



Generator Eventów

*Autor dokumentacji oraz Generатора Eventów: Krzysztof „SKP” Kukła
Korekta tekstu i oprawy: Marcel „mechatronik” Kobyłecki*

Dokumentację stworzono za pomocą: L^AT_EX

Spis treści

1	Podstawowe informacje o generatorze	3
1.1	Do czego służy program?	3
1.2	Funkcje Generатора Eventów	3
1.3	Wymagania Programu	4
1.4	Co się zmieniło od ostatniej wersji...	4
2	Omówienie funkcji Generатора Eventów	5
2.1	Tworzenie najprostszyc misji - <i>Event Multiple</i>	5
2.2	Dźwięki w scenariuszu - <i>event i node sound</i>	8
2.3	<i>Memcell</i> - nie taki diabeł straszny jak go malują	8
2.4	<i>Eventlauncher</i> - a to co?	9
2.5	Zdarzenia Losowe	9
2.6	Wstawianie Sygnalizatorów	10
2.7	Zdarzenie Losowe - algorytm YouBy'a	10
2.8	Inne Szablony	10
2.8.1	Zgłaszanie się na wjeździe	10
2.8.2	Zmienne	11
2.9	Prosty opis scenariusza w HyperText Markup Language (HTML)	11
3	Pozostałe przydatne informacje	12
3.1	Korzystanie z podglądu terenu <i>Rainsted</i>	12
3.2	FAQ	13
3.3	Tworzenie misji	13

Rozdział 1

Podstawowe informacje o generatorze

1

Niniejszy rozdział zawiera informacje o programie i jego funkcjach.

1.1 Do czego służy program?

1.1

Generator Eventów służy do zautomatyzowania procesu tworzenia scenariuszy oraz umożliwienia ich tworzenia do *SPS MaSzyna* bez konieczności znajomości składni eventów. Zawiera szereg szablonów dzięki którym wystarczy wypełnić kilka pól, aby otrzymać bardziej skomplikowane logicznie (tzn. inne niż proste) zdarzenia. Ponadto ogranicza możliwość popełnienia przez scenarzystę błędu do minimum.

1.2 Funkcje Generатора Eventów

1.2

W wersji 1.2.13 program ma następujące funkcje:

- Generowanie *Event Multiple*
- Generowanie *Node i Event Sound*
- Generowanie *Node Memcell i Event Updatevalues*
- Generowanie *Node Eventlauncher*
- Generowanie zestawu zdarzeń składowych do zdarzeń losowych 2 - 3 wariantowych
- Generowanie wpisów *Include* sygnalizatorów
- Generowanie zestawu zdarzeń składowych do zdarzeń losowych 2 - 8 wariantowych wg algorytmu losowania YouBy'a
- Generowanie zestawów zdarzeń składowych innych często spotykanych zdarzeń
 - Zgłaszanie się na wjeździe
 - Zmienne, czyli niejawne komórki pamięci

- Generowanie prostego opisu HTML scenariusza (jeden screen + tekst w czterech miejscach)

1.3 Wymagania Programu

1.3

Do poprawnego działania programu potrzebne jest środowisko uruchomieniowe .NET Framework.

Proszę nie zgłaszać błędów wynikających z braku lub niepoprawnego zainstalowania .NET Framework.

1.4 Co się zmieniło od ostatniej wersji...

1.4

CHANGELOG:

- Poprawiono błąd w funkcji przywracania automatycznego dopisywania eventów do torów
- Dodano możliwość zdefiniowania domyślnie wczytywanych plików scenerii (ścieżki zapisywane w config.evgc)
- Dodano zakładkę ustawienia
- Nowe okno informacji o programie
- Nowy sposób numeracji programu (po lekturze artykułu na Wikipedii, doszedłem do wniosku, że nru Revision nie trzeba zerować przy inkrementacji Minor, przy głównych zmianach inkrementować Minor, a przy poprawkach Build).
- Poprawione skalowanie przy zmianie rozmiaru okna grupy kontrolki odpowiedzialnych za wczytywanie SCM
- Dodano zakładkę *Hamowanie nagłe*

Rozdział 2

Omówienie funkcji Generатора Eventów

2

Ten rozdział traktuje o funkcjach Generatora Eventów i zawiera ich szczegółowy opis.

