



**AGH**

**AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE**

**WYDZIAŁ ELEKTROTECHNIKI, AUTOMATYKI,  
INFORMATYKI I INŻYNIERII BIOMEDYCZNEJ**

**Symulacja Dyskretna Systemów Złożonych**

*Symulacja zagrożenia lawinowego w Tatrach Polskich*

Autorzy:

Kierunek studiów:

Opiekun pracy:

*Marek Jachym, Wojciech Konieczkiewicz*

*Informatyka*

*dr hab. inż. Jarosław Wąs*

Kraków, 2020

## Spis treści

<b>1. Wprowadzenie</b>	3
1.1. Cel	3
1.2. Opis problemu	3
1.3. Możliwe rozwiązanie	3
<b>2. Stworzony model zjawiska</b>	5
2.1. Opracowanie danych topograficznych	5
2.1.1. Format .las	5
2.1.2. Wybór cech	6
2.1.3. Triangulacja Delaunaya	6
2.2. Opracowanie danych pogodowych	7
2.2.1. Progi pogodowe	7
2.2.2. Pobieranie danych pogodowych	8
2.3. Narzędzia	8
2.4. Przygotowanie dokumentu	8
<b>3. Symulacja zjawiska - implementacja i szczegóły techniczne</b>	9
3.1. Wybór języka programowania	9
3.2. Narzędzia wykorzystywane w trakcie tworzenia projektu	9
3.2.1. Środowisko programowania	9
3.2.2. Wybrane biblioteki zewnętrzne języka Python	10
3.2.3. Usługi chmurowe Google Cloud Platform i dane pogodowe	10
<b>4. Graficzny interfejs użytkownika</b>	13
4.1. Pierwszy etap implementacji	13
4.1.1. Mapa Tatr	13
4.2. Etap drugi - główne okno aplikacji	14
4.3. Etap trzeci - interaktywna mapa Tatr	15
4.4. Etap czwarty - okna detaliczne	16
4.5. Obsługa aplikacji i jej działanie	17

# **1. Wprowadzenie**

## **1.1. Cel**

Celem niniejszej pracy jest stworzenie symulacji, która dostarcza w czasie rzeczywistym informacji dotyczących możliwego zagrożenia lawinowego na terenie Tatr w Polsce.

## **1.2. Opis problemu**

Lawiny śnieżne są powszechnie występującym zagrożeniem na całym świecie i stanowią niebezpieczeństwo zarówno dla ludzi, jak i biznesu oraz infrastruktury. Instytucje w różnych państwach alarmują o zagrożeniu lawinowym w podobny sposób, choć można zauważyć, że im większe zagrożenie stanowi zjawisko samoistnego ruchu śniegu, tym bardziej szczegółowe są prognozy.

Głównym problemem przy określaniu ryzyka lawinowego jest złożoność tego zjawiska. Zależy ono zarówno od czynników stałych, do których zalicza się np. ukształtowanie terenu (1. typ uwarunkowań) oraz zmiennych, dotyczących stanu śniegu (2. typ uwarunkowań) oraz warunków meteorologicznych (3. typ uwarunkowań) (Woszczek 2016). Polskie instytucje wydają regularnie tzw. komunikaty lawinowe, są one jednak jedynie ogólnym zapisem zagrożenia - ratownicy zastrzegają, że informacje zawarte w komunikacie stanowią tylko podstawę do samodzielnej oceny.

Przedmiotem poniższej pracy jest więc próba zastosowania technologii, by zapewnić system wspomagający decyzje ekspertów w celu tworzenia jeszcze bardziej precyzyjnych ostrzeżeń. Symulacja sama w sobie nie stanowi profesjonalnego narzędzia, ale zawiera koncepcje i rozwiązania, które można zastosować przy tworzeniu niezawodnego i złożonego narzędzia dla państwowych instytucji.

