Mapa cieplna ruchu myszki po ekranie

Wojciech Waniek

Start

Program używa bazy plikowej SQLite. Jeśli przy starcie baza nie istnieje to jest tworzona.

```
protected override void OnModelCreating(DbModelBuilder modelBuilder)
{
        modelBuilder.Entity<ScreenUnit>();
        var sqliteConnectionInitializer = new
                SqliteCreateDatabaseIfNotExists<MouseHeatmapDbContext(modelBuilder);
         Database.SetInitializer(sqliteConnectionInitializer);
}
```

Baza danych

Jedna encja: ScreenUnit.

```
public class ScreenUnit
{
    public int ScreenUnitId { get; set; }
    public int Y { get; set; }
    public int X { get; set; }
    public long MousePassedCount { get; set; }
    public long MouseFinishedCount { get;  set; }
    public long SpeedCount { get;  set; }
}
```

Baza danych

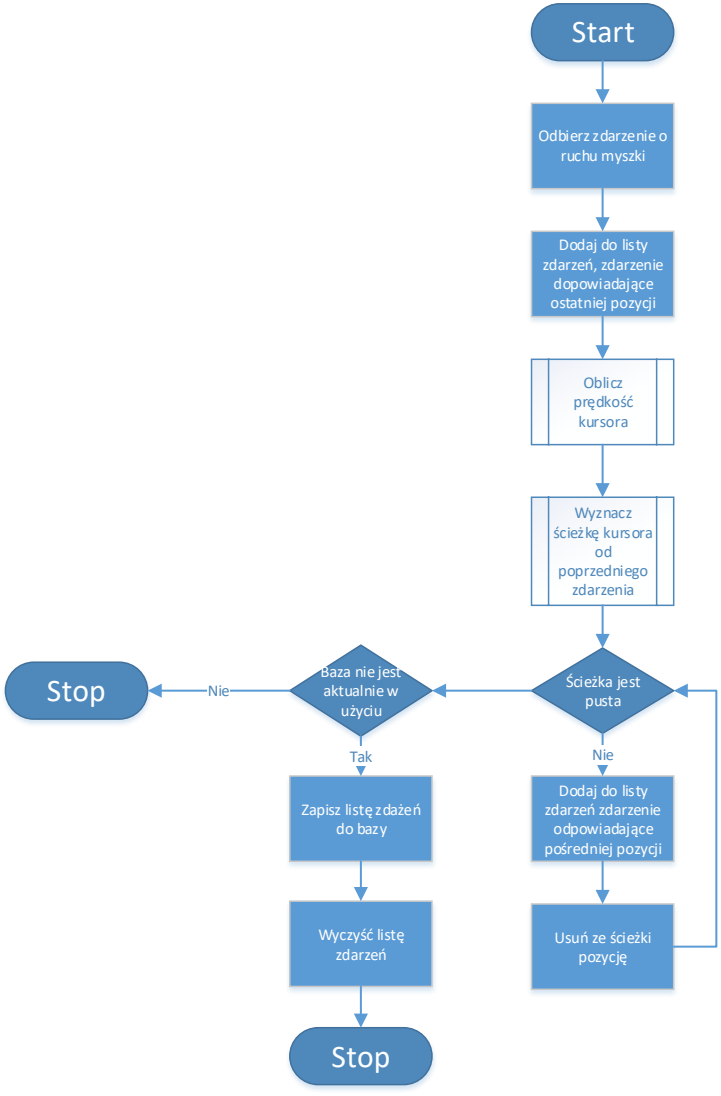
Tylko jeden proces może w danym momencie pisać do plikowej bazy danych.

```
private void AttemptSavingToDatabase()
{
        if (DatabaseUpdateTask.IsCompleted)
    {
        Log.Debug("Trying to save: " + _screenUnitsForSaving.Count);

                DatabaseUpdateTask =_dataRecorder
                .SaveAsync(new List<ScreenUnit>(_screenUnitsForSaving));

            _screenUnitsForSaving.Clear();
     }
}
```