2.1 Tworzenie najprostszych misji - *Event Multiple*

2.1

Event Multiple jest najprawdopodobniej najprostszym przy czym najczęściej wykorzystywanym rodzajem zdarzenia w MaSzynie. Ma on za zadanie uruchomić inne zdarzenia - od jeden do ośmiu. Dodatkowo jego wykonanie można uzależnić od kilku warunków.

Nowością zastosowaną po raz pierwszy w wersji 1.1.11.0 (wg starej numeracji 1.11) jest bezpośredni zapis eventów do pliku scenarii. Do tej pory trzeba było dopisywać je ręcznie, aktualnie wystarczy podać linię, w której znajduje się wpis toru (więcej o tym w punkcie na temat korzystania z Rainsteda). Domyślnie trzeba tak jak dawniej ręcznie dopisywać eventy do torów, można to jednak zmienić na dole zakładki *start*. W przypadku błędnego dodania eventów, można przywrócić zawartość pliku scenarii do stanu sprzed wczytania.

- Jeśli zdarzenie jest wyzwalane przez SHIFT+..., nazwy nie trzeba podawać, bo i tak nie jest potrzebna.
- Jeśli zdarzenie jest wyzwalane przez zajęcie/zwolnienie odcinka izolowanego, w polu *Nazwa* powinna się znaleźć nazwa tegoż odcinka. O tworzeniu odcinków izolowanych (*Isolated*) można przeczytać na stronie www.rainsted.com/pl.
- Jeśli zdarzenie jest wyzwalane przez rozprucie rozjazdu, w polu *Nazwa* winna się znaleźć nazwa rozjazdu.
- Jeśli zdarzenie ma być wywołane przez inny event typu *Multiple*, należy wybrać jedną z pierwszych trzech opcji z listy "Wyzwolenie", jednak nie trzeba edytować wpisów torów, wystarczy sam kod wygenerowany do dolnego pola tekstowego.
- Jeśli zdarzenie ma być wyzwolone przez:
 - Pojazd stojący na torze
 - Pojazd wjeżdżający na tor z punktu 1
 - Pojazd wjeżdżający na tor z punktu 2

**UWAGA! PONIŻSZY OPIS DODAWANIA EVENTÓW DO TORÓW
DOTYCZY TYLKO OPCJI, KIEDY ZDECYDOWALIŚMY SIĘ NA RĘCZNE
DOPISYWANIE ZDARZEŃ DO TORU. OD WERSJI 1.11 ISTNIEJE
MOŻLIWOŚĆ AUTOMATYCZNEGO DOPISYWANIA EVENT0/1/2 DO
TORU.**

Konieczne będzie zmodyfikowanie wpisu toru. Za pomocą programu Rainsted można się dowiedzieć w której linii pliku znajduje się wcześniej wspomniany wpis. Przykładowy wpis wygląda tak:

```
node 1000 0 START track normal 100.0 1.435 0.25 25.0 20 0 flat vis
  rail_screw_used1 4 tpbps-new2 0.2 0.5 1.1
0.396825 0.2 136.111 0.0 //point 1
0.0 0.0 0.0 //control vector 1
0.0 0.0 0.0 //control vector 2
0.396825 0.2 236.111 0.0 //point 2
0
endtrack
```

START jest nazwą toru. Jeśli wybierzemy jedną z trzech pierwszych opcji wyzwolenia, w polu tekstowym pojawi się wpis zaczynający się od *event...* Należy go dodać do wpisu toru do liniiki wyżej niż *endtrack*, a jeśli jest *velocity*, też powinno się znaleźć nad nim. Zmodyfikowany w ten sposób powyższy wpis toru ostatecznie winien wyglądać mniej więcej tak:

```
node 1000 0 START track normal 100.0 1.435 0.25 25.0 20 0 flat vis
  rail_screw_used1 4 tpbps-new2 0.2 0.5 1.1
0.396825 0.2 136.111 0.0 //point 1
0.0 0.0 0.0 //control vector 1
0.0 0.0 0.0 //control vector 2
0.396825 0.2 236.111 0.0 //point 2
0
event2 odjazd_Calkowo
endtrack
```

- Można (ale nie trzeba) wybrać jeden z 5 warunków wykonania eventu. Nie można ich używać jeśli zaznaczona jest opcja *Wykonaj tylko raz*
- W przypadku wybrania warunku *Tor Zajęty/Tor Wolny*, zdarzenie wykona się tylko jeśli podany tor będzie zajęty/wolny. W przypadku wybrania tego warunku należy obowiązkowo podać nazwę toru.
- Prawdopodobieństwo - Zdarzenie wykona się tylko jeśli wylosowana liczba z zakresu 0..1 będzie mniejsza od podanej w warunku. Prawdopodobieństwo wyrażone jest ułamkiem dziesiętnym. Przykładowo, dla 10% podaje się 0.1.
Aby stworzyć zdarzenia losowe z kilkoma wariantami, konieczne będzie użycie komórek pamięci, dlatego w przypadku problemów z nimi, zalecam skorzystania z zakładki: *Zdarzenia Losowe* i *Zdarzenia Losowe - algorytm youBy'a*.

- Komórka pamięci - zdarzenie wykona się tylko jeśli wartości wskazanej komórki pamięci będą takie same jak wartości podane w warunku. Można pominąć sprawdzanie niektórych wartości poprzez zaznaczenie odpowiedniej opcji. Więcej o własnościach komórek pamięci można przeczytać w punkcie *Memcell*.
- Zmienna - podobnie jak Komórka pamięci
- Wykonaj tylko raz - nie pozwala na wykonanie zdarzenia więcej niż jeden raz.

2.2 Dźwięki w scenariuszu - *event i node sound*

2.2

Każdy dobry scenariusz powinien zawierać także rozmowy radiotelefoniczne i zapowiedzi pociągów na dworcach. Dzięki wstawieniu dźwięków znacznie wzrasta realizm scenariusza.

- Nazwa pliku dźwiękowego będzie użyta przez program do stworzenia wpisów *node* i *event*.
- Współrzędne można sprawdzić za pomocą programu *Rainsted*.
- Dźwięk musi być zgodny z wytycznymi dla twórców (<http://eu07.pl/forum/index.php/topic,15306.0.html>):

Standardem formatu dźwięków używanych w symulatorze jest wav PCM 22kHz, 8 bit, mono. Wyjątkiem od tej reguły są dźwięki modulowane (szybkość odtwarzania takich dźwięków jest zależna od prędkości pojazdu). Do każdego dźwięku modulowanego muszą zostać podane zalecane parametry odtwarzania, gdyż nie wiadomo, czy użytkownik miał wcześniej dźwięk z paczki, czy z jakiegoś dodatku.

2.3 *Memcell* - nie taki diabeł straszny jak go malują

2.3

Komórki pamięci (*memcell*) zdają się być początkującym scenarzystom czymś nie do ogarnięcia. Jednak wbrew pozorom ich użycie i tworzenie jest banalnie proste. Użycie komórek pamięci:

- Konstruowanie skomplikowanych instrukcji warunkowych
- Wykonywanie komend do AI (lista komend w *RFC.doc*)

O komórkach pamięci:

