

AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA
STASZICA

KRAKÓW

Symulator pożaru lasu

Autorzy:

Sebastian KATSZER

Katarzyna KOSIAK

1 grudnia 2016

1 Wstęp

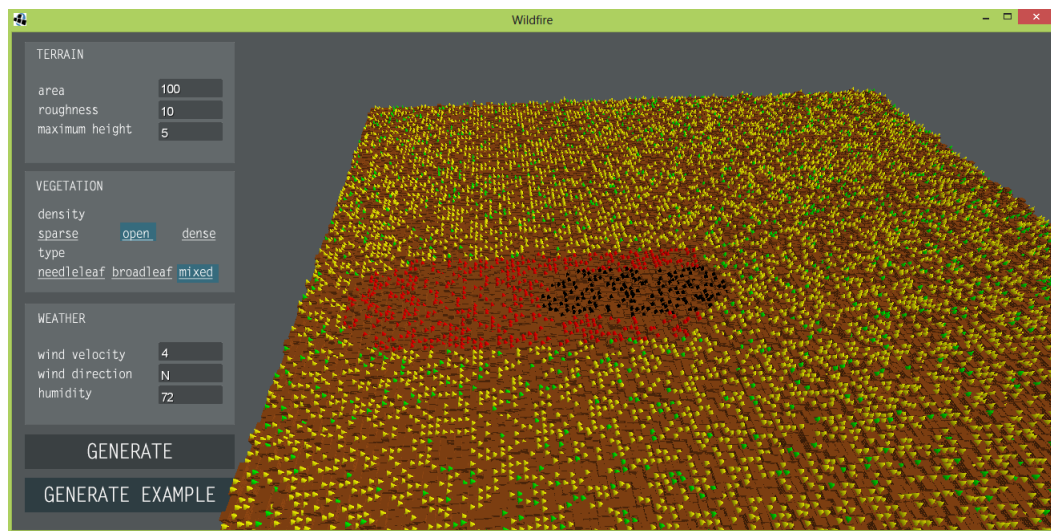
Niniejszy dokument stanowi specyfikację programu do symulowania pożarów lasów, który został stworzony na potrzeby przedmiotu "Modelowanie i symulacja systemów" i rozbudowany na potrzeby przedmiotu "Studio projektowe 2".

2 Funkcjonalność aplikacji

Program przyjmuje dane wejściowe od użytkownika i uruchamia symulację pożaru. Jest też możliwość uruchomienia wgranego, przykładowego pożaru bazowanego na początkowych etapach pożaru Kuźni Raciborskiej w 1992 roku.

Dane wejściowe wprowadzane przez użytkownika podzielone są na sekcje:

- Teren - długości boków terenu, nierówność terenu, maksymalna wysokość.
- Roślinność - trzy poziomy gęstości zalesienia (rzadki, zwykły, gęsty), trzy rodzaje lasów (liściasty, iglasty i mieszany).
- Pogoda - prędkość wiatru, kierunek wiatru, wilgotność powietrza.
- Szybkość symulacji.



Model wykorzystuje automaty komórkowe i model Rothermela do symulowania przebiegu pożaru. Linia ognia przybiera kształt eliptyczny i mają na nią wpływ wszystkie zadane przez użytkownika dane poza wilgotnością powietrza oraz nachyleniem stoku. Drzewa mają różny czas zapłonu w zależności od gatunku.

Trójwymiarowy model terenu można obracać w trakcie trwania symulacji. Po zakończeniu symulacji wyświetlany jest krótki raport informujący o czasie symulacji, ilości wszystkich drzew, ilości i procencie spalonych i żywych drzew.