## **1.3. Możliwe rozwiązanie**

Na świecie modele przewidujące zagrożenie są tworzone w oparciu o metody statystyczne oraz metody uczenia maszynowego takie jak analiza najbliższego sąsiedztwa (ang. nearest neighbor analysis), analiza skupień (ang. cluster analysis), czy drzewa klasyfikacyjne (ang. classification trees) (Joshi, Kumar, Srivastava, Sachdeva, Ganju 2018). Są to jednak rozwiązania stworzone na podstawie wieloletnich

pomiarów (również dotyczących warunków pokrywy śnieżnej), do których nie uzyskano dostępu. Zdecydowano się więc oprzeć na wnioskach autorów publikacji, charakteryzujących najważniejsze czynniki stwarzające ryzyko.

W celu jak najdokładniejszego określania ryzyka wykorzystano bardzo precyzyjne informacje dotyczące ukształtowania terenu oraz dane pogodowe (dane dotyczące śniegu nie są ogólnodostępne i ich zmierzenie wymaga specjalistycznej wiedzy oraz narzędzi).

Korzystając z danych topograficznych uproszczono ukształtowanie powierzchni Tatr (każdy z obszarów o powierzchni około 4 km<sup>2</sup> reprezentowany jest przy pomocy około 110 punktów) i obliczono odpowiednie cechy. Następnie w połączeniu z cechami dotyczącymi warunków atmosferycznych możliwe stało się określenie ryzyka dla każdego takiego obszaru przy pomocy stworzonego wcześniej drzewa decyzyjnego. Dzięki takiemu podejściu, cały obszar Tatr Polskich podlega obserwacji, a w razie wystąpienia sprzyjających warunków (wykorzystano 2 z 3 istniejących uwarunkowań) w sposób zautomatyzowany wydaje się odpowiednie ostrzeżenia.

## 2. Stworzony model zjawiska

Niniejszy rozdział opisuje szczegółowo kolejne kroki oraz wykorzystane algorytmy niezbędne do uzyskania efektu końcowego.

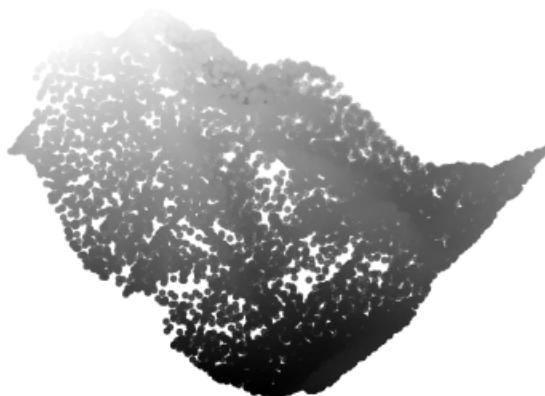
### 2.1. Opracowanie danych topograficznych

#### 2.1.1. Format .las

Na początkowym etapie pracy otrzymano pliki w formacie .las - każdy z nich reprezentujący ukształtowanie terenu wybranego obszaru.

Format .las został stworzony do przechowywania zbioru punktów w przestrzeni trójwymiarowej (ang. point cloud), otrzymanych przy pomocy metody Lidar, która polega na oświetlaniu wybranych punktów na powierzchni Ziemi laserem i zapisie jego odbicia przy pomocy sensorów. Dzięki tej metodzie powstają mapy o wysokiej rozdzielczości, stosowane w naukach o Ziemi (źródło: angielska wiki).

Otrzymane pliki zawierały średnio około 11 milionów punktów, przy czym każdy z nich reprezentował powierzchnię około 4 km<sup>2</sup>.



**Rys. 2.1.** Uproszczona wizualizacja zbioru punktów z pliku .las przy użyciu biblioteki matplotlib

### 2.1.2. Wybór cech

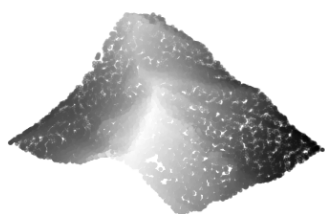
Jak wspomniano już wcześniej, do oszacowania ryzyka lawinowego konieczne jest posiadanie danych dotyczących cech terenu. Bazując na pracy pani Izabeli Woszczak, skupiono się na obliczeniu takich cech, jak:

- forma terenu (skupiono się na żlebach);
- ekspozycja słoneczna;
- nachylenie powierzchni;
- piętro;
- wysokość.