- Każda komórka pamięci powinna mieć unikalną nazwę w skali scenarii
- Komórkę pamięci można porównać do zmiennej, z tym że zawiera nie jedną, a 3 wartości:
 - Wartość tekstowa (np. *ZjechałJark*)
 - Wartość liczbowa 1 (np. *0*)
 - Wartość liczbowa 2 (np. *0*)
- Można zmienić wartości komórki pamięci używając eventu *updatevalues* (prawa ramka)
- Event *updatevalues* można wykonać wybierając w *Multiple* rodzaj zdarzenia *Inne* i podając tam nazwę eventu *updatevalues*

UWAGA! Jednym z najczęstszych użyć komórek pamięci jest zabezpieczenie zdarzenia przed ponownym użyciem oraz losowanie spośród kilku zdarzeń. Nie ma potrzeby ręcznego tworzenia tych zdarzeń, ponieważ zarówno jedno jak i drugie można zrobić za pomocą generatora dużo szybciej i prościej. Program w takim wypadku sam tworzy odpowiednie komórki pamięci i inne zdarzenia, a użytkownik nie musi się niczym przejmować.

2.4 *Eventlauncher* - a to co?

2.4

Choć na podstawie informacji zawartych w poprzednich paragrafach można zrobić cały scenariusz, jest jeszcze jeden dość często spotykany rodzaj zdarzenia. A właściwie nie zdarzenia, gdyż eventlauncher należy do klasy node a nie event (co oznacza, że nie można go wywołać innym eventem np. przez *Inne w Multiple*), ale jeśli nie rozumiesz o co chodzi, nie ma się czym przejmować, nie musisz tego wiedzieć ;) Co można zrobić za pomocą eventlaunchera:

- Uruchomić zdarzenie w wyniku naciśnięcia klawisza alfanumerycznego na klawiaturze (z SHIFTem lub bez)
- Uruchomić zdarzenie o określonej godzinie (oczywiście liczy się czas w symulacji, a nie w systemie)
- Uruchamiać zdarzenie co określoną liczbę sekund

Jak stworzyć eventlauncher za pomocą generatora:

- Nazwa powinna być unikalna w skali scenerii
- *Maksymalna odległość obserwatora od obiektu* - czyli maksymalnie ile metrów od współrzędnych eventlaunchera winniśmy się znajdować, by eventlauncher się wykonał. Aby zdarzenie wykonało się niezależnie od pozycji kamery, pozostaw pole puste
- Bezwzględnie należy podać nazwę zdarzenia wywołwanego przez eventlauncher

UWAGA! Aby symulować prośbę maszynisty o wjazd na stację, można skorzystać z *Innych szablonów*, w których w zakładce *Zgłaszanie się na wjeździe* program stworzy nie tylko eventlauncher, ale też na życzenie użytkownika dodatkowe zdarzenia jak blokada przed ponownym wykonaniem.

2.5 Zdarzenia Losowe

2.5

Jak już wcześniej było wspomniane, zdarzenia losowe można stworzyć na kilka sposobów. Nie jest to zbyt trudne, ale wymaga umiejętności tworzenia komórek pamięci, zmieniania ich zawartości i wiele innych. Nawet jeśli komuś nie sprawia to problemu, to trzeba napisać dość dużo zdarzeń i zajmuje to więcej czasu.

Dlatego w generatorze wystarczy podać prawdopodobieństwo i nazwy zdarzeń, a program stworzy na podstawie identyfikatora wszystkie składowe *event* i *node*.

- Identyfikator powinien być unikalny w skali innych zdarzeń losowych
- Prawdopodobieństwo podaje się w ułamku dziesiętnym (np. dla 10% podaje się 0.1)
- Zdarzenie losujące (podane w dolnym polu) należy koniecznie wykonać co najmniej 10 sekund przed wykonaniem właściwego zdarzenia. W przeciwnym wypadku wynik będzie "zafałszowany". Najlepiej to uczynić od razu po uruchomieniu misji.
- Aby wykonało się jedno (losowo wybrane) zdarzenie, należy wywołać event o nazwie takiej samej jak nazwa identyfikatora. Można to zrobić w *Multiple* wybierając typ *Inne*.

