Projekt Concurrent ZOO

Autorzy:
Kinga Florek, Michał Worsowicz

Programowanie współbieżne i rozproszone

Wydział Elektrotechniki, Automatyki, Informatyki i Inżynierii Biomedycznej

1 Cel programu

Celem naszego projektu Concurrent ZOO jest stworzenie okienkowej aplikacji symulującej działanie ogrodu zoologicznego. Zakłada ona, że czynności takie jak: kupowanie biletów, wstęp do zoo, oglądanie zwierząt, karmienie zwierząt itd. są współbieżnymi procesami. Dla uproszczenia przyjęliśmy, że zwierzęta występujące w naszym zoo to pandy i lisy. Program pozwala na śledzenie tego co dzieje się w ZOO - zarówno od strony odwiedzającego jak i zwierząt.

2 Opis instalacji / kompilacji / uruchomienia

Do implementacji GUI wybraliśmy bibliotekę GtkAda, jednak po przeczytaniu dokumentacji okazało się, że GTK+ nie współgra z wątkami (Taskami) w Adzie. Na stronie Tasking with GtkAda możemy przeczytać: *"Note that Gtk+ under Windows does not interact properly with threads, so the only safe approach under this operating system is to perform all your Gtk+ calls in the same task."* Chcąc mimo to wykorzystać tą bibliotekę do stworzenia GUI aktualizowanego z poziomu różnych tasków, znaleźliśmy dodatkową open-sourcową bibliotkę, która mimo to umożliwa współgranie GtkAda z Taskami. Cała dokumentacja dodatkowej użytej przez nas biblioteki dostępna jest na stronie: GtkAda Contributions, a poniżej opiszemy w skrócie konieczne dodatkowe kroki, aby skompilować i uruchomić nasz projekt w GNAT Studio. Cały opis odnosi się do instalacji na systemie Windows. Konieczne jest posiadanie GNAT Studio.

Na samym początku konieczne jest pobranie GtkAda ze strony AdaCore Download. Należy wybrać platformę x86 Windows (64 bits), wtedy pojawi się GtkAda, a następne pobrać i uruchomić plik **gtkada-2020-x86_64-windows-bin.exe**.

W kolejnym kroku należy wejść na stronę GtkAda Contributions i pobrać skompresowany folder **gtkada_contributions_3_28.tgz** znajdujący się przy Source distribution (any platform), a następnie wypakować go do folderu include w miejscu gdzie zainstalowany został GNAT Studio. Przykładowa poprawna ścieżka: `C:\GNAT \2020\include \gtkada _contributions_3_28`

Ostatnim krokiem jest otworzenie naszego projektu w GNAT Studio i edycja

pliku projeku **concurrent_zoo.gpr**. Jeżeli ten plik nie jest widoczny należy kliknąć na główny folder projektu Concurret_Zoo -> Project -> Edit source file. Po otowrzeniu pliku należy w pierwszej linjce podać właściwą ścieżkę do **gtkada_contribiutions.gpr**. Przykładowo: `with "C:\GNAT \2020\ include \gtkada _contributions_3_28\gtkada _contributions.gpr";`

Tak przygotowany projekt można bez problemu skompilować i uruchomić w GNAT Studio.

