagę, jak rozszerzają się wymiary płaszczyzny podczas edycji skali.
- Teraz musimy dodać skrypt do naszego obiektu Map_Tile, który wyrenderuje naszą mapę. W tym momencie unikniemy tworzenia nowych skryptów i zamiast tego dodamy skrypt z importowanego zasobu. Oczywiście w dalszej części będziemy budować nowe scenariusze. Skrypty, których potrzebujemy, będą znajdować się w pobranym kodzie w folderze Chapter_2_Assets. Otwórz pozycję menu Zasoby | Importuj pakiet | Pakiet niestandardowy..., aby otworzyć okno dialogowe Importuj pakiet.
- Użyj okna dialogowego, aby przejść do folderu Chapter_2_Assets znajdującego się w folderze z pobranym kodem źródłowym i zaimportuj zasób Chapter2.unitypackage, klikając Otwórz.
- Pojawi się okno dialogowe postępu, które pokaże ładowanie zasobów, które zostaną szybko zastąpione oknem dialogowym Import Unity Package. Upewnij się, że wszystkie elementy w oknie dialogowym są zaznaczone i dostępne do zaimportowania, a następnie kliknij Importuj. Pojawi się okno dialogowe Import Unity Package z importowanymi skryptami:



9. Po zaimportowaniu zasobów zauważysz nowy folder utworzony w obszarze Zasoby w oknie Projekt. Zachęcamy do zapoznania się z tym nowym folderem i jego zawartością, aby zapoznać się ze sposobem organizowania zasobów projektu. Zwróć uwagę, że zawartość folderu dokładnie pasuje do tego, co zaimportowaliśmy.

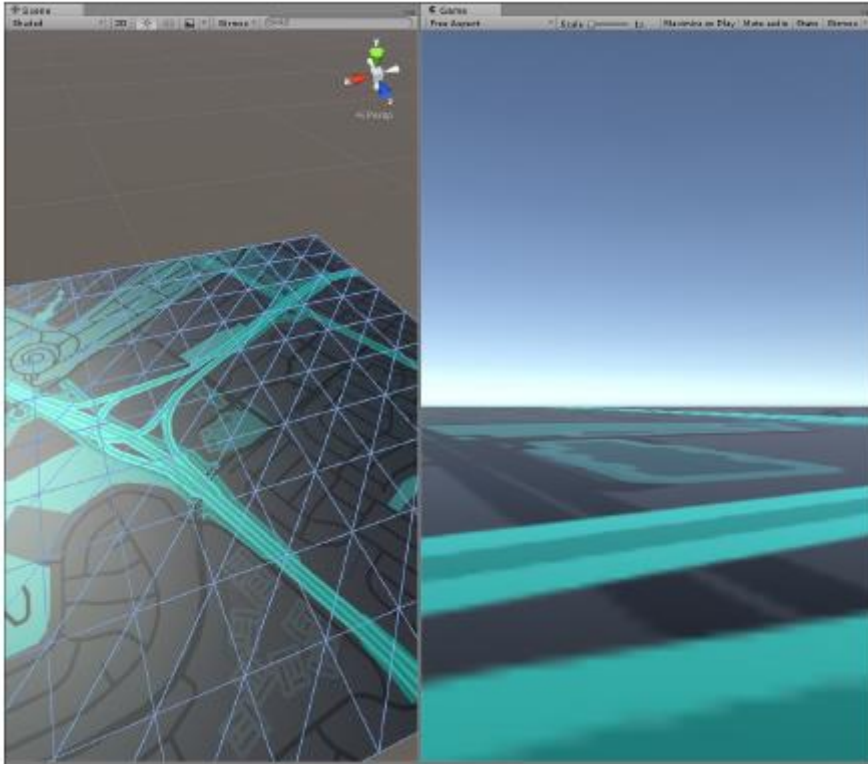
10. Wybierz obiekt Map_Tile w oknie Hierarchia. Kliknij przycisk Dodaj składnik widoczny na dole okna Inspektora. Otworzy się menu kontekstowe z listą komponentów; wybierz Mapowanie | Mapa kafelków Google. Spowoduje to dodanie komponentu skryptu mapowania Google do obiektu gry Map_Tile. Google Maps for Unity dostępne bezpłatnie w Unity Asset Store było inspiracją dla kodu mapy kafelkowej Google. Kilka rzeczy zostało zmienionych w bardziej zaawansowanej wersji używanej w grze.

11. W oknie Inspektora edytuj skrypt Google Map Tile, aby dopasować wartości w następujący sposób:



12. Mamy nadzieję, że te parametry mapowania mają sens po poprzednim wprowadzeniu. Na razie użyjemy poziomu powiększenia 15, aby sprawdzić, jak działa nasza marka. Te współrzędne lokalizacji dotyczą firmy Google zlokalizowanej w San Francisco. Oczywiście połączymy GPS urządzenia i wykorzystamy Twoje lokalne współrzędne później.

