Binary Land

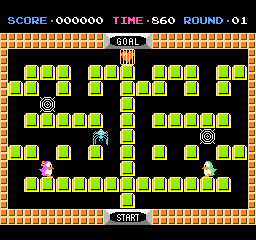
Programowanie w Java - projekt.

Autor: Kamil Pastwa

Prowadzący: mgr inż. Grzegorz Górecki

Opis

Projekt Binary Land to dwuwymiarowa gra wideo. Jest to remake oryginalnej gry z 1983 roku wydanej na platformę MSX i inne, później na NES. Opis oryginału znajduje się pod tym linkiem:

http://en.wikipedia.org/wiki/Binary\_Land  


Gra nie ma być w założeniu wierną kopią oryginału, tekstury, bohaterowie, oraz mapy być może zostaną zmienione. W moim projekcie wykorzystam pomysł na grę. Gracz steruje jednym z bohaterów (na przykład zielonym). Podczas gdy gracz idzie zieloną postacią w prawo, to drugi bohater- różowy idzie w lewo i na odwrót- gdy gracz skręca w lewo, to różowa postać w prawo. Gracz wygrywa rundę gdy obie postacie spotkają się przy sercu na górze planszy. W tym przeszkadzają im rozstawione pajęczyny oraz chodzące pająki, które unieruchamiają postacie rzucając na nie sieć. Gracz przegrywa rundę gdy czas się skończy, gdy obydwie postacie będą unieruchomione w tym samym czasie, lub gdy któraś postać zostanie złapana przez pająka.

Harmonogram

1. **Stworzenie podstawy gry przy użyciu The Lightweight Java Game Library (LWJGL).**

* Stworzenie funkcji wyświetlającej mapę, oraz graczy. (zaj. nr 2) 17.03.2014
* Konieczne będzie skorzystanie z wczytywania mapy z pliku tekstowego.
* Obsługa poruszania się (za pomocą strzałek klawiatury). (zaj. nr 3) 24.03.2014
* Obsługa wczytywania tekstur. (zaj nr 2) 17.03.2014
* Logika animacji obiektów na planszy.(zaj nr 6) 14.04.2014
* Dodanie Menu gry (w Swing) (zaj nr 5) 7.04.2014

1. **Zapis i odczyt plików.**

* Wczytywanie mapy gry z pliku tekstowego. (zaj. nr 3) 24.03.2014
* Zapisywanie lokalnej tablicy najlepszych wyników.
* Wczytywanie muzyki w tle, oraz dźwięków gry.
* Zapisywanie historii gier do pliku log.

1. **Współbieżność: wątki, operacje atomowe, itp.**

* Obliczanie pozycji gracza 1, gracza 2, oraz pająków w osobnych wątkach.
* Zapewnienie nieprzerwanego działania aplikacji podczas pobierania mapy dzięki użyciu wątków.

1. **Bazy danych: ORM (np. Hibernate) lub JDBC dla: MySQL, PostgreSQL lub SQLite\*.**

* Dodanie do gry ekranu logowania.
* Zaprojektowanie bazy danych użytkowników, połączenie z bazą za pomocą JDBC.
* Zapisywanie najlepszych wyników (w przypadku zalogowania) w bazie danych.
* Dodanie opcji wyświetlania najlepszych wyników.

1. **Komunikacja sieciowa: sockety lub RMI.**

* Synchronizacja bazy danych użytkowników z danymi na serwerze.
* Synchronizacja bazy najlepszych wyników z danymi na serwerze.
* Pobieranie dodatkowych map z serwera.
* Sprawdzenie aktualizacji gry. (?)

1. **Zaproponowane przez studenta (np. wzorce projektowe).**

* Stworzenie graficznego edytora map. (zaj. nr 4) 31.03.2014
* Opracowanie mechanizmu zarządzania kolizjami (zaj. nr 4) 31.03.2014
* Połączenie gry w LWJGL z menu SWING.(zaj. nr 4) 31.03.2014