Schemat blokowy algorytmu po ruchu myszką



Testy

Testy integracyjne. Baza testowa jest tworzona dla każdego testu.

```
[Test]
public async Task adds_correct_entries_to_database()
{
    SetNow(0);
    StartCollecting();

    SetNow(1000000);
    MoveMouseTo(25, 0);

    await WaitForLastDatabaseUpdate();

    long speedCount = 25*(long)Math.Pow(10,7)/ 1000000;
    var expectedScreenUnits = new List<ScreenUnit>
    {
            ...
    };

    GetScreenUnits().ShouldAllBeEquivalentTo(
    expectedScreenUnits,
    options => options.Excluding(su => su.ScreenUnitId));
}
```

Generowanie mapy cieplnej

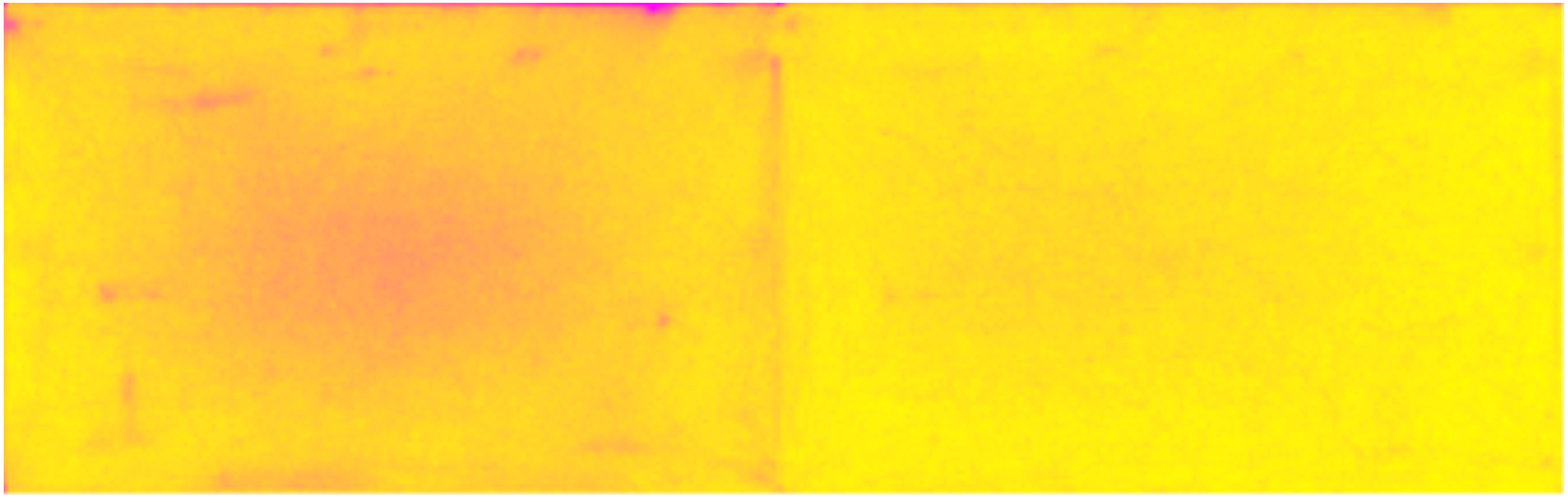
Czym dana pozycja na ekranie jest częściej odwiedzana tym będzie zaznaczona bardziej na czerwono.

```
private static Color TranslateValueToColor(long count, long min, long max)
        {
            double relativeValue = (double)(count   min) / (max   min);

            return Color.FromArgb(
                255,
                (int)(255 * (1   relativeValue)),
                (int)(255 * relativeValue));
        }
```

Wyniki

Mapa pokazująca ostatnie położenia kursora:



Wszystkie położenia kursora:

Wyniki

Prędkość kursora: