

RAD w aplikacjach mobilnych i enterprise

C++ Builder – aplikacje dla systemu Android – interfejs użytkownika

opracował: dr inż. Andrzej Czajkowski, a.czajkowski@issi.uz.zgora.pl

1 Cel ćwiczenia

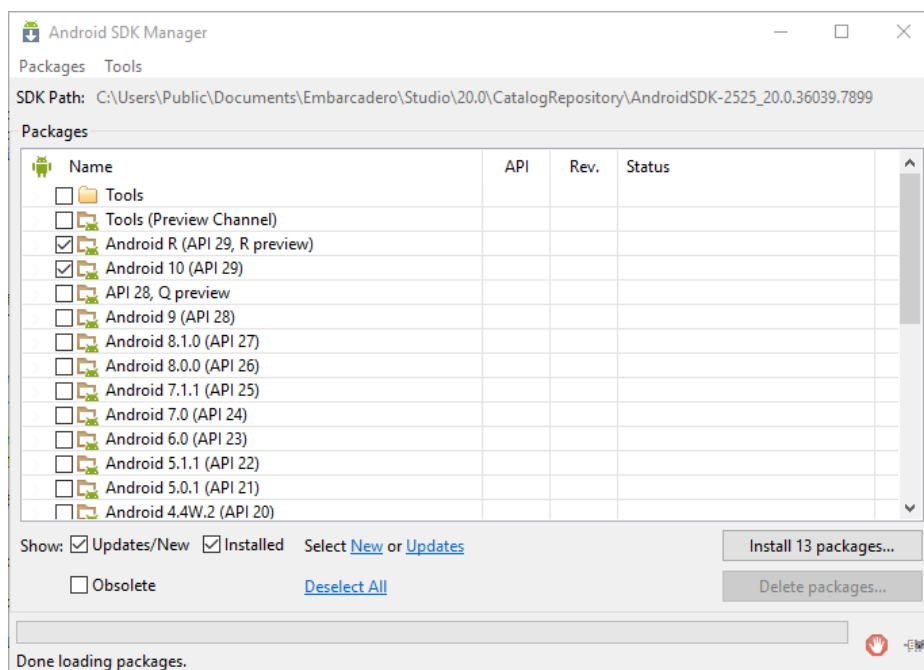
Celem ćwiczenia jest zapoznanie się z podstawową funkcjonalnością środowiska C++ Builder w kontekście tworzenia aplikacji mobilnych dla systemu Android. W ramach ćwiczenia studenci zapoznają się z procedurą tworzenia urządzeń wirtualnych (emulatorów urządzeń mobilnych), generowanie i deployowaniem plików apk oraz tworzeniem podstawowych interfejsów graficznych.

2 Przebieg zajęć

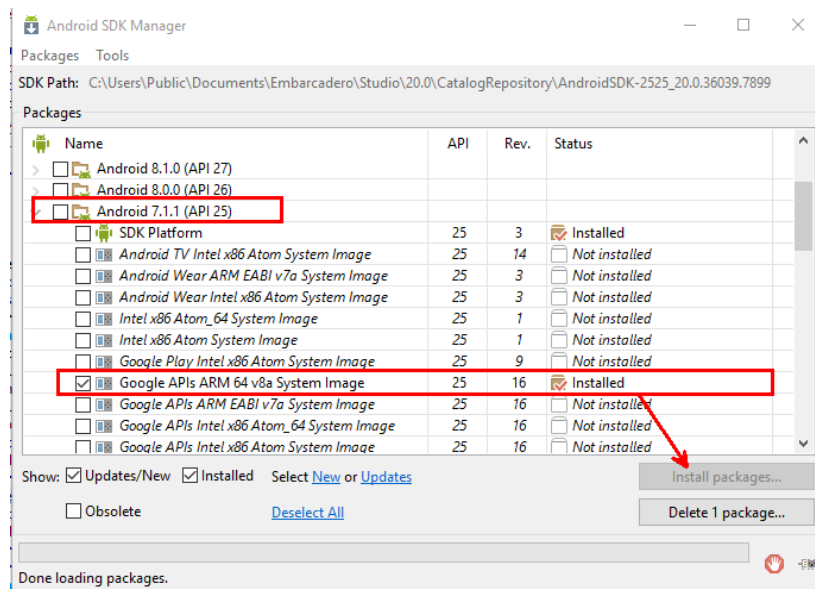
2.1 Emulator systemu Android

W celu testowania aplikacji niezwykle przydatnym jest narzędzie emulujące system Android (emulator). Z poziomu środowiska C++ Builder, możliwa jest automatyczna instalacja i uruchomienie tworzonych aplikacji na emulatorze.

