

Martin Gladis Gr I

Grafika komputerowa I Projekt nr I

Opis kodu

Projekt polega na zaprezentowaniu animacji za pomocą znacznika `<canvas>` i podpiętych skryptów z funkcjami oraz bibliotekę „*dat.gui*”. Animacja prezentuje samochody poruszające się po ulicy.

Plik HTML zawiera wyłącznie podpięte pliki „*dat.gui.min.js*” i „*script.js*” który zawiera wszelkie funkcje do rysowania animacji oraz do znacznika „*body*”, w którym jest odnośnik wczytujący do funkcji „*draw*”, dodano znacznik „*p*”, który został wyśrodkowany, a zaś do niego wstawiono znacznik „*canvas*”, któremu nadano wysokość, szerokość oraz kolor.

W pliku „*script.js*” Zadeklarowano wszystkie funkcje niezbędne do narysowania poszczególnych figur geometrycznych. Na początku zadeklarowano dwa obiekty (Circle oraz Rect), w którym utworzono punkty X oraz Y o wartościach 0, promień (czy też wysokość i szerokość dla „Rect”). O wartościach 1 oraz kolor, który otrzymuje wartość z argumentu obiektu. Następnie w ramach obiektu utworzono funkcje „*display()*” oraz „*setReferencePoint()*”, w których kolejno wyświetlamy dany obiekt oraz ustalamy nowe punkty *x0* oraz *y0*.

Kolejne linijki kodu przedstawiają funkcje o nazwach „*drawRect()*” oraz „*drawCircle()*”. Służą one do tworzenia poszczególnych obiektów, nadawaniu im zmodyfikowaną szerokość i wysokość, na nadaniu nowych punktów początkowych, oraz na wywołaniu funkcji o tej samej nazwie, jednak z dopisanym przedrostkiem „*re-*”. Funkcje „*reDrawRect()*” oraz „*reDrawCircle()*” służą do implementacji wcześniej zainicjowanych danych w macierzy transformacji oraz na namalowaniu danego obrazka ze zmodyfikowanej wartości.

W następnych liniach kodu pojawiają się deklaracje funkcji „*myControls*”, która posłuży później do inicjacji „*dat.GUI()*” oraz zmiennych globalnych. Funkcja „*animation()*” służy do wywołania tych wszystkich funkcji rysujących figury z odpowiednimi parametrami.

Na początku zostaje użyta funkcja „*requestAnimationFrame()*”, która przyjmuje funkcję „*animation*” jako argument. Funkcja ta służy to ciągłego wywołania funkcji w celu stworzenia animacji. Następnie jest użyta funkcja czyszcząca całego *canvas'a*, w celu ponownego narysowania kształtów w następnej klatce. W następnych liniach kodu są inicjalizacji zmiennych (*width*, *height*, *x*, *y*) oraz wywołujemy funkcję „*drawRect()*” a jako argumenty są dane powyższe zmienne oraz kolor w postaci szesnastkowej w celu narysowania prostokąta, który ma być drogą. Następnie po zmianie szerokości, pozycji *x* oraz wysokości program maluje pasy w pętli (*posy[i]* jako punkt *y*, który się zmienia w ramach z każdą iteracją w celu narysowania więcej pasów). Następnie do zmiennej jest dodawana wartość „*carRight*”. Dzięki czemu użytkownik może sam nadać tą wartość za pomocą suwaka. W następnej linijce jest warunek, w którym jak pas przekroczy *canvas'a* to wędruje on na samą górę, poza zasięg widoku by mógł robić wrażenie, że nowy pas się namalował.

W następnym kroku rysuje się samochód w formie prostokąta w kolorze czerwonym w pozycji równej pozycji $\frac{3}{4}$ szerokości drogi odjętej od połowy szerokości samochodu. Następnie są dorysowywane szyby na pozycjach względnie do samochodu również w postaci prostokątów jednak w innym kolorze. Światła w samochodzie są robione z półelipsy (funkcja *drawCircle* z argumentami *x*, *y*, *r*, koloru, oraz kątów rozpoczęcia i zakończenia rysunków) na pozycji kolejno na $\frac{1}{4}$ oraz $\frac{3}{4}$ szerokości samochodu. Drugi

samochód jest robiony analogicznie jak poprzedni, jednak jest robione „do góry nogami”. Oraz pozycja y się zmienia wraz z kolejną klatką „tak samo jak z pasami”. Drzewo jest to połączenie prostokąta jako pnia oraz elipsy (zmodyfikowane koło) jako liści. Pozycja x jest to liczba losowa z zakresu od 30 do 180 i jest losowana z za każdym razem jak się drzewo chowa w dół i dzięki czemu jak się znów pojawia jest w innej pozycji. W tym przypadku pozycja y się również zmienia z każdą nową klatką.

Droga boczna jest również ruchomy elementem. Rysowana jest podobnie jak ulica, jednak pozycja x jest równa sumie pozycji x głównej drogi oraz jej szerokości. Pasy na tej drodze są w tym przypadku ciągłe podwójne i znajdują się mniej więcej w połowie drogi. W miejscu samochodu przy bocznej drodze jest warunek w którym licznik, który jest iterowany wcześniej, jest równy 1 i w takim przypadku jest rysowany samochód. Jest on tych samych rozmiarów co poprzednie dwa auta, jednak wartości wysokości i szerokości są zamienione w celu uzyskania samochodu stojącego bokiem. Jest os rysowany względem tej bocznej uliczki żeby samochód się poruszał razem z tą uliczką.

Funkcja „draw()” jest to funkcja która się wczytuje zaraz po załadowaniu strony. Znajduje się w niej inicjalizacja „myGUI”, któremu przypisujemy obiekt „dat.GUI”. Następnie do obiektu jest utworzony katalog z dwoma suwakami, które mają dostosowywać prędkość samochodów, a następnie jest on otwierany. Na samym końcu jest utworzony uchwyt do obiektu canvas, jest utworzony kontekst, który renderuje dwuwymiarowo oraz wywołanie funkcji animation.

Do uruchomienia jest wymagana przeglądarka wspierająca HTML 5. Wystarczy uruchomić plik „index.html”. Animacja powinna się sama załadować.