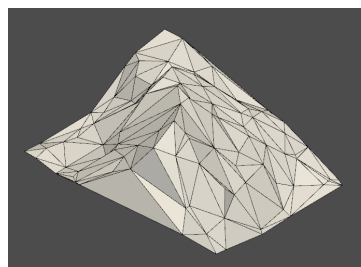
### 2.1.3. Triangulacja Delaunaya

Sam zbiór punktów nie oferuje możliwości łatwego obliczania wyżej wymienionych cech, dlatego zastosowano uproszczenie powierzchni terenu przy pomocy triangulacji Delaunaya.

Jest to algorytm, który na podstawie zbioru punktów tworzy zbiór trójkątów, gdzie wierzchołki każdego trójkąta stanowią owe punkty. Własnością algorytmu jest, że maksymalizuje on najmniejsze z kątów w powstałych trójkątach, unikając tzw. sliver triangles.



**Rys. 2.2.** Obszar przedstawiony jako zbiór punktów



**Rys. 2.3.** Ten samo obszar poddany triangulacji

Tutaj możesz dodać potem jak dokładniej policzyłeś ostatecznie te nachylenia, bo szczerze mówiąc nie zaglądałem do tego jakoś szczegółowo xD

## 2.2. Opracowanie danych pogodowych

Dane pogodowe są kolejnymi kluczowymi danymi potrzebnymi do przewidzenia wystąpienia lawiny na danym obszarze. Wyodrębnione zostały 4 najważniejsze cechy wpływające na występowanie lawiny.

1. Grubość pokrywy śnieżnej
2. Zmiany temperatury
3. Opad deszczu
4. Prędkość wiatru

### 2.2.1. Progi pogodowe

Na podstawie słowackiej strony <http://www.laviny.sk> przyjęliśmy następujące progi dla poszczególnych cech.

#### 2.2.1.1. Grubość pokrywy śnieżnej

Dolny próg wysokości pokrywy śnieżnej niezbędnej do wystąpienia lawiny obraliśmy jako 20 centymetrów. Ze względu na bardzo ograniczone dane o zalegającej pokrywie śnieżnej na chwilę obecną tę cechę jako spełnioną uznajemy w momencie, gdy w ostatnich 48 godzinach opad śniegu przekroczył próg 200mm.

#### 2.2.1.2. Zmiany temperatury

Temperatura, zwłaszcza w okresie wczesnej lub późnej zimy, często ulega zmianom w okolicach 0°C. Pokrywa śnieżna może nadtopnieć w dodatniej temperaturze i następnie zamarznąć, tworząc śliską warstwę która, kiedy ma na sobie warstwę śniegu, może w wysokim stopniu sprzyjać osunięciu się pokrywy i spowodowaniu lawiny. Z tego powodu przyjmujemy iż kiedy w ciągu ostatnich 48 godzin temperatura ulegnie zmianie z ujemnej na dodatnią, i do tego średnia temperatura jest powyżej zera, uznajemy tę cechę jako zwiększającą zagrożenie lawiną.

#### 2.2.1.3. Opad deszczu

Nie ma podanych konkretnych wartości co do opadów deszczu, jedynym punktem odniesienia jest to iż opad musi być intensywny żeby przyczynił się do powstania lawiny. Z tego powodu przyjęliśmy iż jeżeli w ciągu ostatnich 48 godzin wystąpił opad o minimalnej intensywności 10mm/h to ta cecha jest uznawana za spełnioną.

#### 2.2.1.4. Prędkość wiatru

Jeżeli w ciągu ostatnich 48 godzin wystąpił wiatr o minimalnej prędkości 13m/s to tę cechę uznajemy jako zwiększającą zagrożenie lawiną.