2.6 Wstawianie Sygnalizatorów

2.6

W programie znajduje się wyciąg z wytycznych dla twórców, więc nie wymaga to szerszego omówienia. Nazwę tarczy można podać lub nie, w zależności od tego czy to semafor wjazdowy czy wyjazdowy.

2.7 Zdarzenie Losowe - algorytm YouBy'a

2.7

Potocznie nazywany także *drabinką*. Inny sposób losowania zdarzeń z jednakowym prawdopodobieństwem.

- Po wygenerowaniu należy bezwzględnie zapoznać się z tekstem zawartym w dolnym polu tekstowym i się do niego zastosować.
- Nazwa identyfikatora powinna być unikalna w skali zdarzeń losowych.
- Zdarzenie losujące (podane w dolnym polu) należy koniecznie wykonać co najmniej 10 sekund przed wykonaniem właściwego zdarzenia. W przeciwnym wypadku wynik będzie "zafałszowany". Najlepiej to uczynić od razu po uruchomieniu misji.

2.8 Inne Szablony

2.8

Jeśli scenarzyście zależy na jak najwierniejszym odwzorowaniu symulacji, a nie tylko żeby „podać zielone”, często dla każdej stacji musi powielać te same dziesiątki linijek kodu różniące się od siebie zasadniczo szczegółami jak nazwa semafora czy współrzędne dźwięku. Dlatego też w tej zakładce postanowiłem zawrzeć „szablony” popularnych eventów. Tak, by można było podać tylko kilka danych, a wszystkie „memcele” i inne „moce nieczyste” same się tworzą.

2.8.1 Zgłaszanie się na wjeździe

2.8.1

Można stosować również na manewrach. Odmienny i dość nietypowy, a bynajmniej rzadko spotykany wcześniej sposób wyzwalania zdarzeń przez użytkownika. Nie tylko wjazd na jakiś konkretny tor czy o konkretnej godzinie może spowodować uruchomienie zdarzenia, ale także „zgłoszenie się” do „dyżurnego” poprzez wciśnięcie odpowiedniego klawisza. Po konsultacji z innymi twórcami stwierdziłem, że najlepiej nadaje się do tego klawisz W, który nie jest używany do innych funkcji. Zdarzenie awaryjne musi być wywoływane przez pociąg będący w niewielkiej odległości od semafora – wykona się ono tylko jeśli użytkownik nie wcisnął „W” i po określonym czasie. Jako współrzędne, podajemy współrzędne (mniej-więcej) semafora wjazdowego. Zgłosić się do dyżurnego będzie można jedynie w odległości podanej w polu „Promień zasięgu”. Ze względu na pochylenie terenu, łuki itp. w każdym wypadku trzeba dobierać „na oko” inny promień zasięgu, zwykle około 1000m.

W wersji 1.9 wprowadziłem możliwość generowania dodatkowych zdarzeń awaryjnych. Po pierwsze, zdarzenie awaryjne może wywołać inny event niż ten, który wywołuje wciśnięcie W. Możemy

oczywiście podać ten sam event. Po drugie, zdarzenie awaryjne może zadziałać po jednym z dwóch, losowo wybranym opóźnieniu. Dzięki temu można symulować podanie wjazdu przez dyżurnego zanim nie zdążymy go o to poprosić - oczywiście losowo. W przeciwnym wypadku wjazd poda się po dłuższym czasie - o ile wcześniej się nie zgłosimy. Jak zwykle trzeba wcześniej (najlepiej od razu po uruchomieniu symulacji) uruchomić zdarzenie losujące - jego nazwę wskaże program.