13. Naciśnij przycisk Odtwórz. Po kilku sekundach ekran powinien wyglądać podobnie do tego:



Oto krótkie podsumowanie kroków, zakładając, że wcześniej wdrożyłeś grę na swoim urządzeniu:

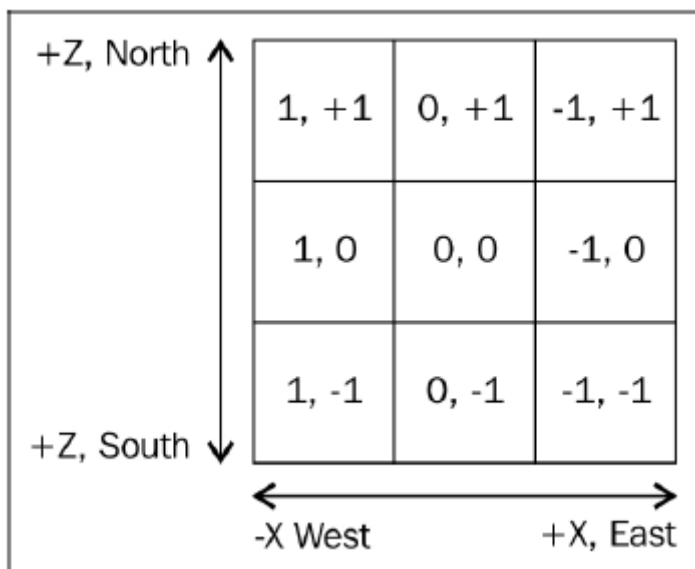
1. Upewnij się, że Twoje urządzenie jest podłączone do maszyny programistycznej Unity za pomocą kabla USB.
2. Zapisz swoją scenę, wybierając Plik | Zapisz scenę.
3. Zapisz swój projekt, wybierając Plik | Zapisz Projekt; pamiętaj, że zawsze dobrze jest zapisać grę przed budowaniem. Unity słynie z zawieszania się edytora.
4. Wybierz pozycję menu Plik | Ustawienia kompilacji..., aby otworzyć okno dialogowe Ustawienia kompilacji.
5. Odznacz scenę Splash, ponieważ jeszcze jej nie potrzebujemy. Kliknij przycisk Dodaj otwarte sceny, aby dodać scenę mapy do kompilacji.
6. Wybierz odpowiednią platformę wdrożeniową, Android lub iOS.
7. Kliknij przycisk Buduj i uruchom, aby rozpocząć proces budowania i wdrażania.
8. Po wyświetleniu monitu zapisz i nadpisz wdrożenie w tej samej lokalizacji, którą wybrałeś wcześniej.
9. Poczekaj, aż gra zakończy budowanie, a następnie zainstaluj ją na swoim urządzeniu.

10. Po załadowaniu gry na urządzeniu sprawdź mapę i obróć urządzenie. W tym momencie gra robi bardzo niewiele, ale to rzeczywiście pokazuje, że mapa działa na twoim urządzeniu.

Pierwszą rzeczą, którą możesz zauważyć, jest jaśniejszy obraz mapy. Ta jasność jest spowodowana naszym oświetleniem i domyślnym materiałem w samolocie. Na szczęście dla nas ten styl wizualny jest tym, czego szukamy, i pozostawimy dodatkową jasność bez zmian. Po drugie, zauważysz, że mapa jest bardziej rozpixselowana niż obraz, który wyrenderowaliśmy na serwerze w poprzednim kreatorze stylów. Ta pikselacja jest wynikiem rozciągnięcia naszego obrazu mapy w płaszczyźnie. Oczywistym rozwiązaniem jest zwiększenie rozmiaru i rozdzielczości obrazu. Niestety maksymalny rozmiar obrazu, którego możemy zażądać z Google Maps, wynosi około 1200 x 1200 pikseli, co już robimy przy podwójnej rozdzielczości. Oznacza to, że musimy znaleźć inne rozwiązanie, aby uzyskać czystsza i wyraźniejszą mapę. W następnym sekcji rozwiążemy ten problem z pikselacją.

Układanie kafelek

Ze względu na poziom szczegółowości map i drobne linie, generalnie zawsze chcemy renderować nasze mapy w najwyższej możliwej rozdzielczości. Niestety renderowanie obrazów w wysokiej rozdzielczości wymaga dużej wydajności i jest podatne na błędy. Na szczęście istnieje wiele przykładów tego, jak inni rozwiązali ten problem z rozdzielczością w mapowaniu, łącząc ze sobą wiele obrazów lub kafełków obrazów. Przyjmijmy dokładnie to samo podejście i rozszerzymy naszą mapę z pojedynczego kafełka do siatki 3 x 3 kafełków w następujący sposób:



Układ kafełków mapy dla siatki 3 x 3

Zauważ, że na diagramie oś x i przesunięcia kafełków są odwrotnie powiązane, czyli innymi słowy przesunięcie kafełka o 1 w kierunku X będzie musiało być przesunięte na osi x w przestrzeni 3D w kierunku ujemnym. Oś z i przesunięcie kafełka Y są w bezpośrednim związku. Oznacza to, że ponieważ przesunięcie kafełka Y jest ustawione na 1, oś z również otrzyma wartość dodatnią. Ponieważ nasz gracz jest blisko ziemi, gra potrzebuje tylko siatki 3x3. Jeśli zbudujesz grę z kamerą wyższego poziomu lub chcesz pokazać ją dalej na horyzoncie, rozszerzysz układ kafełków do 5 x 5, 7 x 7, 9 x 9 lub dowolnego rozmiaru. Zaczniemy więc i rozszerzymy naszą mapę z jednego kafełka do kafełków siatki 3x3. Postępuj zgodnie z tymi instrukcjami w Unity, aby zbudować układ kafełków mapy:

1. Wybierz obiekt gry Map_Tile w oknie Hierarchia. W oknie Inspektora edytuj właściwości obiektu, aby dopasować te wartości:

* Transform, Scale, X: 3

* Transform, Scale, Z: 3

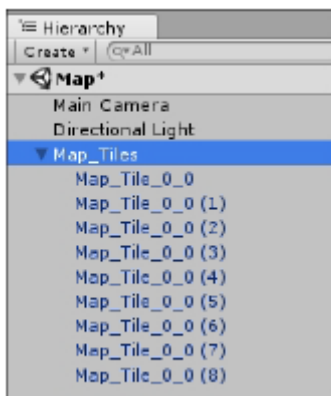
* Google Map Tile, Zoom Level: 17

2. Utwórz nowy folder w folderze FoodyGO, klikając prawym przyciskiem myszy (polecenie Naciśnij i kliknij Mac), aby otworzyć menu kontekstowe i wybierz pozycję Utwórz | Nowy folder. Zmień nazwę folderu w podświetlonym oknie edycji na Prefabs.