### 2.2.2. Pobieranie danych pogodowych

Do pobrania danych pogodowych skorzystaliśmy z Open Weather Map API umożliwiającego łatwe i szybkie pobranie danych pogodowych.

## 2.3. Narzędzia

### 2.4. Przygotowanie dokumentu

Plik źródłowy  $\text{\LaTeX}$ a jest zwykłym plikiem tekstowym. Przygotowując plik źródłowy warto wiedzieć o kilku szczegółach:

- Poszczególne słowa oddzielamy spacjami, przy czym ilość spacji nie ma znaczenia. Po kompilacji wielokrotne spacje i tak będą wyglądały jak pojedyncza spacja. Aby uzyskać *twardą spację*, zamiast znaku spacji należy użyć znaku *tyldy*.
- Znakiem końca akapitu jest pusta linia (ilość pustych linii nie ma znaczenia), a nie znaki przejścia do nowej linii.
- $\text{\LaTeX}$  sam formatuje tekst. **Nie starajmy się go poprawiać**, chyba, że naprawdę wiemy co robimy.



## 3. Symulacja zjawiska - implementacja i szczegóły techniczne

### 3.1. Wybór języka programowania

Językiem który postanowiliśmy wybrać do implementacji symulacji przedstawionego powyżej problemu wybrano język Python w wersji **3.7.x**.

Ze względu na jego prostotę, wydajność i obszerny wybór bibliotek zewnętrznych wybraliśmy zamiast innych, również bardzo wydajnych ale bardziej skomplikowanych języków takich jak C++ czy Java.

### 3.2. Narzędzia wykorzystywane w trakcie tworzenia projektu

#### 3.2.1. Środowisko programowania



Rys. 3.1. PyCharm Community

Wykorzystano **PyCharm Community 2019.3**, rozbudowane, potężne i wyposażone w dużą ilość dodatkowych narzędzi środowisko stworzone do pracy z językiem Python. Dostępne jest w dwóch wersjach, Professional oraz Community będąca wersją open-source przez co idealnie nadała się do wykorzystania przy naszym projekcie.

### 3.2.2. Wybrane biblioteki zewnętrzne języka Python

Jak wspomniano wcześniej, Python jest bardzo popularnym językiem przez co posiada wiele bibliotek zewnętrznych.

Do implementacji symulacji problemu wykorzystano następujące biblioteki zewnętrzne:

1. **PyQt5** - oferująca zestaw narzędzi do budowy graficznego interfejsu użytkownika
2. **pyowm** - oferująca zestaw narzędzi do eksploatacji API Open Weather Map
3. **laspy** - zestaw funkcji umożliwiający wykonywanie wielu operacji na plikach .las
4. **paramiko** - moduł umożliwiający komunikację SSH z poziomu skryptu w Pythonie
5. **pyvista** - biblioteka zawierająca implementację algorytmu triangulacji Delaunaya

### 3.2.3. Usługi chmurowe Google Cloud Platform i dane pogodowe



Kluczowym, z punktu widzenia projektu, okazało się znalezienie rozwiązania problemu ciągłej aktualizacji danych pogodowych. Z powodu iż dane muszą być nieustannie aktualizowane co 8 godzin, jedna z maszyn roboczych autorów musiałaby pracować bez przerwy. Nie jest to w żadnym stopniu rozwiązanie optymalne.

Zdecydowano się na skorzystanie z darmowego okresu próbnego oferowanego na platformie **Google Cloud Platform**. Oferta Google zawierała w sobie możliwość stworzenia instancji maszyny wirtualnej operującej na systemie **Debian 10 Buster** co otworzyło drogę do rozwiązania problemu ciągłej aktualizacji danych pogodowych.