1. W celu utworzenia emulatora należy uruchomić SDK Manager dla systemu Android (instalowany razem z C++ Builderem, domyślnie w ścieżce C:\Users\Public\Documents\Embarcadero\Studio\20.0\CatalogRepository\AndroidSDK):

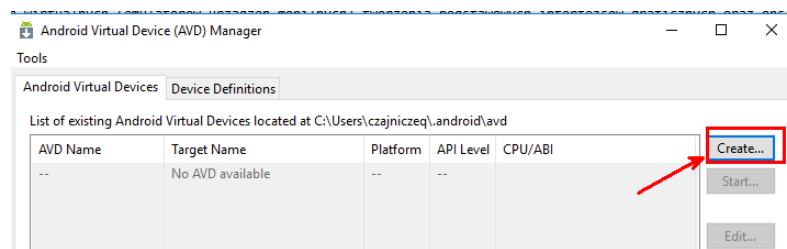


2. Następnie należy doinstalować obraz systemu, który będzie emulowany (zalecana wersja zgodna z cpu typu ARM):

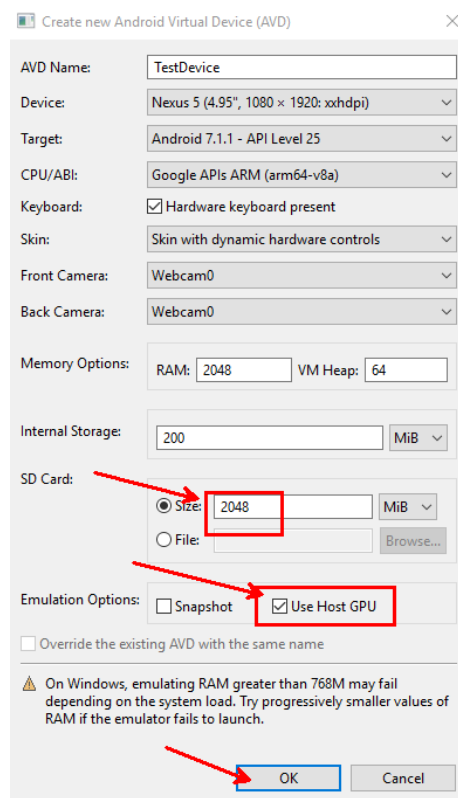


W przypadku posiadania procesora firmy Intel można spróbować obraz nowszej dystrybucji.

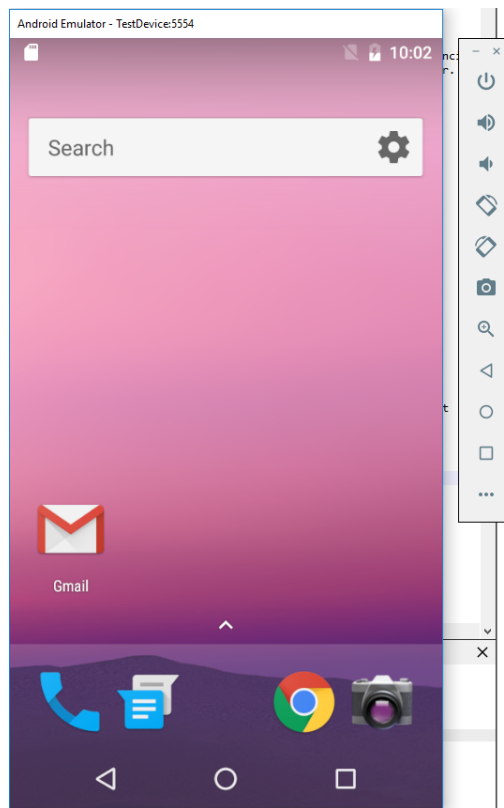
3. Następnie, należy uruchomić AVD manager i utworzyć nowe wirtualne urządzenie (Android Virtual Device znajduje się w tym samym folderze co SDK manager):



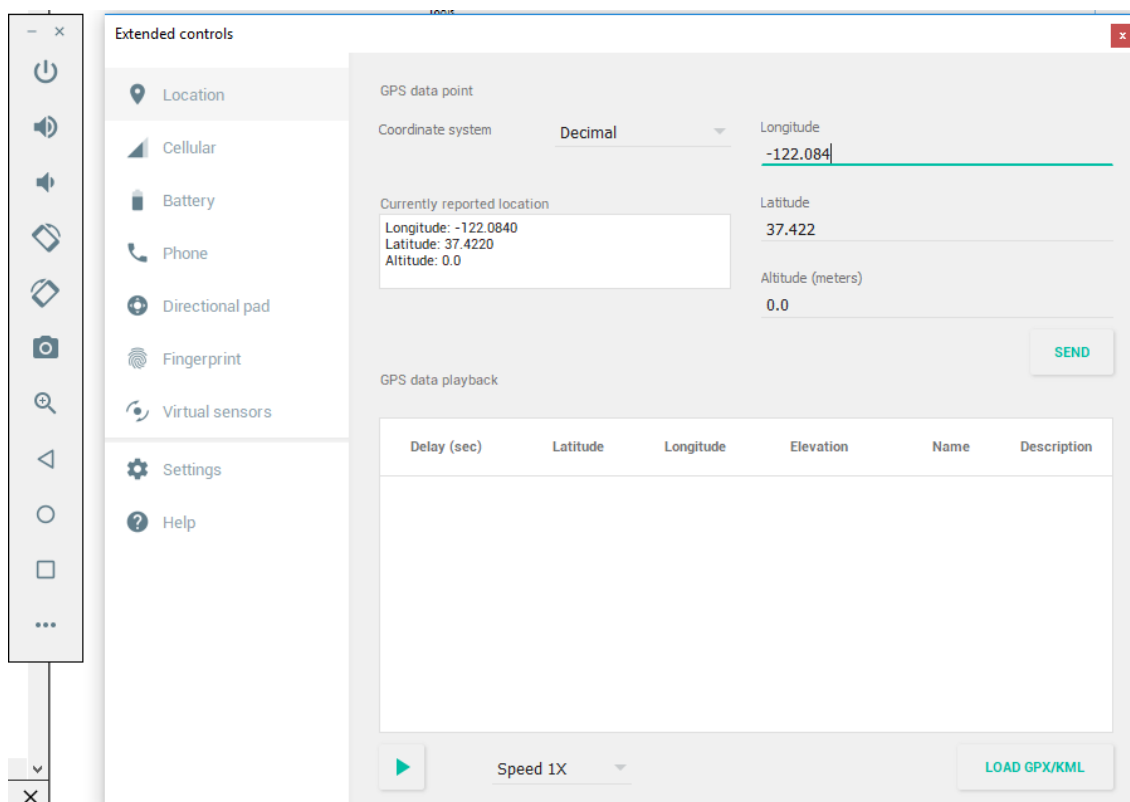
4. Konfigurację nowej maszyny ustawić jak na slajdzie (w przypadku niedostatecznej płynności działania emulatora można wybrać urządzenie o słabszych parametrach):



5. Urządzenie uruchomić opcją **Start** i dobrać skalowanie ekranu do własnego monitora. Proces uruchamiania maszyny może zająć kilka minut (bootowanie maszyny jest sygnalizowane pulsującym napisem ANDROID na urządzeniu wirtualnym). Po uruchomieniu możliwe jest korzystanie z emulatora jak z dowolnego urządzenia pod kontrolą systemu Android:



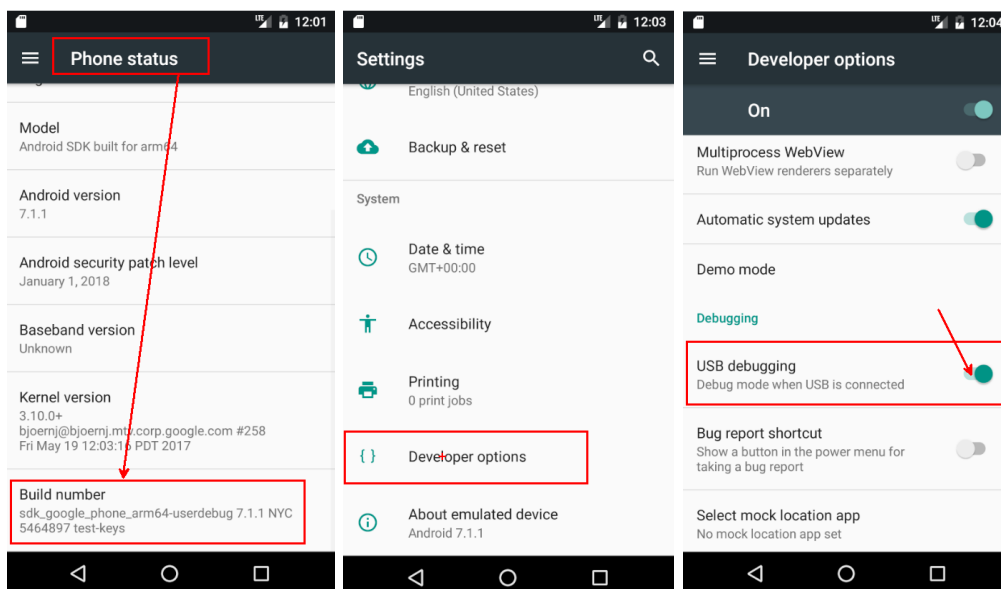
6. W narzędziach dostępnych z prawej strony emulatora można manipulować fizycznymi przyciskami urządzenia, jego orientacją i innymi zewnętrznymi funkcjonalnościami w celu symulowania różnych scenariuszy funkcjonowania:



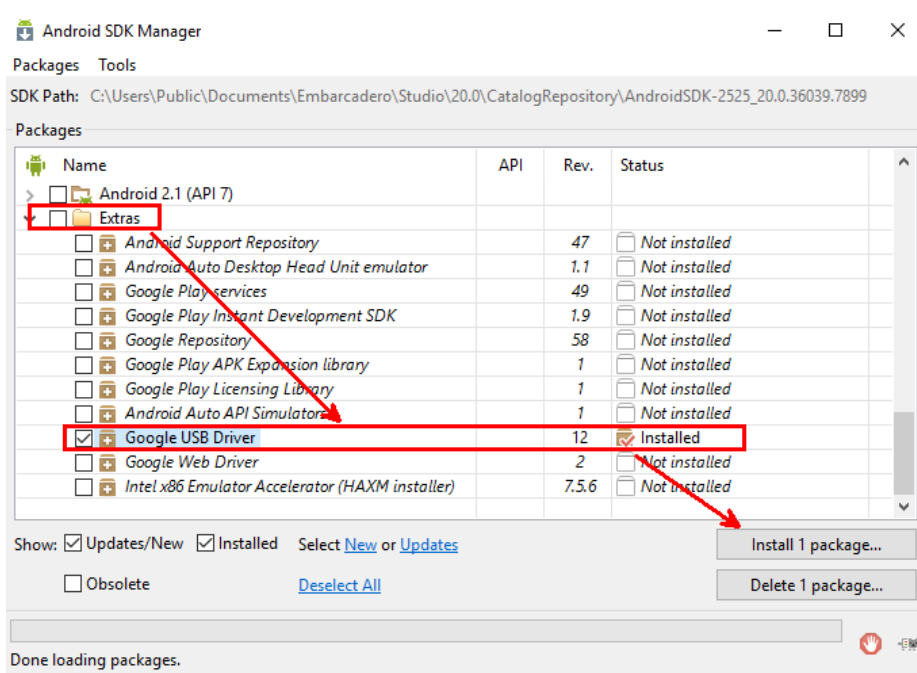
7. Wgrywanie i uruchamianie aplikacji zarówno dla emulatora jak i fizycznego urządzenia zostanie przedstawione w sekcji 2.4.

2.2 Podłączenie urządzenia zewnętrznego

1. Aby uzyskać możliwość wgrywania konieczne jest uruchomienie trybu "Debugowania USB" w urządzeniu. W tym celu należy mieć uruchomiony tryb programisty (w większości urządzeń jest on aktywowany poprzez 5-krotne naciśnięcie na numer kompilacji systemu w ustawieniach => informacje o oprogramowaniu):

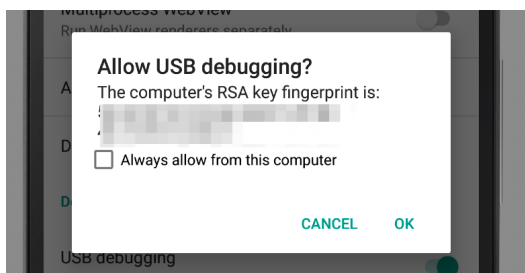


2. Następnie należy zainstalować sterowniki ADB dla swojego urządzenia. Większość urządzeń korzysta ze sterowników Googlem, dostępnych z SDK manager'a:



3. W przypadku gdy w kolejnych krokach nie będzie możliwe wgranie aplikacji, najprawdopodobniej dla danego urządzenia wymagane są sterowniki OEM, dostępne pod linkiem <https://developer.android.com/studio/run/oem-usb>

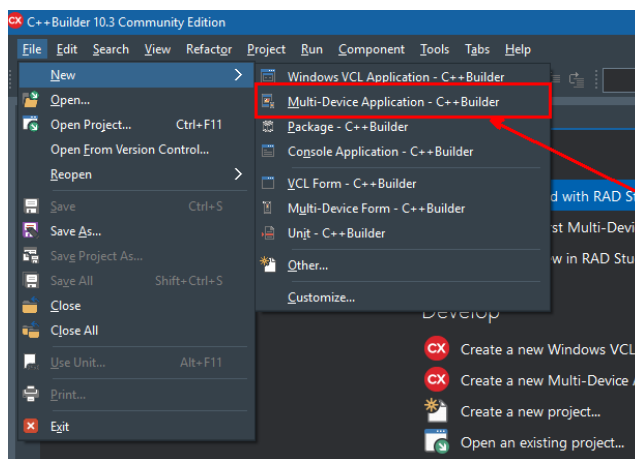
- Po instalacji sterowników i po podłączeniu urządzenia kablem USB, należy zaakceptować połączenie telefonu z komputerem:



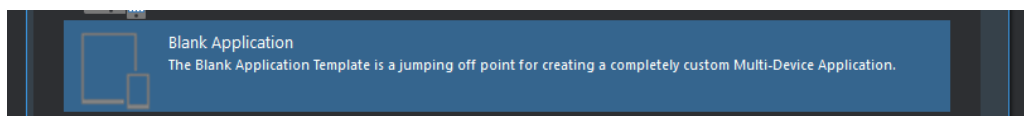
- Wgrywanie i uruchamianie aplikacji zarówno dla emulatora jak i fizycznego urządzenia zostanie przedstawione w sekcji 2.4.