2.8.2 Zmienne

2.8.2

Zmienne to nic innego jak komórki pamięci, z tym że nie wymagają podawania takich rzeczy jak współrzędne czy tor. Dzięki temu użytkownicy, którzy z niewiadomych przyczyn traktują komórki pamięci jako zło wcielone, mogą bez problemu skorzystać ze zmiennych. W Multiple powstał warunek *Zmienna*, przystosowany specjalnie do tych specyficznych komórek pamięci.

2.9 Prosty opis scenariusza w HyperText Markup Language (HTML)

2.9

W tej zakładce użytkownik ma możliwość wygenerowania prostego opisu i wyeksportowania go do pliku HTML otwieranego przeglądarką internetową. Choć domyślnie ustawiono pewne czcionki i kolory, zaleca się wybór innych. Ponadto należy używać najpopularniejszych czcionek: *Times New Roman*, *Arial*, *Verdana*. Jeśli wybierzemy zbyt "egzotyczną", jest możliwość, że użytkownik nie będzie jej posiadał - wtedy zostanie użyta czcionka *Arial*. Tytuł powinien zawierać tylko nazwę scenariusza. W tekście pod obrazkiem można zawrzeć jednozdaniowe streszczenie scenariusza. W głównym tekście powinien się znajdować opis właściwy, a w tekście końcowym ostateczne słowo autora, pozdrowienia, itp. Do wyboru mamy 2 rodzaje tła: kolorowe i obrazkowe. W przypadku wybrania drugiej opcji należy podać nazwę pliku graficznego wraz z rozszerzeniem. Plik ten musi się znajdować w katalogu */files* będącym podrzędnym katalogiem w folderze w opisie. Należy też podać nazwę screena - także z katalogu */files* i z rozszerzeniem. Opis powinien się znajdować w *inne/nazwa-scenariusza*.

Rozdział 3

Pozostałe przydatne informacje

3

Rzeczy, które mogą się przydać przy tworzeniu scenariusza, nie tylko za pomocą generatora. FAQ.

3.1 Korzystanie z podglądu terenu *Rainsted*

3.1

Niejednokrotnie wspomniałem, że należy podać współrzędne (X Y Z). kilka sposobów na ich sprawdzenie, ale moim zdaniem najprościej i najszybciej interesujące nas dane uzyskamy dzięki programowi Rainsted. Kolejne kroki:

- Należy uruchomić w/w program i upewnić się, że tryb pracy jest ustawiony na *specjalny – dla twórców*.
- Wybieramy interesującą nas scenerię i przechodzimy do zakładki *struktura*.
- Wciskamy *Wczytaj wszystkie include*
- W zakładce *debugger* wciskamy najpierw *Połącz tory...*
- ... a później *Podgląd terenu (to nie jest edytor)*
- Z prawej strony wybieramy zakładkę *punkty* - Pojawia nam się tam współrzędna X i Z miejsca na które najedziemy myszką.

Nazwę toru możemy sprawdzić poprzez kliknięcie na dany odcinek. Nazwa pojawi się na belce tytułowej okna podglądu. Pojawi się także informacja o tym, w której linii, którego pliku znajduje się wpis danego toru (szczególnie przydatne, gdy dodajemy wpis eventu do toru). Semaforey są wyświetlane jako czerwone, tarcze ostrzegawcze pomarańczowe, a tarcze manewrowe niebieskie kropczki. Pod nimi znajdują się ich nazwy.

3.2 FAQ

3.2

Odpowiedzi na najczęściej zadawane pytania.