Dane pogodowe pobierane są co 8 godzin, o godzinach 00:00, 08:00 i 16:00 przy pomocy skryptu **weather\_conditions.py** który korzysta z biblioteki **pyowm**

weather\_conditions.py

```
#!/usr/bin/python2.7
import pyowm
import os
import time
import tarfile

#ZMIENNA path REPREZENTUJE ODPOWIEDNIE ŚCIEŻKI DO PLIKÓW
def get_weather_conditions(map_name):
    # UTWÓRZ ODPOWIEDNI FOLDER JEŻELI NIE ISTNIEJE
    if os.path.isdir(path):
        pass
    else:
        path = path
    os.mkdir(path)
    # WSPÓŁRZĘDNE ODPOWIEDNIEGO OBSZARU I CZAS POBRANIA DANYCH
    coords = extract_coords(map_name)
    czas_pomiaru = str(time.ctime()).split("_")
    # POBRANIE DANYCH POGODOWYCH
    owm = pyowm.OWM("7202a85833f71127c0a0b4fefc86ea2a")
    observation = owm.weather_around_coords(float(coords["N_lat"]), float(coords["W_long"]))

    # USUWANIE NAJSTARSZEGO POMIARU I ZASTĄPIENIE GO NOWYM
    data = os.listdir(path)
    data.sort()
    if len(data) >= 6:
        os.remove(path)
    # STWÓRZ PLIK ZAWIERAJĄCY AKTUALNE DANE POGODOWE
    filename = str(observation[0].get_weather().get_reference_time('date'))[0:10]
    data = open(path, "w+")
    data.write(str(observation[0].get_weather().get_temperature('celsius')['temp']) + "\n")
    data.write(str(observation[0].get_weather().get_snow()) + "\n")
    data.write(str(observation[0].get_weather().get_wind('meters_sec')) + "\n")
    data.write(str(observation[0].get_weather().get_rain()) + "\n")
```

### 3.2.3.1. Harmonogram pobrania danych

Za odpowiedni czas pobrania danych odpowiada narzędzie uniksopodobnych systemów operacyjnych **cron**. Jest to realizowane za pomocą odpowiednich wpisów w pliku **crontab**

```
0 6 * * * /usr/bin/python /home/marekjachym99/lavalanche/weather_conditions.py
```

```
0 14 * * * /usr/bin/python /home/marekjachym99/lavalanche/weather_conditions.py
```

```
0 22 * * * /usr/bin/python /home/marekjachym99/lavalanche/weather_conditions.py
```

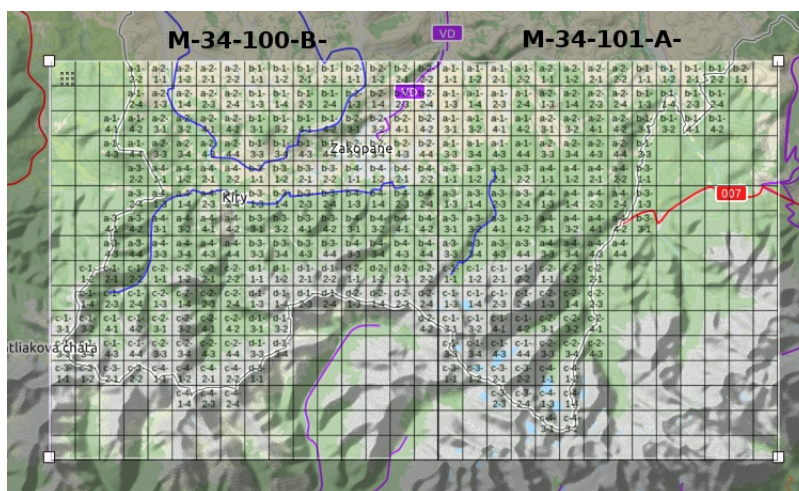
## 4. Graficzny interfejs użytkownika

Do stworzenia GUI (ang. Graphical User Interface) wykorzystano wcześniej wspomnianą bibliotekę PyQt5 oferującą zestaw narzędzi do budowy interfejsu użytkownika z użyciem klas QT

### 4.1. Pierwszy etap implementacji

Czytelny, prosty i przejrzysty interfejs użytkownika jest podstawą każdej dobrej aplikacji, niestety ze względu na wyzwania jakie niosła za sobą synchronizacja posiadanych map reprezentowanych przez zbiór plików .las z mapą geograficzną np. z Google Maps, nanoszenie zagrożeń na interaktywną mapę okazało się chwilowo poza zasięgiem.