3. Stworzymy teraz prefabrykat naszego obiektu Map_Tile. Możesz myśleć o prefabrykacie jako kopiach obiektów gry lub szablonach. Aby stworzyć prefabrykaty, wybierz i przeciągnij obiekt gry Map_Tile do nowego folderu Prefabs, który właśnie utworzyliśmy. Zobaczysz, że nowy prefabrykat pojawi się w folderze o nazwie Map_Tile. Po utworzeniu prefabrykatu zauważysz, że obiekt gry Map_Tile zmienia kolor na niebieski w oknie Hierarchii. Niebieskie podświetlenie oznacza, że obiekt gry jest powiązany z prefabrykatem.

4. Wróć do okna Hierarchia i ponownie wybierz obiekt Map_Tile i zmień jego nazwę na Map_Tile_0_0. Zrobiliśmy to, aby oznaczyć to jako środek lub kafelki 0,0.

5. Wybierz obiekt gry Map_Tile_0_0 w oknie Hierarchii i wprowadź Ctrl + D (polecenie + D na Macu), aby zduplikować kafelki mapy. Zrób to osiem razy, aby utworzyć osiem dodatkowych kafelków mapy w następujący sposób:



Skopiowane kafelki mapy pokazane jako rodzic obiektu gry Map_Tiles

6. Zmień nazwy skopiowanych kafelków mapy i ustaw właściwości każdego z nich w oknie Inspektora, jak pokazano poniżej:

Obiekt gry : Właściwości

Map_Tile_0_0 (1) :

Name: Map_Tile_0_1

Transform.Position.X: 0

Transform.Position.Z: 30

GoogleMapTile.TileOffset.X: 0

GoogleMapTile.TileOffset.Y: 1

Map_Tile_0_0 (2) :

Name: Map_Tile_0_-1

Transform.Position.X: 0

Transform.Position.Z: -30

GoogleMapTile.TileOffset.X: 0

GoogleMapTile.TileOffset.Y: -1

Map_Tile_0_0 (3) :

Name: Map_Tile_1_0

Transform.Position.X: -30

Transform.Position.Z: 0

GoogleMapTile.TileOffset.X: 1

GoogleMapTile.TileOffset.Y: 0

Map_Tile_0_0 (4) :

Name: Map_Tile_-1_0

Transform.Position.X: 30

Transform.Position.Z: 0

GoogleMapTile.TileOffset.X: -1

GoogleMapTile.TileOffset.Y: 0

Map_Tile_0_0 (5) :

Name: Map_Tile_1_1

Transform.Position.X: -30

Transform.Position.Z: 30

GoogleMapTile.TileOffset.X: 1

GoogleMapTile.TileOffset.Y: 1

Map_Tile_0_0 (6) :

Name: Map_Tile_-1_-1

Transform.Position.X: 30

Transform.Position.Z: -30

GoogleMapTile.TileOffset.X: -1

GoogleMapTile.TileOffset.Y: -1

Map_Tile_0_0 (7) :

Name: Map_Tile_-1_1

Transform.Position.X: 30

Transform.Position.Z: 30

GoogleMapTile.TileOffset.X: -1

GoogleMapTile.TileOffset.Y: 1

Map_Tile_0_0 (8) :

Name: Map_Tile_1_-1

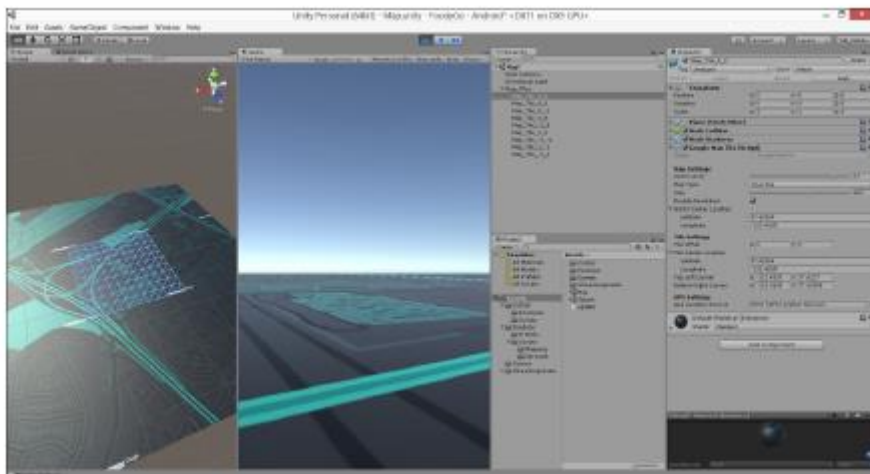
Transform.Position.X: -30

Transform.Position.Z: -30

GoogleMapTile.TileOffset.X: 1

GoogleMapTile.TileOffset.Y: -1

7. Naciśnij przycisk odtwarzania, aby uruchomić grę. Gdy gra jest uruchomiona, wybierz Map_Tile_0_0 w oknie Hierarchy i wpisz F , aby wykadrować obiekt w oknie Scena. Zwróć uwagę, jak znacznie zmniejszono piksele na mapie. Powinieneś zobaczyć coś podobnego do poniższego zrzutu ekranu:



Zrozumienie kodu

Świetnie, mamy teraz fajnie wyglądającą mapę w naszej grze. Oczywiście proces tworzenia mapy był nieco powtarzalny; ale jeśli jesteś ostrożny, nie powinno to trwać zbyt długo. Jeśli spodziewałeś się dużo matematyki, aby wyrównać te kafelki, na szczęście wszystko to odbywa się w skrypcie GoogleMapTile. Skorzystajmy z okazji, aby zrobić sobie przerwę od Unity i spojrzeć na skrypt GoogleMapTile w MonoDevelop. W Unity wybierz Map_Tile_0_0 w oknie Hierarchy. Przejdź do okna Inspektora i w komponencie skryptu Google Map Tile kliknij ikonę koła zębatego, aby otworzyć menu kontekstowe. Z menu wybierz Edytuj skrypt. Zobaczysz pasek postępu, a po kilku sekundach otworzy się MonoDevelop.

Wskazówka: MonoDevelop jest domyślnym edytorem skryptów dla Unity. Jeśli programujesz w systemie Windows, społeczność Visual Studio lub nowsza jest również doskonałą opcją. Inną dobrą opcją jest Visual Studio Code, który jest lekką alternatywą dla systemów Windows, Mac i Linux.

W MonoDevelop zobaczysz otwarty skrypt GoogleMapTile. Jak wspomniano we wstępnych wymaganiach książki, powinieneś mieć podstawową wiedzę na temat C#, więc zawartość skryptu nie powinna wyglądać zbyt onieśmielająco. Jeśli jesteś nowy w pisaniu skryptów Unity, to w porządku, ponieważ później omówimy więcej szczegółów na temat pisania skryptów. Na razie skoncentrujemy się na kilku obszarach kodu, które pokażą, jak działa kafelkowanie mapy. Przewiń kod w dół, aż dojdiesz do metody IEnumerator _RefreshMapTile(). Oto fragment najważniejszych wierszy tej metody, którym przyjrzymy się bardziej szczegółowo:

```
IEnumerator _RefreshMapTile ()
{
//find the center lat/long of the tile
tileCenterLocation.Latitude =
GoogleMapUtils.adjustLatByPixels(worldCenterLocation.Latitude, (int)(size * 1 * TileOffset.y),
zoomLevel);
tileCenterLocation.Longitude =
GoogleMapUtils.adjustLonByPixels(worldCenterLocation.Longitude, (int)(size * 1 * TileOffset.x),
zoomLevel);
```

Jak wspomniano w komentarzu, te dwa wiersze kodu znajdują środek kafelka we współrzędnych mapy szerokości i długości geograficznej. Robią to, biorąc rozmiar obrazu kafelka (rozmiar) i mnożąc go przez TileOffset.y dla szerokości geograficznej i TileOffset.x dla długości geograficznej. Wynik tego mnożenia i zoomLevel są przekazywane do funkcji pomocniczych GoogleMapUtils w celu obliczenia dostosowanej szerokości lub długości geograficznej kafelka. Wydaje się proste, prawda? Oczywiście większość pracy jest wykonywana w funkcjach GoogleMapUtils, które są standardowymi funkcjami matematycznymi GIS do konwersji odległości. Jeśli jesteś ciekawy, spójrz na kod GoogleMapUtils, ale na razie będziemy przyglądać się tylko metodzie _RefreshMapTile. Kontynuuj przewijanie kodu, aż dojdiesz do tego fragmentu:

```
//build the query string parameters for the map
tile request
queryString += "center=" + WWW.UnEscapeURL
(string.Format ("{0},{1}",
tileCenterLocation.Latitude,
tileCenterLocation.Longitude));
queryString += "&zoom=" + zoomLevel.ToString ();
queryString += "&size=" + WWW.UnEscapeURL
(string.Format ("{0}x{0}", size)); queryString +=
"&scale=" + (doubleResolution ? "2" : "1");
```



```

queryString += "&maptype=" + mapType.ToString
().ToLower ();
queryString += "&format=" + "png";
//adding the map styles
queryString +=
"&style=element:geometry|invert_lightness:true|weight:3.1|hue:0x00ffd5";
queryString +=
"&style=element:labels|visibility:off";

```

Jak opisano w komentarzu, ta sekcja kodu tworzy parametry zapytania, które są przekazywane do interfejsu API Map Google w celu zażądania obrazu mapy. Ponieważ przekazujemy te parametry w adresie URL, musimy upewnić się, że kodujesz znaki specjalne i to właśnie robią wywołania `WWW.UnEscapeURL`. Zauważ, że na dole dodajemy również kilka stylów. W części 9, Zakończenie gry, przyjrzymy się, jak łatwo dodać własne style za pomocą Kreatora stylów Map Google. Na koniec przewiń do dołu metody `_RefreshMapTile`; poniżej znajduje się fragment kodu:

```

//finally, we request the image
var req = new WWW(GOOGLE_MAPS_URL + "?" +
queryString);
//yield until the service responds
yield return req;
//first destroy the old texture first
Destroy(GetComponent< Renderer >
().material.mainTexture);
//check for errors
if (req.error != null)
{
print(string.Format("Error loading tile
{0}x{1}: exception={2}",
TileOffset.x, TileOffset.y,
req.error));
}
else
{
//no errors render the image

```

```

//when the image returns set it as the tile
texture
GetComponent< Renderer >
().material.mainTexture = req.texture;
print(string.Format("Tile {0}x{1}
textured", TileOffset.x, TileOffset.y));
}