3 Diagramy UML

Ze względu na duże wymiary plików z diagramami klas, nie zostały one włączone do tego dokumentu. Diagramy klas znajdują się w załączonym folderze o nazwie pictures. Znajdują się w nim:

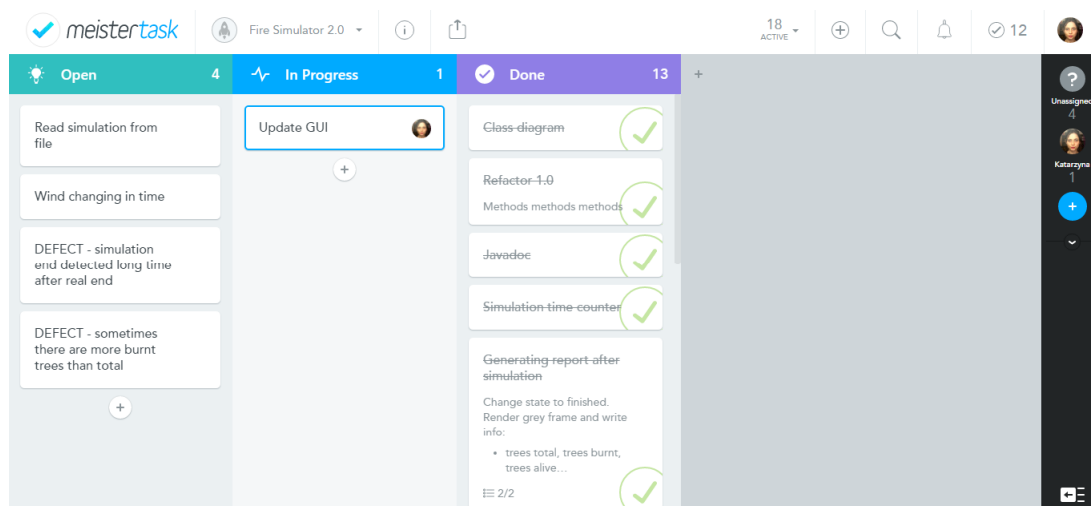
- simulation – Główna paczka odpowiedzialna za logiczną sferę symulacji
- data – Zawiera dane i stałe, na których bazuje symulacja, mogą być one aktualizowane podczas symulacji.
- graphic – Paczka ta zawiera klasy odpowiedzialne za wizualną stronę aplikacji.
- model – Przechowuje wykorzystywane abstrakcyjne modele odzwierciedlające rzeczywistość.

4 Zarządzanie projektem

W początkowych fazach projektu w rozporządzaniu zadaniami pomagała nam webowa aplikacja Trello oparta na tablicy kanban. Mimo przyjaznego interfejsu, z powodu zbyt małej elastyczności aplikacji przenieśliśmy się na inną webową aplikację - MeisterTask. Spełnia ona nasze oczekiwania dotyczące przejrzystości interfejsu i rozporządzania zadaniami w czasie.



Rysunek 1: Przykładowy stan naszego projektu na Trello



Rysunek 2: Przykładowy stan naszego projektu na MeisterTask

5 Krótki opis poszczególnych klas

- **GraphicController** - kontroluje wyświetlanie odpowiednich elementów GUI, obsługuje akcje użytkownika, zarządza wyświetlaniem symulacji i raportu.
- **GraphicUtils** - jest zbiorem dodatkowych metod przydatnych przy wyświetlaniu wewnętrznych danych w odpowiedniej formie i wykrywaniu akcji użytkownika.
- **Models** - zawiera i inicjalizuje trójwymiarowe modele dla celów symulacji.
- **Textures** - zawiera i inicjalizuje tekstury dla interfejsu użytkownika.
- **Simulation** - posiada ogólne informacje o symulacji i jej przebiegu, np. czas rozpoczęcia.
- **Cell** - jest najmniejszym elementem (komórką automatu) znajdującym się na symulowanym terenie. Reprezentuje roślinność i posiada swój stan.
- **Terrain** - jest zbiorem komórek i posiada metody odpowiadające za przebieg pożaru.
- **Data** - zbiór szczegółowych danych aktualnej symulacji oraz przykładowej symulacji pożaru Kuźni Raciborskiej.

6 Szczegółowy opis metod klas

Na stronie <http://student.agh.edu.pl/~kosiak/wildfireDocumentation/> znajduje się dokumentacja wygenerowana przy pomocy Javadoca. Zawiera ona krótkie opisy istotnych metod i podsumowanie funkcjonalności wszystkich klas.