- Po uruchomieniu programu wyskakuje jakiś błąd i aplikacja się wyłącza.
 - Najprawdopodobniej nie posiadasz środowiska uruchomieniowego .NET Framework, lub masz starą wersję. Zainstaluj lub zaktualizuj .NET Framework bądź pobierz wersję programu kompatybilną z Twoją wersją .NET.
- Dlaczego tworząc event multiple mogę podać tylko 8 zdarzeń? Jak wywołać za jednym razem więcej zdarzeń?
 - 8 zdarzeń to ograniczenie w symulatorze, a nie tylko w generatorze. Pisząc *eventy* ręcznie też można użyć tylko 8 zdarzeń. Można jednak stworzyć część eventów w innym *multiple*, i wywołać go w tym „głównym”, wybierając *Inne* i podając nazwę tego eventu.
- Podałem sygnał zezwalający na jazdę składowi AI, a on nie jedzie.
 - Jeśli nie masz zamiaru uruchamiać AI komendą z listy RFC, skład prowadzony przez komputer powinien mieć niezerową prędkość początkową we wpisie trainset (zwykle podaje się 0.1). To pierwszy warunek, aby AI "patrzyło" na semafor. Po drugie, AI "zobaczy" sygnalizator dopiero, gdy na swoim lub którymś z następnych torów znajdzie się event *_sem_info*. Należy więc we wpisie toru koło semafora, analogicznie jak było to wytłumaczone w punkcie Multiple, dodać *event1 nazwa-semafora_sem_info* dla wjazdu z punktu 2 lub *event2 nazwa-semafora_sem_info* dla wjazdu z punktu 1.
- Dopisałem event do toru używając funkcji autodopisywania, a mimo to event się nie uruchamia.
 - Podczas dopisywania eventów może dojść do sytuacji kiedy w torze wcześniej istniał inny event wyzwalany przez wjazd w tym samym kierunku (np. dwa eventy event2...). W takim wypadku należy odnaleźć wadliwy tor i zamienić 2 wpisy eventX... na jeden o innej nazwie, który wyzwała oba eventy znajdujące się w torze.

3.3 Tworzenie misji

3.3

Jeśli dotarłeś aż tu to super :-). Starałem się w dość dokładny i przystępny sposób opisać jak należy korzystać z programu oraz jak się deklaruje poszczególne typy zdarzeń. Jednak to co przedstawiłem do tej pory to nie wszystko. Nadal nie napisałem co zrobić, aby misja pojawiła się w starterze, aby można było ją uruchomić i opublikować na stronie dla szerszego grona użytkowników. W takim razie działamy! ;-)

W pierwszej kolejności należy zdecydować, na jakiej scenerii powstanie scenariusz. Szukamy odpowiedniego pliku *.SCN* z daną scenerią bezpośrednio w folderze *scenery*. Jeśli nie chcemy tracić oryginalnych scenariuszy, plik ten należy skopiować do tego samego katalogu i nadać mu inną nazwę (bez polskich znaków). Następnie otwieramy plik. W zależności od tego jaka to sceneria,

wewnątrz może to wyglądać różnie. Czasami, w pliku tym będzie tylko wpis prowadzonej lokomotywy, czasami będą wszystkie pojazdy na scenerii, a niekiedy będą też wszystkie zdarzenia. Początek pliku będzie jednak wygląd prawie zawsze tak samo (oczywiście opis dla każdej scenerii może a nawet musi być inny):

```
// $n Nowy Świat-Dzierżawa
// $d Linia pasażersko-towarowa. Długość: 60,3km
// $d Główne stacje: Nowy Świat, Krosowo, Rabinów, Dzierżawa.
// $d
// $d © 2004-2005 by Hunter. Kopiowanie materiałów bądź rozprowadzanie ich
bez wiedzy autora zakazane.
// $i 1546_sm42.jpg
// $g EU07-424 516 460

sky cgskj_sunset094.t3d endsky
atmo 0 0 0 1000 2000 0.086 0.086 0.086 endatmo
light -500 70 -3000 0.157 0.157 0.157 0.996 0.769 0.259 0.247 0.412 0.502 endlight
```

Na początku jest opis, który jest widoczny w Rainstedzie, zaraz za nim plik graficzny ze *scenery/images*, także wyświetlany w starterze. Resztą nie musisz się przejmować. Później najprawdopodobniej spotkasz mniej więcej taki ciąg linijek:

```
include linia546/linia546_sm42_dane.scm end
include linia546/linia546_trakcja.scm end
include linia546/linia546_sm42_events.inc end
```

