

Objective C

Aggiornare, gestire e visualizzare le posizioni dell'utente

```
- (void) onUnityLocations:(NSDictionary *)locations
{
    for(NSNumber *key in [locations allKeys])
    {
        ProfileLocation *affiliateLocation = [locations objectForKey:key];

        if(![self.unityAffiliateMarkers objectForKey:key])
        {
            Profile *affiliateProfile = [[Profile alloc] init];
            affiliateProfile.entityId = affiliateLocation.profileId;
            if(!affiliateProfile.entityId==0)
            {
                [self readUnityObject:affiliateProfile withCompletionBlock:^(id object, bool success, NSString * errorMessage)
                {
                    if(success) // generate a GMSMarker containing the profile object
                    {
                        GMSMarker *affiliateLocationMarker = [GMSMarker markerWithPosition:CLLocationCoordinate2DMake(affiliateLocation.latitude,
affiliateLocation.longitude)];
                        affiliateLocationMarker.flat = YES;
                        affiliateLocationMarker.zIndex = -1;
                        affiliateLocationMarker.appearAnimation=kGMSMarkerAnimationPop;
                        affiliateLocationMarker.groundAnchor = CGPointMake(0.5, 0.5);
                        [affiliateLocationMarker setUserData:((Profile *)object)]; // this sets the profile to the marker
                        affiliateLocationMarker.icon= [TenDegreesHelpers roundImageCorners:[self applyAlpha:1 toImage:[self image:[TenDegreesHelpers
stringToUIImage:((Profile *)object).icon] scaledToSize:CGSizeMake(30.0f, 30.0f)]]];

                        if(![self.unityAffiliateMarkers objectForKey:key])
                        {
                            {
                                affiliateLocationMarker.map=self.googleMap;
                                [self.unityAffiliateMarkers setObject:affiliateLocationMarker forKey:key];
                            }
                        }
                        else
                        {
                            {
                                NSLog(@"%@",errorMessage);
                            }
                        }
                    }
                }];
            }
        }

        }
    }
    else // users location and profile are already cached - just update their location on the map
    {
        GMSMarker *affiliateLocationMarker = [self.unityAffiliateMarkers objectForKey:key];
        [affiliateLocationMarker setPosition:CLLocationCoordinate2DMake(affiliateLocation.latitude, affiliateLocation.longitude)];
    }
}

for(NSNumber *key in [self.unityAffiliateMarkers allKeys])
{
    if(![locations objectForKey:key])
    {
        GMSMarker *affiliateLocationMarker = [self.unityAffiliateMarkers objectForKey:key];
        affiliateLocationMarker.map=nil;
        [self.unityAffiliateMarkers removeObjectForKey:key];
    }
}
```

```
[_extendUpdateDelegate onLocations:locations];  
}
```

Visualizzare la data in termini di "ieri", "oggi"...

```
-(NSString *)relativeDateStringForDate:(NSDate *)date  
{  
    NSCalendarUnit units = NSCalendarUnitDay | NSCalendarUnitWeekOfYear |  
        NSCalendarUnitMonth | NSCalendarUnitYear;  
  
    NSDateComponents *components = [[NSCalendar currentCalendar] components:units  
        fromDate:date  
        toDate:[NSDate date]  
        options:0];  
  
    if (components.year > 0) {  
        return [NSString stringWithFormat:@"%ld anni fa", (long)components.year];  
    } else if (components.month > 0) {  
        return [NSString stringWithFormat:@"%ld mesi fa", (long)components.month];  
    } else if (components.weekOfYear > 0) {  
        return [NSString stringWithFormat:@"%ld settimane fa", (long)components.weekOfYear];  
    } else if (components.day > 0) {  
        if (components.day > 1) {  
            return [NSString stringWithFormat:@"%ld giorni fa", (long)components.day];  
        } else {  
            return @"Ieri";  
        }  
    } else {  
        return @"Oggi";  
    }  
}
```