3 Opis struktury programu

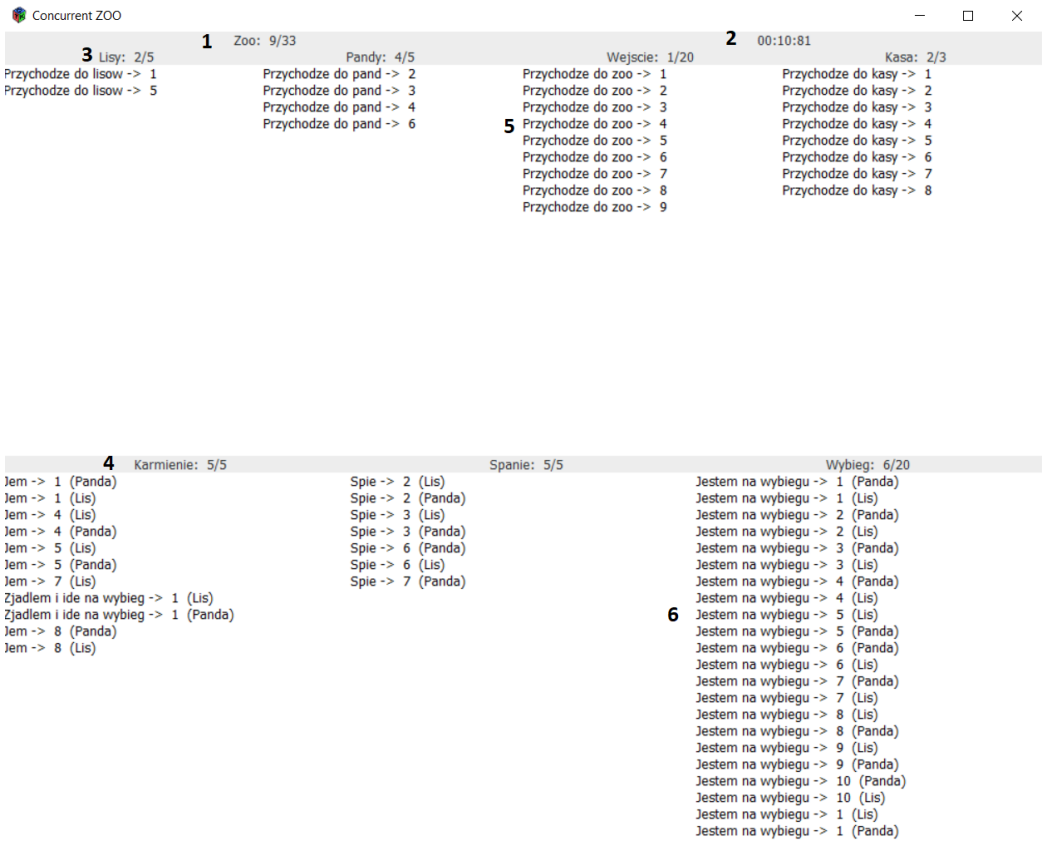
Program składa się z 18 zadań wpółbieżnych (tasków), które to zdefiniowane zostały na podstawie zgeneralizowanych typów tasków **ScrollTaskPeople** oraz **ScrollTaskAnimals**. W zależności od zmiennych z jakimi zostały zainicjalizowane pełnią różne funkcje związane z czynnościami wykonywanymi w ZOO, takimi jak zakup biletów, przemieszczanie się między wybiegami, jedzenie oraz spanie. Zadania te komunikują się za pomocą buforów (procedura **Buf**), które to definiują ile osób/zwierząt może wykonywać daną czynność równocześnie, a także ile czasu ma ona zająć. Ograniczeniem ilości zajmuje się zmienna **Counter**, która to określa czy do danego bufora można dopisać kolejny numer identyfikacyjny oraz czy można coś z niego wyjąć. W celu określenia czy dana czynność została wykonana, a więc minął odpowiednio długi czas od jej rozpoczęcia, wykorzystujemy tablicę **Delayx** przechowującą wartości czasu, od których identyfikator może opuścić bufor.

Procesem wypisywania logów w interfejsie zajmują się procedury **HandlerPeople** oraz **HandlerAnimals**, które to przyjmują od zadań informację o tym kto wykonywał jaką czynność aby następnie wypisać ją w wyznaczonym okienku. Aktualizacją liczników osób/zwierząt wykonujących daną czynność zajmują się procedury ***Nazwa*Update**, które to wywoływane są po wszelkich zmianach wpływających na ich wartość, aby przesłać do interfejsu zaktualizowaną liczbę. Osobną kwestię stanowi za to licznik czasu, który wykorzystuje zmienną **TimeCounter** do wypisania czasu jaki upłynął od startu programu w przyjemnej dla oka formie.

Sam interfejs powstał za pomocą bibloteki **GtkAda**, przy użyciu której zdefiniowane zostaje widoczne dla uzytkownika okno oraz jego elementy - **Label**'e odpowiedzialne za wyświetlanie liczników oraz **Scroll**'e, w których wypisywane są zdarzenia.

4 Krótka instrukcja obsługi

Po uruchomieniu naszego projektu w GNAT Studio możemy zauważyć, że do ZOO zaczynają przychodzić odwiedzający. Początkowo ustawiają się w kolejce do kas i następnie przechodzą losowo do pand lub lisów. Z tego miejsca odwiedzający może opuścić ZOO lub też udać się do drugiego wybiegu. Zwierzęta za to znajdują się na wybiegu, mając opcje, oczywiście tylko gdy istnieje wolne miejsce, aby pójść spać lub jeść. Cały projekt jest symulacją ZOO, która pozwala na obserwację tego co się dzieje w danym momencie poprzez analizę logów w konretnych miejscach w ZOO oraz analizę liczników. Po uruchomieniu programu użytkownik spotyka się z następującym interfejsem:



Gdzie:

1. Licznik osób w ZOO,
2. Licznik czasu,
3. Liczniki ilości osób w danych częściach ZOO,
4. Liczniki ilości zwierząt przy danych czynnościach,
5. Pole tekstowe ze zdarzeniami ludzi,
6. Pole tekstowe ze zdarzeniami zwierząt.

5 Możliwe kierunki rozbudowy

Oczywistym możliwym kierunkiem rozbudowy jest dodanie większej ilości różnych gatunków zwierząt do naszego ZOO czy zwiększenie dostępnych akcji dla zwiedzających (na przykład korzystanie z restauracji w ZOO). Ponadto obok akcji dla zwiedzających i zwierząt, można dodać pracowników ZOO i możliwe akcje dla nich (na przykład sprzątanie). W ten sposób możliwa by była bardziej wiarygodna i rzeczywista symulacja ZOO. Innym pomysłem jest również dodanie w programie przycisków, pozwalających na interakcję użytkownika z symulacją. Użytkownik mógłby sterować ilością zwiedzających (na przykład dodawać nowych) lub zmieniać czas spania/ karmienia zwierząt.