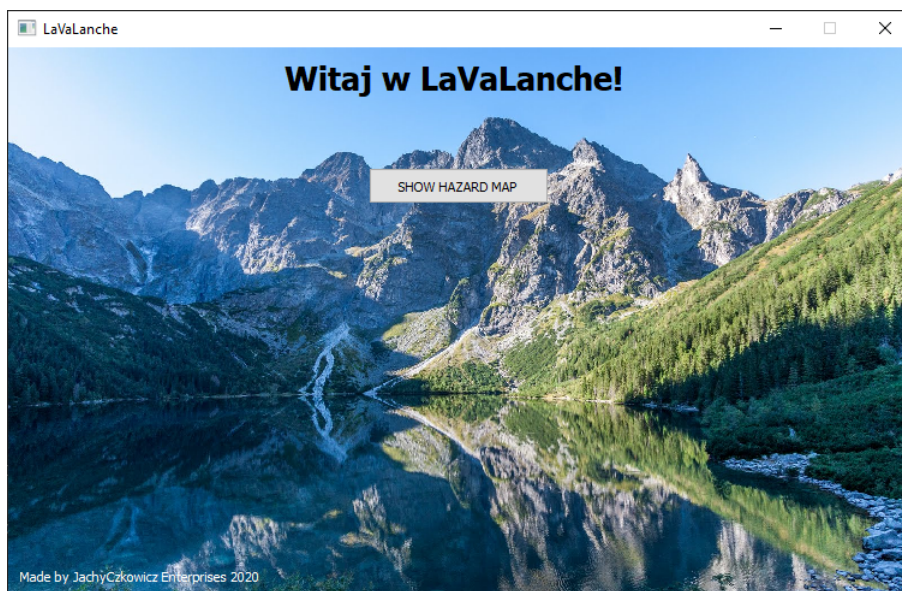
#### 4.1.1. Mapa Tatr



Rys. 4.1. Oryginalna mapa

## 4.2. Etap drugi - główne okno aplikacji

Stwierdzono, że główne okno aplikacji powinno zawierać jak najmniej zbędnych elementów i być czytelne jak i przykuwające uwagę, z tego powodu na ekranie głównym aplikacji widnieje piękny widok na Morskie Oko wraz z okalającymi je szczytami widziane ze schroniska.

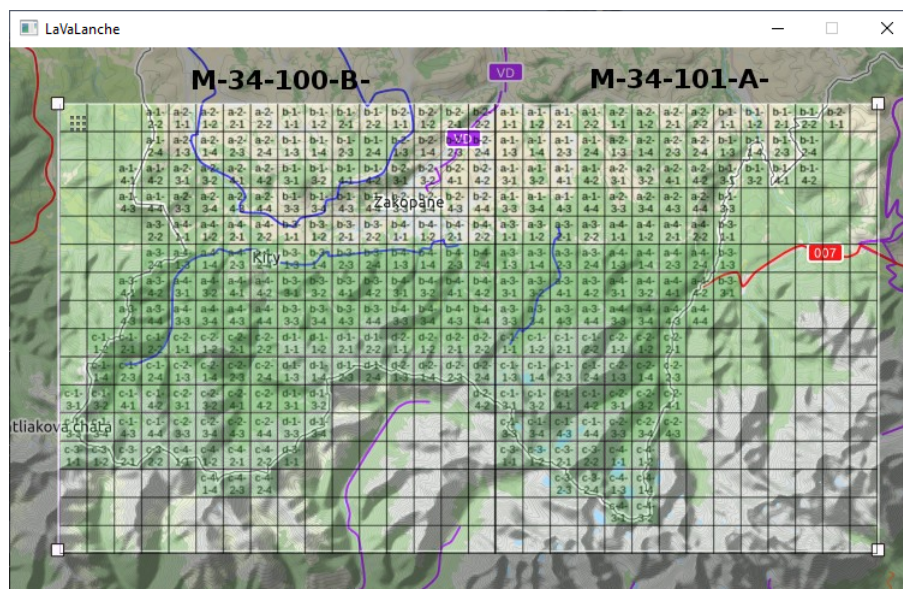


**Rys. 4.2.** Okno główne aplikacji

źródło: <http://podrozniczeretrospekcje.blogspot.com>

### 4.3. Etap trzeci - interaktywna mapa Tatr

Wykorzystując otrzymaną wcześniej mapę Tatr, postanowiono nałożyć na poszczególne kwadraty przyciski z których każdy odnosi się do zawartego pod nim obszaru.



Rys. 4.3. Mapa z naniesionymi na nią przyciskami

Założono, że zagrożenie lawiną obliczane będzie tylko dla obszarów znajdujących się na południe od Zakopanego z tego względu iż ukształtowanie terenu na północ od tego miejsca praktycznie wyklucza wystąpienie jakiegokolwiek lawiny i nie ma tam historii występowania lawin, dlatego przyciski, które kolorowane są według poniższej legendy, znajdują się tylko na mniej więcej połowie obszarów.

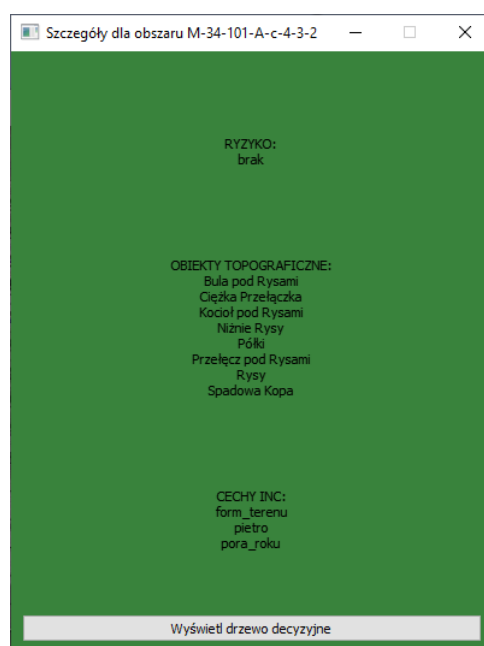


Rys. 4.4. Legenda kolorów przycisków

#### 4.4. Etap czwarty - okna detaliczne

Ostatni, lecz nie mniej ważny niż pozostałe etap, zakładał implementację okien detalicznych dla każdego obszaru autonomicznego tj. obszaru reprezentowanego przez jeden plik .las. W oknie detalicznym znajdują się takie dane jak:

- Stopień ryzyka
- Charakterystyczne obiekty na danym terenie np. szczyty
- Cechy zwiększające zagrożenie



Rys. 4.5. Okienko detaliczne



## 4.5. Obsługa aplikacji i jej działanie

Przed każdym uruchomieniem aplikacji aktualizowane są dane pogodowe z całego obszaru Tatr po polskiej stronie. Następnie ukazuje się okno główne aplikacji.

Na samym środku widnieje przycisk "SHOW HAZARD MAP" który po przyciśnięciu wyłącza okno główne i przechodzi do okna z interaktywną mapą obszaru.

W oknie interaktywnej mapy widnieją przyciski o kolorze odpowiadającym stopniowi zagrożenia, każdy z przycisków włącza okno detaliczne dla odpowiedniego obszaru autonomicznego. Na chwilę obecną nie ma możliwości powrotu do ekranu głównego.



## Bibliografia

- [1] A. Diller. *LaTeX wiersz po wierszu*. Gliwice: Wydawnictwo Helion, 2000.
- [2] L. Lamport. *LaTeX system przygotowywania dokumentów*. Kraków: Wydawnictwo Ariel, 1992.
- [3] M. Szpyrka. *On Line Alvis Manual*.  
<http://fm.ia.agh.edu.pl/alvis:manual>. AGH University of Science and Technology. 2011.