Payment Plugin for Unity 3.4

Build for iOS

After make Unity XCode project - add framework: StoreKit.

Testing for Mac OS

Build application with enabled «Mac App Store Validation». Sign up and run the application. This will open a window in which to enter your login and password for the test account. After that, your application will run in Sandbox.

Features

- Check support payment service
- Retrive all products information with one method
- Retrive receipt in success payment callback for server-side checking (tempory only for iOS)
- Restore completed transactions

Using

1. Declare payment events in project class:

```
void PaymentEvent_Products( Product[] _Products )
{
      // show products on GUI with all information from Product class
}

void PaymentEvent_Complete( string _ProductID, byte[] _Receipt )
{
      // success payment, use receipt data if you want check payment
      // on you server
}

void PaymentEvent_Failed( string _ProductID, string _Description )
{
      // failed payment, show error with description
}

void PaymentEvent_Canceled( string _ProductID, string _Description )
{
      // canceled payment, show error with description
}
```

2. Register events in payment plugin

```
Payment.Products += PaymentEvent_Products;
Payment.Complete += PaymentEvent_Compete;
Payment.Failed += PaymentEvent_Failed;
Payment.Canceled += PaymentEvent_Canceled;
```

3. Check payment support:

```
if (Payment.canMakePayment())
{
      // payment is available
}
else
{
      // payment is unavailable
}
```

4. If payments is available retrive all products information and retrive result in method PaymentEvent_Products:

```
Payment.getProductInfo( new string[]{
            «product1»,
            «product2»,
            «productN»
      } );
```

5. If user click in product use method:

Payment.makePayment(«product2»);

As result, plugin call one of the methods: PaymentEvent_Complete, PaymentEvent_Failed, PaymentEvent_Canceled. If you have client-server product, you can check payment with receipt in PaymentEvent_Complete method.

Restore Completed Transactions

1. Declare all events for restoring

```
void PaymentEvent_RestoreFailed( string _Description )
{
      // restore failed, show error with description
}

void PaymentEvent_RestoreFinished()
{
      // restore finished
}

void PaymentEvent_Restored( string _ProductID, byte[] _Receipt )
{
      // success restore, use receipt data if you want check payment
      // on you server
}
```

2. Register restore events in payment plugin

```
Payment.RestoreFailed += PaymentEvent_RestoreFailed;
Payment.RestoreFinished += PaymentEvent_RestoreFinished;
Payment.Restored += PaymentEvent_Restored;
```

3. Check payment support

```
if (Payment.canMakePayment())
{
      // payment is available
}
else
{
      // payment is unavailable
}
```

4. If payment available begin restore completed transactions

```
Payment.restoreCompletedTransactions();
```

As result, plugin call PaymentEvent_Restored for each transaction and finish all calls with event PaymentEvent_RestoreFinished. If cant finish restore plugin call event PluginEvent_RestoreFailed.