

Uniwersytet Wrocławski
Instytut Informatyki
Zespołowy projekt programistyczny

GameMaker

Dokumentacja

Kamil Kwaśny, Michał Lech, Paweł Krzywodajć

Wrocław, 11 czerwca 2014

Spis treści

1. Wstęp	2
1.1. GameMaker Editor	2
1.2. GameMaker Client	2
2. Cele projektu	2
3. Zagadnienia techniczne	3
4. Praca w zespole	3
5. Uwaga	3
6. Dokumentacja techniczna	3
6.1. Struktura projektu	3
6.2. GUI	3
6.3. Biblioteka 2D	4
6.4. Klasa <code>Game</code>	4
6.5. Klasa <code>Level</code>	4
6.6. Pliki	4
6.7. Podsumowanie	4
7. Możliwości rozwoju projektu	4

1. Wstęp

GameMaker to projekt złożony z dwóch aplikacji: **GameMaker Editor** oraz **GameMaker Client**. Oba programy będą aplikacjami desktopowymi. Komunikacja z użytkownikiem będzie przebiegać poprzez graficzny interfejs użytkownika (ang. *GUI*), który udostępniać będzie całą funkcjonalność aplikacji.

1.1. GameMaker Editor

GameMaker Editor ma być środowiskiem, umożliwiającym tworzenie prostych komputerowych gier platformowych ¹. Aplikacja ta będzie udostępniać szereg opcji, które pozwolą na proste, szybkie i przyjemne tworzenie popularnych i grywalnych platformówek. Użytkownik programu będzie mógł na przykład stworzyć nową grę, dodawać do niej poziomy, ustawiać tła poziomów, dodawać obiekty do poziomów (np. przeszkody lub przeciwników) itp. W zależności od ilości dostępnego czasu, zakres funkcjonalności programu może być rozbudowany i poszerzony.

1.2. GameMaker Client

GameMaker Client będzie programem, który pozwoli użytkownikom grać w gry stworzone w środowisku **GameMaker Editor**. Użytkownik programu będzie mógł używać go nie tylko do grania w stworzoną wcześniej grę, gdyż aplikacja ta może być również przydatna w fazie testowania tworzonego projektu. Podobnie jak we wcześniejszej aplikacji, tak i tutaj, zakres funkcjonalności będzie uzależniony od czasu, który autorzy będą mogli poświęcić na utworzenie projektu.

2. Cele projektu

Można wymienić kilka głównych celów tworzonego projektu:

- stworzenie aplikacji **GameMaker Editor** i **GameMaker Client**,
- zaprojektowanie i zaimplementowanie interfejsów graficznych dla obu programów,
- rozbudowanie funkcjonalności obu programów,
- rzetelne przetestowanie obu aplikacji,
- utworzenie oraz prezentacja prostej gry platformowej, przy użyciu stworzonego wcześniej oprogramowania.

Największym sukcesem dla autorów byłoby wykorzystanie aplikacji edytora do utworzenia projektu na kształt znanej wszystkim gry *Super Mario Bros* ².

¹Patrz: http://pl.wikipedia.org/wiki/Komputerowa_gra_platformowa

²Patrz: http://en.wikipedia.org/wiki/Super_Mario_Bros.

3. Zagadnienia techniczne

Obie, opisane wcześniej, aplikacje napisane będą w języku **Java**, przy wykorzystaniu biblioteki graficznej 2D (wybór biblioteki jest w fazie dyskusji). Do tworzenia graficznych interfejsów, wykorzystany będzie prawdopodobnie **Swing**. Autorzy programu będą korzystać ze środowiska programistycznego **NetBeans**.

4. Praca w zespole

Aby usprawnić pracę w zespole, podczas tworzenia projektu wykorzystany będzie system kontroli wersji **git**. Poniżej znajduje się odnośnik do repozytorium, w którym umieszczone będą kody źródłowe obu aplikacji oraz dokumentacja projektu.

<https://github.com/lechoPl/GameMaker>

Żeby praca nad projektem mogła być prowadzona równolegle przez wszystkich autorów, zdecydowano się na cotygodniowe spotkania organizacyjne, podczas których odpowiednie zadania będą przyporządkowywane odpowiednim osobom. Poniżej znajduje się odnośnik do tablicy, na której można znaleźć obecne, nowe oraz ukończone zadania.

<https://trello.com/b/jF8rshEY/gamemaker>

5. Uwaga

Treść i wygląd niniejszego dokumentu mogą zmieniać się podczas tworzenia projektu. Obecna wersja jest zawsze dostępna w repozytorium, do którego odnośnik podano w sekcji 4.

6. Dokumentacja techniczna

W poniższej sekcji zawarto całą dokumentację techniczną projektu **GameMaker**.

6.1. Struktura projektu

Projekt składa się właściwie z trzech projektów środowiska programistycznego **NetBeans**, zaimplementowanych w języku **Java**: **GameMaker Editor**, **GameMaker Client** oraz **GameMaker Library**. Pierwsze dwa z nich są aplikacjami desktopowymi, których funkcjonalność opisana jest w sekcji 1 oraz instrukcji użytkownika dołączonej do projektu.

Projekt **GameMaker Library** jest biblioteką, która udostępnia odpowiedni kod dla dwóch pozostałych projektów. Został on utworzony, ponieważ **Editor** oraz **Client** w wielu miejscach muszą korzystać z tych samych klas, fragmentów kodu i funkcjonalności.

6.2. GUI

Graficzny interfejs użytkownika w aplikacjach **Editor** oraz **Client** został w całości wykonany przy użyciu bibliotek języka **Java**: **AWT** oraz **Swing**.

6.3. Biblioteka 2D

W fazie tworzenia projektu zrezygnowano z wykorzystania zewnętrznej biblioteki 2D. Cała obsługa obrazków, kształtów, animacji, kolizji itd. została zaimplementowana przy użyciu biblioteki `Swing` i możliwości klasy `JPanel`.

6.4. Klasa `Game`

Klasa `Game` jest jedną z najważniejszych klas utworzonego projektu. Zawiera ona informacje dotyczące zarówno struktury gry (poziomy, rozmiary, nazwy itd.), jak i jej zasobów (obiekty, obrazki, dźwięki itd.). Klasa gry zawiera również kontroler gracza, który wykorzystywany jest podczas gry.

6.5. Klasa `Level`

Klasa `Level` przechowuje wszystkie informacje istotne dla konkretnego poziomu: jego wysokość, szerokość, nazwę, tło, gracza oraz obiekty (przeszkody, przeciwników itp.). Klasa poziomu zawiera również bardzo ważne metody, które używane są do obsługi wyświetlania gry (`render`), fizyki (`update`) oraz wyszukiwania kolizji (`checkCollision`).

6.6. Pliki

Aplikacje **GameMaker Editor** oraz **GameMaker Client** obsługują pliki z rozszerzeniem `.gmf`. Plik taki jest w rzeczywistości archiwum `zip`, w skład którego wchodzi zserializowane obiekty klas `Game` oraz `Level`, a także zasoby używane w grze (np. obrazki, animacje, dźwięki). Dzięki takiemu rozwiązaniu, projekt gry tworzonej w edytorze można w prosty sposób zapisać do pojedynczego pliku, a następnie otworzyć w aplikacji klienckiej. Zastosowania archiwizacji upraszcza również zdecydowanie przesyłanie/wymianę projektów gier pomiędzy klientami aplikacji **GameMaker**.

6.7. Podsumowanie

Niniejsza dokumentacja zawiera wyłącznie najważniejsze i najistotniejsze zagadnienia techniczne projektu zespołowego. Wiele kwestii pominięto w tym dokumencie, gdyż są one omówione w instrukcji użytkownika, która również jest częścią projektu.

7. Możliwości rozwoju projektu

Prace nad projektem **GameMaker** zakończyły się. Nie oznacza to jednak, że nie można go dalej rozszerzać. Autorzy projektu mają przynajmniej kilka pomysłów na jego rozwój. Wśród rzeczy, które mogłyby usprawnić i upiększyć działanie programów **GameMaker Editor** oraz **GameMaker Client**, znajdują się:

- dodanie możliwości ustawiania ścieżki dźwiękowej gry,
- dodanie możliwości ustawiania efektów dźwiękowych dla różnych akcji w grze,
- usprawnienie działania niektórych elementów interfejsu użytkownika (np. zapamiętywanie otwartych folderów drzewa struktury),

- dodanie możliwości tworzenia i modyfikacji ekranów (np. ekranu menu, ekranu końca gry, ekranu rozpoczynającego poziom itd.),
- zwiększenie kontroli użytkownika nad akcjami w grze (np. dodanie języka skryptowego, za pomocą którego użytkownik mógłby definiować zachowanie kreatur),
- dodanie możliwości walki z kreaturami na odległość (np. posiadania broni lub możliwości rzucania zaklęć),
- dodanie możliwości tworzenia gier wieloosobowych,
- dodanie możliwości ustawiania w grze obiektów - bonusów, które zmieniałyby działanie gry.