Jest to nic innego, jak deklaracja plików include. Plik scenarii powinien mieć wpisy eventów, torów, budynków i wiele innych. Nie ma większego sensu umieszczanie wszystkiego jednym plikiem. Dlatego też, w tym znajdującym się bezpośrednio katalogu scenery są tylko podstawowe informacje, a reszta znajduje się w innych plikach – wspomnianych includach. Dzięki temu, że ich nazwy są sensowne, można się łatwy sposób domyślić za co każdy z nich odpowiada.

W przypadku jak powyżej, to oczywiście, że zostawiamy sieć trakcyjną i tory (nazwa dane, gdyż zawiera także teren).

Jest kilka sposobów, żeby sprawdzić co dany plik zawiera. Po pierwsze, wielkość – pliki z torami mają z reguły kilka lub nawet kilkanaście megabajtów. Po drugie, jak już wspomniałem – nazwa. Po trzecie – zawartość, ale trzeba się na tym choć trochę znać.

Eventy jednak nie są nam potrzebne – tworzymy przecież własne. Dlatego linijkę z plikiem z eventami można usunąć. W jej miejsce wstawiamy na wzór tejsze linijki, wpis do własnego pliku include z eventami. Oczywiście ten plik należy najpierw utworzyć – można to zrobić za pomocą programu.

Najprawdopodobniej od zaraz za wpisami include, aż do słowa kluczowego *FirstInit* będą albo eventy, albo nie ma nic. Jeśli są to eventy, należy je usunąć.

Słowo *FirstInit* należy obowiązkowo zostawić – po nim znajdują się wpisy taboru na scenarii. Skoro tworzymy nowy scenariusz, najprawdopodobniej pojazdy na scenarii będą też inne – wtedy należy skasować wszystko. Można jednak niektóre z nich zostawić, nie robi to większego problemu. Musimy jednak pamiętać o tym, aby był tam co najmniej wpis – naszego pojazdu.

Tworzenie wpisu pojazdu jest bardzo proste. Jednak w przypadku problemów zajrzyj na forum www.eu07.pl/forum.

Została jeszcze jedna ważna rzecz – usunięcie wpisów eventów z torów. Pamiętasz, jak pisałem o tym, że gdy tworzymy *multiple*, trzeba jeszcze dopisać parę rzeczy do wpisów torów? Teraz musimy pousuwać pozostałości po starym scenariuszu w postaci takich wpisów. Można robić to ręcznie albo można też użyć gotowych narzędzi. *Generator eventów* takiej opcji, można jednak użyć

innych aplikacji, jak np. *EventoUsuwacz @Pawilonka*. Osobiście polecam zmienić konfigurację tego programu na następującą:

```
[usun]
eventall2
eventall1
eventall0
event2
event1
event0
```

```
[wyjatki]
sem_info
zamykaj
otwieraj
rez
sbl
s1
shp
S1
SHP
```

Taki pozbawiony eventów plik przenosimy do podkatalogu w *scenery* z daną scenerią, zmieniamy jego nazwę na jednoznacznie go identyfikującą i zmieniamy wpis include – tak, aby nazwa się zgadzała.

Ufff... Dotarliśmy do końca. W programie opisałem wiele funkcji, jednak początkującemu wystarczą podstawowe informacje dotyczące *event multiple*. Za jego pomocą można naprawdę zrobić całą i w pełni działającą misję. Ponadto nowa wersja zawiera sporo uproszczeń i nie trzeba się wcale znać na komórkach pamięci, żeby zrobić np. zdarzenia losowe, czy blokadę przed ponownym wykonaniem zdarzenia.

Wszelkie problemy związane z obsługą programu proszę zgłaszać na forum symulatora MaSzy-na.

Miłego tworzenia nowych scenariuszy do MaSzyzny życzy SKP ;)