```

W pierwszym wierszu kod wykorzystuje klasę WWW do wykonania żądania do GOOGLE_MAPS_URL dołączonego do wcześniej skonstruowanego queryString. Klasa WWW jest klasą pomocniczą Unity, która pozwala nam wywoływać adresy URL dla praktycznie wszystkiego. W dalszej części książki użyjemy tej klasy do wysłania innych zgłoszeń serwisowych. Następną linią, yield return req;, zasadniczo mówi Unity, aby kontynuowała, dopóki to żądanie nie odpowie. Możemy to zrobić tutaj, ponieważ ta metoda jest współprogramowa. Współprogramy to metody, które zwracają IEnumerator i są eleganckim sposobem zapobiegania blokowaniu wątków. Jeśli kiedykolwiek robiłeś bardziej tradycyjne programowanie asynchroniczne w C#, z pewnością docenisz piękno współprogramów. Tak jak poprzednio, omówimy więcej szczegółów na temat współprogramów, gdy przejdziemy do pisania skryptów. Następnie wywołujemy Destroy na bieżącej teksturze obiektu. Destroy to publiczna metoda klasy MonoBehaviour, która w bezpieczny sposób pozwala nam niszczyć obiekty i wszystkie komponenty dołączone do obiektu. Jeśli jesteś doświadczonym programistą C# Windows lub Web, ten krok może wydawać się dość obce dla Ciebie. Pamiętaj tylko, że musimy uważać na zarządzanie pamięcią, które może szybko wymknąć się spod kontroli podczas uruchamiania gry. W tym przykładzie, gdybyśmy usunęli ten wiersz kodu, gra prawdopodobnie uległaby awarii z powodu wycieku pamięci tekstur. Po wywołaniu Destroy sprawdzamy błędy, aby upewnić się, że podczas żądania kafelka obrazu nie wystąpiły żadne błędy. Jeśli wystąpi błąd, po prostu wyświetlamy komunikat o błędzie. W przeciwnym razie zamieniamy bieżącą teksturę na nowy pobrany obraz. Następnie używamy print, aby napisać komunikat debugowania w oknie konsoli. Metoda print jest taka sama jak wywołanie Debug.log, ale jest dostępna tylko z klasy pochodnej MonoBehaviour. Nasze ostateczne spojrzenie na kod będzie polegało na zrozumieniu, kiedy wywoływana jest metoda _RefreshMapTile. Przewiń kod w górę, aż znajdziesz metodę Update w następujący sposób:

```

// Update is called once per frame
void Update ()
{
//check if a new location has been
acquired
if (gpsLocationService != null &&
gpsLocationService.IsServiceStarted &&
lastGPSUpdate <
gpsLocationService.Timestamp)
{

```

```

lastGPSUpdate =
gpsLocationService.Timestamp;
worldCenterLocation.Latitude =
gpsLocationService.Latitude;
worldCenterLocation.Longitude =
gpsLocationService.Longitude;
print("GoogleMapTile refreshing
map texture");
RefreshMapTile();
}
}

```

Update to specjalna metoda Unity dostępna w każdej klasie pochodnej MonoBehaviour. Jak wspomniano w komentarzu, metoda Update nazywana jest każdą klatką. Oczywiście nie chcemy odświeżać kafelka mapy co klatkę, ponieważ jest mało prawdopodobne, że żądanie tak szybko zwróci w każdym razie. Zamiast tego chcielibyśmy najpierw upewnić się, że korzystamy z usługi lokalizacyjnej i to się zaczęło. Następnie sprawdzamy, czy usługa lokalizacji wykryła ruch, sprawdzając zmienną timestamp. Jeśli przejdzie te trzy testy, aktualizujemy znacznik czasu, uzyskujemy nowe centrum świata, drukujemy wiadomość i na koniec wywołujemy RefreshMapTile. RefreshMapTile wykonuje wywołanie StartCoroutine(_RefreshMapTile), które rozpoczyna odświeżanie kafelka. Ponieważ nie zaczęliśmy jeszcze podłączać usługi GPS, to wszystko może wydawać się obce. Nie martw się, wkrótce do tego dojdziemy, ale na razie pomocne będzie zrozumienie, jak często będą przerysowywane kafelki mapy. W tej sekcji poprawiliśmy rozdzielczość naszej mapy gry, renderując kafelki obrazów zamiast pojedynczego obrazu. W naszym celu nadal używaliśmy dość dużego rozmiaru kafelka dla każdego obrazu kafelka mapy. Uchodzi nam to na sucho, ponieważ nasza kamera będzie znajdować się nad graczem patrzącym w dół. Jak jednak widać, tworzenie mapy kafelkowej dowolnego rozmiaru jest bardzo proste. Jeśli zdecydujesz się stworzyć większą mapę, pamiętaj tylko, że pobranie kilku kafelków mapy może znacznie zwiększyć zużycie danych przez gracza.

Konfigurowanie usług

Usługi mogą mieć szeroką definicję w zależności od aplikacji i potrzeb. Dla nas będziemy używać terminu usługi do oznaczenia dowolnego kodu, który działa jako klasa samoorganizująca, z której korzystają inne obiekty gry. Usługi różnią się od biblioteki lub globalnej klasy statycznej, takiej jak klasa GoogleMapUtils, ponieważ działają jako obiekt lub obiekty. W niektórych przypadkach możesz zdecydować się na wdrożenie usług przy użyciu wzorca singleton. Jeśli chodzi o tę książkę, naszym zamiarem jest napisanie prostszego kodu, więc będziemy tworzyć i wykorzystywać usługi jako obiekty gry. W tej części skonfigurujemy dwie usługi. Usługa lokalizacji GPS do znajdowania lokalizacji gracza i CUDLR do debugowania. Zaczniemy od uruchomienia usługi CUDLR, ponieważ pomoże nam to w debugowaniu wszelkich problemów, które możemy mieć podczas konfigurowania usługi lokalizacji.

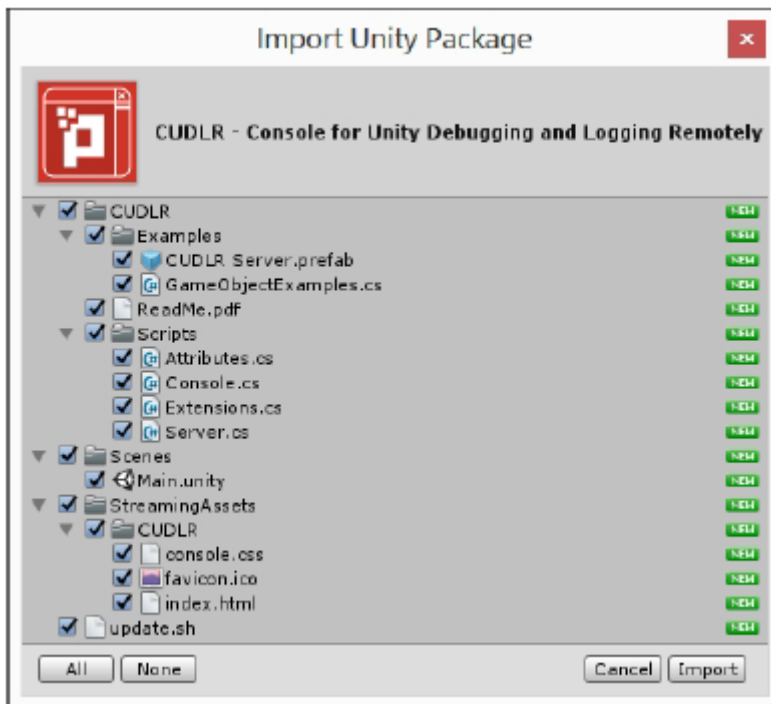
Konfigurowanie CUDLR

CUDLR to skrót od Console for Unity Debugging and Logging Remotely i jest darmowym zasobem dostępnym w Unity Asset Store. Użyjemy CUDLR, aby nie tylko obserwować aktywność naszego urządzenia podczas gry, ale także zdalnie wykonywać kilka prostych poleceń konsoli. W następnej części przyjrzymy się Unity Remote, które jest kolejnym narzędziem diagnostycznym. Jest bardzo potężny, ale może być problematyczny w uruchomieniu i często nie ma dostępu do usługi lokalizacji, mimo że Unity twierdzi, że jest to obsługiwane. Gdy zagłębisz się w rozwój gry, zobaczysz, że zawsze przydatna jest zdalną metodę monitorowania i kontrolowania naszej gry.

Wskazówka: Aby korzystać z CUDLR, urządzenie i komputer programistyczny muszą znajdować się w tej samej lokalnej sieci Wi-Fi. Pomiń tę sekcję, jeśli urządzenie mobilne nie może połączyć się z lokalną siecią Wi-Fi.

Wykonaj następujące kroki, aby zainstalować i skonfigurować CUDLR:

1. Otwórz okno magazynu zasobów, wybierając pozycję menu Okno | Sklep aktywów. Po otwarciu okna wpisz cudlr w polu wyszukiwania i naciśnij Enter. Po kilku sekundach powinna pojawić się lista zasobów.
2. Kliknij obraz lub link do CUDLR, aby załadować stronę zasobów. Po załadowaniu strony zobaczysz przycisk Importuj; kliknij ten przycisk, aby zaimportować zasób do projektu.
3. Zasób jest dość mały, więc powinien się szybko pobrać. Po zakończeniu pobierania zobaczysz okno dialogowe Importuj pakiet Unity. Tylko upewnij się, że wszystko jest zaznaczone, jak pokazano na poniższym zrzucie ekranu:



4. Ponieważ po raz pierwszy importujemy ten zasób do projektu, zainstalujemy wszystko. Na późniejszym etapie rozwoju zawsze możemy zdecydować o usunięciu niepotrzebnych części zasobu, takich jak folder Przykłady. Kiedy będziesz gotowy do importu, kliknij przycisk Importuj w oknie dialogowym i poczekaj na zakończenie importu.

5. Utwórz obiekt usług nadrzędnych w scenie, wybierając pozycję menu GameObject | Utwórz puste. Spowoduje to utworzenie nowego pustego obiektu gry w oknie hierarchii. Zmień nazwę nowego obiektu Usługi.

6. Kliknij prawym przyciskiem myszy (naciśnij klawisz Ctrl i kliknij komputer Mac) obiekt Usługi w oknie hierarchii, aby otworzyć okno kontekstowe. Z menu kontekstowego wybierz Utwórz puste. Spowoduje to utworzenie pustego obiektu o nazwie GameObject jako elementu podrzędnego obiektu Services. Powtórz ten proces, aby utworzyć kolejny pusty obiekt gry.

7. Wybierz pierwszy pusty GameObject i zmień jego nazwę na CUDLR. Następnie wybierz drugi i zmień jego nazwę na GPS. Później dodamy usługę GPS, ale skoro już tu jesteśmy, bądźmy wydajni.

8. Otwórz folder Assets/CUDLR/Scripts w oknie projektu. Wybierz i przeciągnij skrypt serwera z okna projektu do obiektu gry CUDLR w oknie hierarchii. Spowoduje to dodanie składnika serwera CUDLR do obiektu gry. To wszystko, CUDLR jest gotowy do pracy.

Debugowanie za pomocą CUDLR

To, co sprawia, że CUDLR jest tak użytecznym narzędziem, to fakt, że zamienia on część naszej gry w serwer sieciowy; tak, serwer WWW. Możemy wtedy przeglądać i komunikować się z naszą grą, tak jakbyśmy zainstalowali backdoora. Ponieważ CUDLR jest dostępny dla każdego komputera w sieci, nie musimy mieć fizycznego połączenia ani nawet uruchamiać Unity, aby sterować grą. Oczywiście posiadanie gry jako lokalnego serwera WWW stanowi zagrożenie dla bezpieczeństwa gry i prawdopodobnie urządzenia gracza. Dlatego przed wysłaniem gry po prostu usuń obiekt gry usługi CUDLR, aby go dezaktywować. Postępuj zgodnie z instrukcjami tutaj, aby połączyć się z usługą CUDLR uruchomioną w grze:

1. Otwórz swoje urządzenie mobilne i znajdź adres IP, zapisz go lub zapamiętaj. Wykonaj następujące czynności, aby znaleźć adres IP dla systemu Android lub iOS: Android: Przejdź do Ustawienia | O telefonie | Status i przewiń w dół do pola adresu IP. iOS: Wybierz WiFi, aby otworzyć listę aktualnie dostępnych sieci bezprzewodowych. Znajdź na liście sieć bezprzewodową, z którą jesteś aktualnie połączony, a następnie dotknij niebieskiego kółka z białą strzałką, która pojawia się po prawej stronie nazwy sieci. Pole Adres IP powinno być pierwszym polem na liście pod zakładką DHCP.

2. Zbuduj i wdróż grę na swoje urządzenie mobilne, wykonując czynności opisane wcześniej w tym rozdziale. Upewnij się, że gra jest otwarta i uruchomiona na Twoim urządzeniu.

3. Odłącz kabel USB od urządzenia i komputera.

4. Otwórz przeglądarkę internetową; Chrome to dobry wybór. Wpisz następujący adres URL w pasku adresu: `http://[Adres IP urządzenia]:55055/`.

5. Powinieneś zobaczyć okno konsoli CUDLR w przeglądarce, z wyjściem bardzo podobnym do tego zrzutu ekranu:



6. Podczas ładowania mapy kafelków w oknie powinno być rejestrowanych dziewięć wywołań żądań i dziewięć wywołań odpowiedzi. Mam nadzieję, że do tej pory możesz trochę przeczytać i zrozumieć, co robią te prośby.

7. Co jeszcze może zrobić konsola? Wpisz help w dolnym polu tekstowym poniżej danych wyjściowych konsoli. Spowoduje to wyświetlenie listy wszystkich dostępnych poleceń. Obecnie jest tylko kilka poleceń, ale dodamy je później. Jeśli z jakiegoś powodu konsola nie odpowiada, spróbuj odświeżyć stronę i/lub upewnij się, że gra jest uruchomiona na Twoim urządzeniu mobilnym.

Jeśli masz problemy z uruchomieniem lub połączeniem z CUDLR, zapoznaj się z częścią 10, Rozwiązywanie problemów. Jak wspomnieliśmy wcześniej, przyjrzymy się innym opcjom debugowania i diagnozowania problemów podczas programowania w Unity. Jednak CUDLR, który może działać całkowicie zdalnie, będzie naszą najlepszą opcją do testowania naszej gry, ponieważ testujemy ruch w świecie rzeczywistym i śledzenie GPS. A propos GPS, czas zakończyć ostatnią część i zebrać wszystko razem.

Konfigurowanie usługi GPS

Ostatnim elementem, którego potrzebujemy do wygenerowania mapy opartej na lokalizacji w świecie rzeczywistym, jest znalezienie miejsca, w którym znajduje się urządzenie. Oczywiście najlepszym sposobem, jak dowiedzieliśmy się w poprzednim rozdziale, jest wykorzystanie wbudowanego GPS do określenia, gdzie znajduje się urządzenie we współrzędnych szerokości i długości geograficznej. Podobnie jak w przypadku mapy kafelkowej, użyjemy zaimportowanego skryptu do zbudowania usługi i szybkiego uruchomienia bez wchodzenia w nic

skrypty. Zanim zaczniemy, upewnij się, że Twoje urządzenie ma włączoną usługę lokalizacji, sprawdzając następujące elementy:

* Android: przejdź do Ustawień | Lokalizacja i potwierdź, że usługa jest włączona.

* iOS: postępuj zgodnie z tymi instrukcjami:

1. Stuknij w Prywatność | Usługi lokalizacyjne.
2. Przewiń w dół i dotknij FoodyGO.
3. Zdecyduj, czy zezwolić na dostęp do lokalizacji Nigdy lub Podczas korzystania z aplikacji.