2.3 Budowa interfejsu użytkownika w aplikacji mobilnej.

- Po uruchomieniu środowiska utworzyć nowy projekt jako Multi-Device Application - C++ Builder:

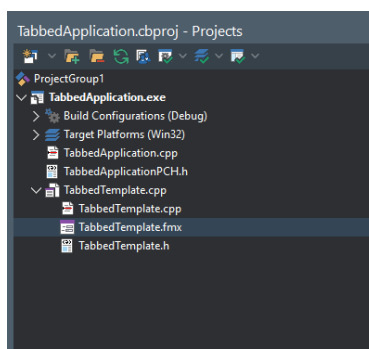


- Z dostępnych początkowych układów wybierz Blank Application:

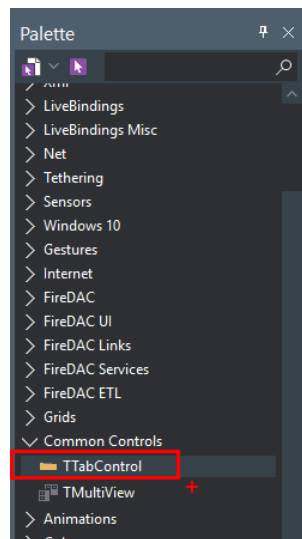


zapisz projekt w wybranej lokalizacji poprzez (Save all).

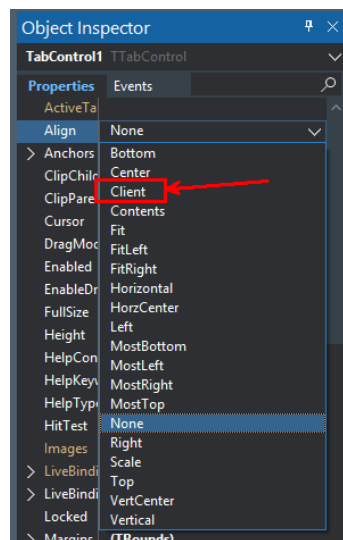
- Przeanalizuj okno plików projektu, zauważ, że po rozwinięciu pliku *.cpp, podwójne kliknięcia na plikach *.h, *.fmx i *.cpp umożliwia edycję różnych elementów aplikacji:



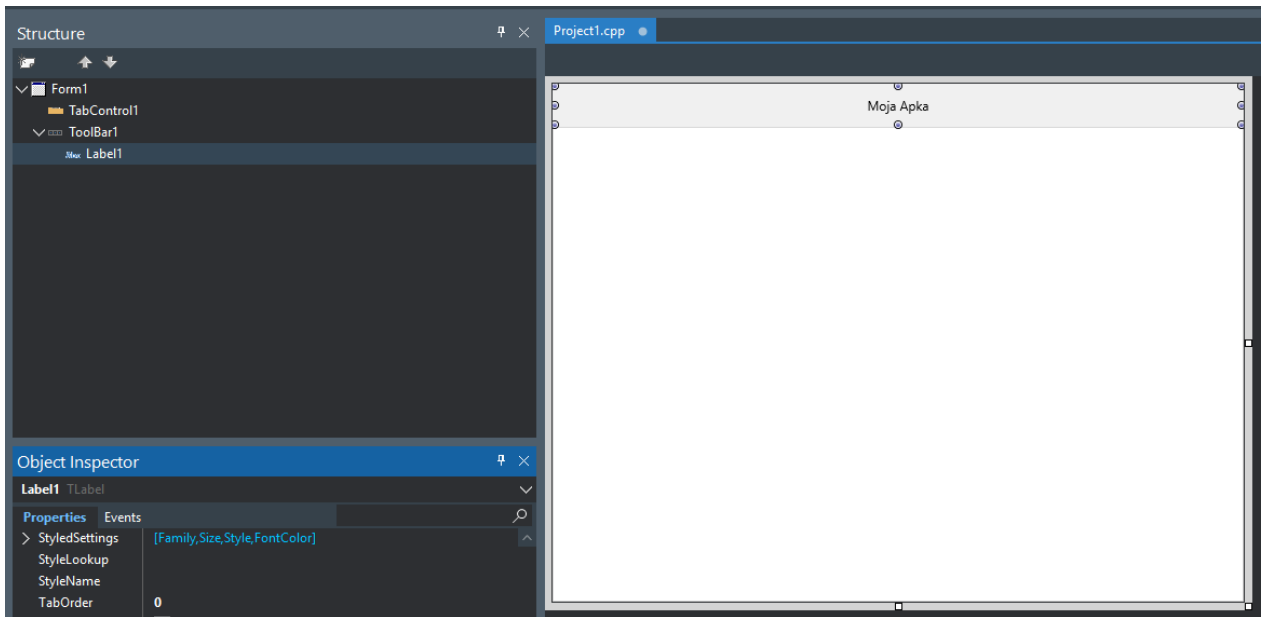
4. Większość aplikacji mobilnych jest realizowana z wykorzystaniem zakładek którymi manipuluje się z wykorzystaniem gestów. W celu utworzenia takiego interfejsu należy w pierwszej kolejności dodać do edytora komponent `TTabControl` poprzez podwójne kliknięcie:



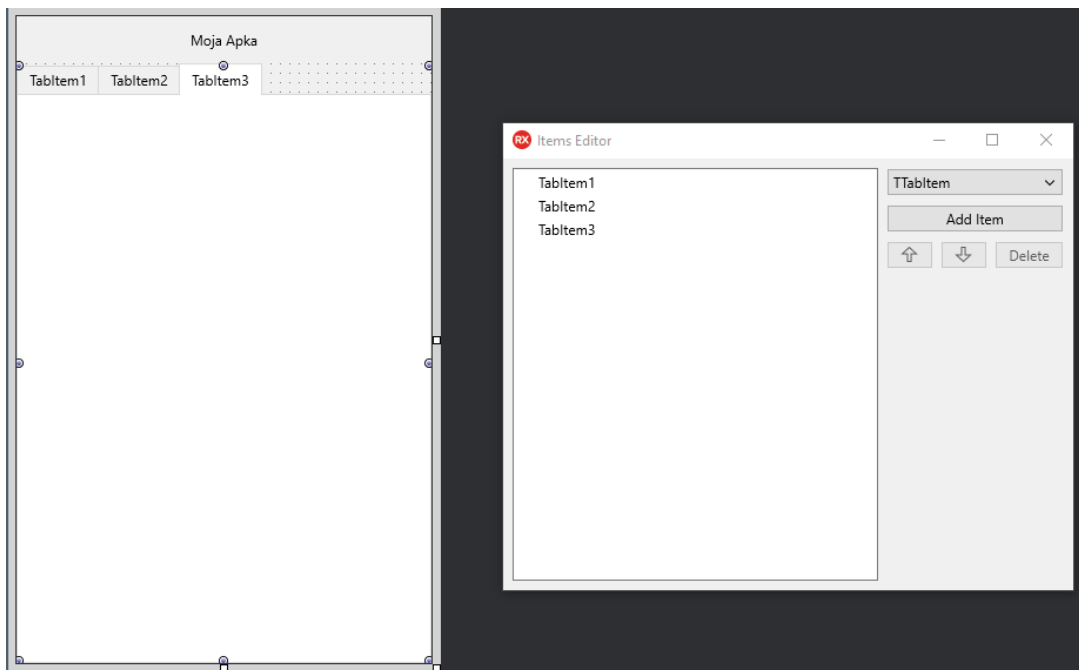
5. Aby kontroler zakładek wypełniał całą dostępną przestrzeń należy z `Object Inspector` w polu `Align` wybrać `Client`:



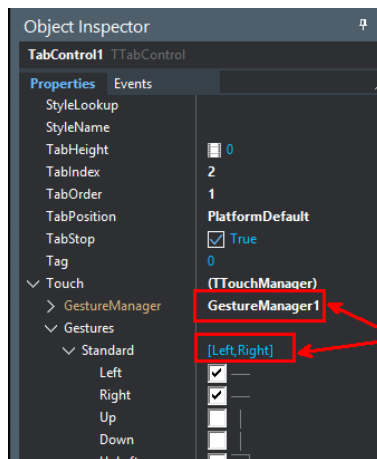
6. Warto dodać do aplikacji belkę z nagłówkiem, w której można umieścić nazwę aplikacji czy dodatkowe odnośniki do menu. W tym celu należy dodać komponent `TToolBar` z opcją `Align` jako `Top`. Dodatkowo do ToolBar należy dodać etykietę jako `TLabel` z opcją `Align` ustawioną na `Contents` dzięki czemu dopasuje się ona do zawartości. W polu `Text` wpisać swoją nazwę aplikacji a w polu `TextSettings>Font>HorzAlign` wybrać `Center`. Aplikacja powinna przyjąć następujący wygląd:



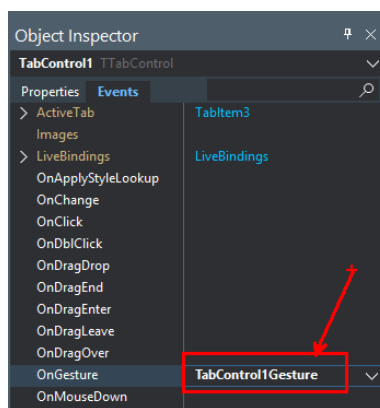
7. Poprzez podwójne kliknięcie komponentu TTabControl uruchomi się Items Editor. Należy w nim dodać trzy zakładki:



8. Następnie należy dodać do aplikacji komponent TGestureRecognizer. Dzięki niemu możliwe będzie dodanie obsługi gestów do zmiany aktualnej zakładki.
9. W komponentcie TTabControl należy ustawić odpowiednie pola aby dodać obsługę gestów:



10. Następnie należy dodać akcję po wykryciu gestów poprzez podwójne kliknięcie zdarzenia onGesture:



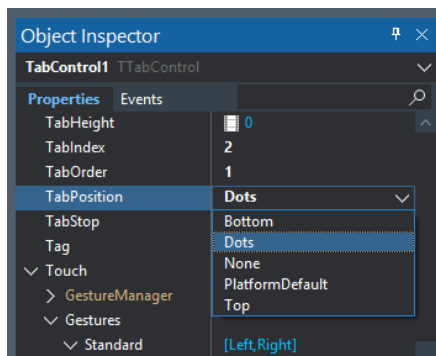
11. Aby dodać funkcjonalność zmian zakładek należy wprowadzić utworzonej metodzie poniższy kod:

```

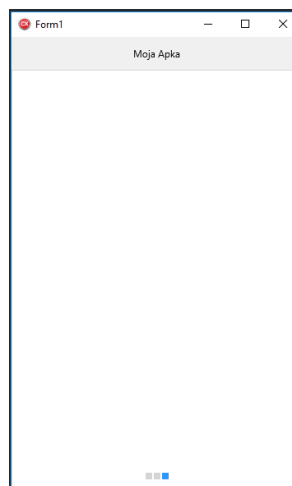
· void __fastcall TForm1::TabControl1Gesture(TObject *Sender,
· const TGestureEventInfo &EventInfo, bool &Handled) {
·     switch (EventInfo.GestureID) {
20     case sgiLeft:
·         TabControl1->ActiveTab =
·             TabControl1->Tabs[(TabControl1->TabIndex + 1)
·                 % TabControl1->TabCount];
·         Handled = true;
·         break;
·     case sgiRight: {
·         int idx = TabControl1->TabIndex - 1;
·         if (idx < 0)
·             idx += TabControl1->TabCount;
30     TabControl1->ActiveTab = TabControl1->Tabs[idx];
·         Handled = true;
·         break;
33     }
·     default: ;
·     }
· }
· // -----

```

12. W celu poprawy przejrzystości interfejsu można zamienić sposób wyświetlania zakładek na bardziej popularne kropki:



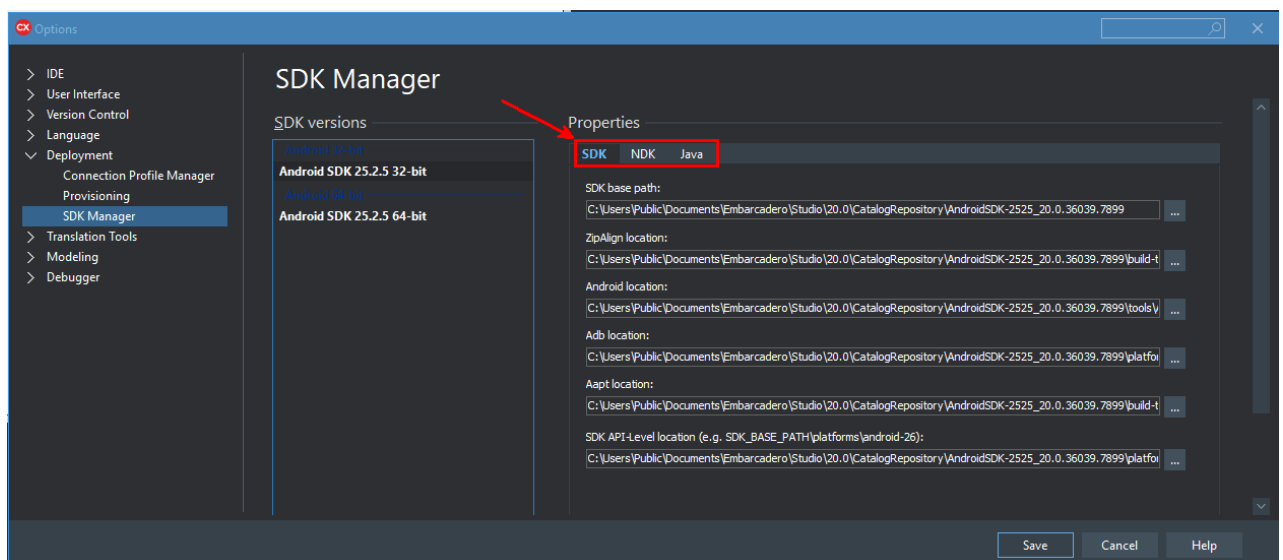
13. Aplikacja po skompilowaniu i uruchomieniu pod kontrolą systemu Windows 10 powinna wyglądać w następujący sposób:



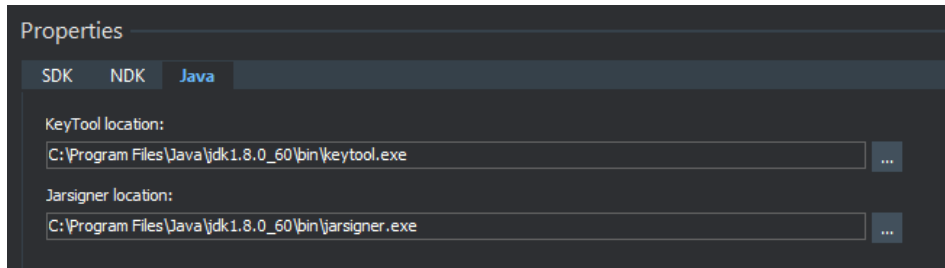
14. Wykonane czynności są praktycznie tożsame z wykorzystaniem szablonu **Tabbed with Navigation** na etapie tworzenia nowej aplikacji.

2.4 Kompilacja i wgrzywanie aplikacji

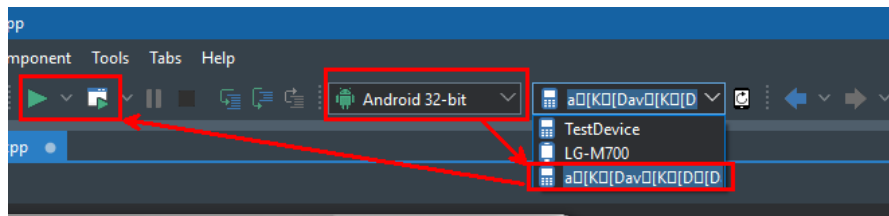
1. Zarówno w przypadku emulatora jak i fizycznie podłączonego urządzenia proces kompilacji i wgrzywania wygląda tak samo.
2. W pierwszym kroku należy się upewnić, że ścieżki zarówno do narzędzi Android SDK, NDK jak i Javy (JDK) są poprawnie ustawione w **Tools -> Options**:



3. Narzędzia SDK i NDK są instalowane domyślnie w folderze `C:\Public\Documents\Embarcadero\Studio\20.0\CatalogRepository` i powinny być powiązane ścieżkami automatycznie. W przypadku JDK o ile nie był wcześniej zainstalowany, znajdujący się w powyższej ścieżce instalator należy ręcznie uruchomić i dodać powiązania do JarSigner i KeyTool:

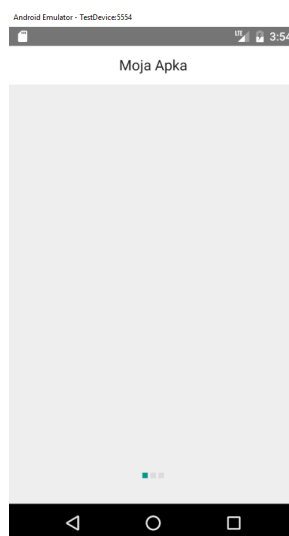


4. Następnie należy wybrać na które urządzenie Android ma zostać wgrana aplikacja i uruchomić opcję Run lub Run without Debugging:



W przypadku emulatora nazwa może przybrać losowy ciąg znaków (nazwa utworzonego emulatora będzie widoczna ale nie będzie dawała możliwości wgrania). Gdy na liście brakujeżądanego urządzenia możliwy jest problem ze sterownikami lub nie uruchomiony emulator.

5. W procesie wgrywania zostanie wygenerowany plik `NazwaProjektu.apk` i umieszczony w katalogu projektu pod ścieżką `\Android\Debug\NazwaProjektu\bin\`
6. Aplikacja po skompilowaniu, wgraniu i uruchomieniu na emulatorze systemu Android powinna wyglądać w następując sposób:



3 Zadania do samodzielnego wykonania

1. (obowiązkowe do zaliczenia) Do utworzonych zakładek w aplikacji dodać:
 - (a) Na zakładce pierwszej dodać 2 przyciski pozwalające na przełączenie na zakładki 2 i 3, oraz etykietę reprezentującą wybraną datę. Po uruchomieniu aplikacji ma być pokazywana aktualna data.
 - (b) Na zakładce drugiej dodać komponent kalendarza ustawiający datę na zakładce pierwszej oraz przycisk powrotu.
 - (c) Na zakładce trzeciej dodać pole adresu (TEdit) oraz komponent TWebBrowser. Po wpisaniu adresu i zatwierdzeniu powinna uruchomić się wskazana strona WWW.
2. (zadanie dodatkowe) n.d.