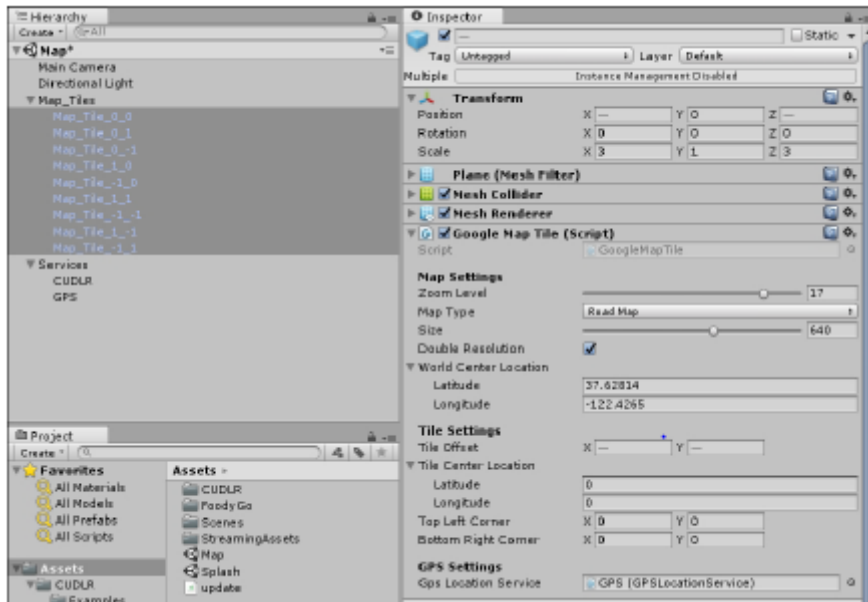
Teraz wykonaj następujące instrukcje, aby zainstalować kod usługi GPS i przetestować grę:

1. Wybierz obiekt gry GPS w sekcji Usługi w oknie Hierarchia.
2. Kliknij przycisk Dodaj składnik w oknie Inspektora,
i z listy komponentów wybierz Usługi | Usługa lokalizacji GPS.

3. Zobaczysz, że składnik `CLLocationService` zostanie dodany do obiektu usługi GPS.

4. Wybierz `Map_Tile_0_0` w oknie Hierarchy, a następnie naciśnij klawisz Shift i kliknij (tak samo w przypadku komputerów Mac) na dolnym kafelku mapy na liście, aby wybrać wszystkie dziewięć kafelków mapy razem.

5. Po zaznaczeniu wszystkich dziewięciu kafelków mapy przeciągnij obiekt usługi GPS w oknie Hierarchia do pola Usługa lokalizacji GPS w następujący sposób:



6. To, co właśnie zrobiliśmy, to edycja wszystkich dziewięciu kafelków mapy w tym samym czasie i dodanie do każdego z nich usługi `CLLocationService`. Pamiętaj, że w naszym badaniu skryptu `GoogleMapTile` zauważyliśmy, że kafelki mapy będą wywoływać usługę `CLLocationService` w celu znalezienia współrzędnych centrum świata mapy.

7. Teraz, gdy usługa GPS jest połączona ze wszystkimi kafelkami mapy, naciśnij Odtwórz i zobaczymy, jak to wygląda.

8. Jeśli drapiesz się po głowie zastanawiając się, co poszło nie tak, nie bój się. Właściwie nic nie poszło nie tak. Problem polega na tym, że edytor Unity nie ma dostępu do usługi lokalizacyjnej ani GPS podczas działania na twoim komputerze. To, co musimy zrobić, to wdrożyć grę na Twoje urządzenie.

9. Twórz, wdrażaj i uruchamiaj grę na swoim urządzeniu tak jak poprzednio. Powinieneś teraz czuć się komfortowo, uruchamiając grę.

10. Powinieneś teraz zobaczyć mapę okolicy wokół ciebie. Mapa prawdopodobnie będzie wyglądać na przesuniętą, a to ze względu na położenie kamery. Nie martw się, naprawimy to w następnym rozdziale. Jeśli masz problemy z wyświetleniem innej mapy, po prostu sprawdź każdy z powyższych kroków. Dla tych, którzy nadal mają problemy nawet po potwierdzeniu poprzednich kroków, zapoznaj się z rozdziałem 10, Rozwiązywanie problemów.

11. Upewnij się, że wróciłeś do przeglądarki i odświeżysz stronę konsoli CUDLR. Zwróć uwagę na wszystkie wyjścia konsoli, które teraz widzimy. Zwróć szczególną uwagę na żądania dotyczące kafelków mapy; te współrzędne centrum powinny teraz odpowiadać współrzędnym lokalizacji.

12. Odłącz urządzenie od komputera i chodź po domu lub posesji. Nie idź tak daleko, aby odłączyć urządzenie od sieci, ale postaraj się, aby GPS zaktualizował lokalizację. Być może nawet poproś znajomego, który wędruje z urządzeniem mobilnym, podczas gdy Ty oglądasz konsolę CUDLR.

Baw się dobrze...

Mamy nadzieję, że ta ostatnia część rozdziału jest satysfakcjonującym zakończeniem bardzo szybkiego wprowadzenia do GIS, mapowania i GPS.

Podsumowanie

Zapoznałeś się z podstawowymi informacjami na temat GIS, mapowania i GPS. Ta podstawowa wiedza pomogła nam zdefiniować pewne terminy dotyczące pracy z Google Maps API i ładowania go w Unity. Następnie dodaliśmy mapę do gry, ale uznaliśmy, że brakuje jej jakości. To doprowadziło nas do zbudowania systemu map kafelkowych dla mapy gry. Następnie zrobiliśmy krótką przerwę, aby wprowadzić narzędzie do debugowania konsoli o nazwie CUDLR. CUDLR pomógł nam debugować fundamentalny element naszej gry i znaleźć lokalizację gracza za pomocą GPS. To pozwoliło nam zakończyć rozdział, dodając GPS do naszej gry za pomocą usługi lokalizacji GPS. Teraz, gdy mamy już ustalone podstawy, możemy zająć się bardziej praktycznym tworzeniem gier. Poniższy rozdział będzie błyskawicznym wprowadzeniem do dodawania w pełni sfałszowanej postaci do naszej sceny, a także obejmie mobilne wejście dotykowe, kamery o swobodnym wyglądzie i dostęp do czujników ruchu urządzenia.