



Игровые решения на Java

Андрей Гребнев

Sun Campus Ambassador
Andrey.Grebnev@Sun.COM



План

- Проект DarkStar и основанные на нем Wonderland, MPK20,
- Game API в MIDP 2.0.
- Mobile 3D Graphics API

Видео игры

- 28 миллиардов долларов – рынок видео игр.
- Рынок видео игр побил музыку, книги и другие развлечения, кроме фильмов.
- Растет рынок игр на архитектуре клиент-сервер т.е. multi-player.
- Sun Microsystems имеет виртуальное представительство в Second Life.

Проект Darkstar

- Darkstar – 100% Java бесплатный с открытым кодом игровой интернет сервер.
- Клиентские интерфейсы: JSE, JME, C++ (Win32 + PlayStation).
- Инфраструктура (основа) для построения асинхронных, параллельных многопользовательских игр.
- Абстрактный протокол на базе TCP, UDP или HTTP.
- Поддержка транзакций, автоматической персистенции.
- Масштабируемость (кластеризация).
- Надежность 5 девяток. Без остановки игры можно добавлять/удалять сервера, исправлять серверный код.

Sun Game Server (SGS)

- Однопоточная модель программирования.
- Событийная модель программирования.
- Game Logic Objects (GLOS), например: меч, монстр, комната и пр.
- Персистентное хранение GLOS.
- Коммуникация: механизм публикации и подписки.

Wonderland & MPK20 на Darkstar



Рабочие пространства людей



Рабочие пространства команд



Рабочие пространства компаний



Аудио и видео



Работа с приложениями



Работа с документами



Game API в CLDC 1.0 и MIDP 1.0

- Отсутствие арифметики с плавающей точкой.
- Нет поддержки аудио, только простые системные сигналы.
- Приложение не может читать/писать отдельные пиксели.
- Класс Graphics не поддерживает прозрачность.
- Коммуникационный протокол только HTTP.

Canvas

```
public class MyCanvas extends Canvas implements Runnable {  
  
    public void run() {  
        while(true) {  
            repaint(); // update the game state  
        }  
  
        public void paint(Graphics g) {  
            // code for painting  
        }  
  
        protected void keyPressed(int keyCode) {  
            // respond to key events  
        }  
    }  
}
```

Game API в CLDC 1.1 и MIDP 2.0

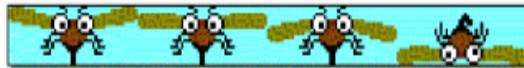
- `javax.microedition.lcdui.game`
- `GameCanvas` – буферизация и запрос состояний клавиш
- `Layer` - положение, размер, видимость
 - > `Sprite` – фреймы, трансформация (флип, поворот), контроль коллизий. Для персонажей.
 - > `TiledLayer` – таблица картинок. Для фона.
- `LayerManager` – автоматизация процесса рендеринга, концепция «окна».

GameCanvas

```
public class MyCanvas extends GameCanvas implements Runnable {
```

```
    public void run() {  
        Graphics g = getGraphics();  
        while(true) {  
            // update the game state  
            // ...  
            int k = getKeyStates();  
            // respond to key events  
            flushGraphics();  
        }  
    }  
}
```

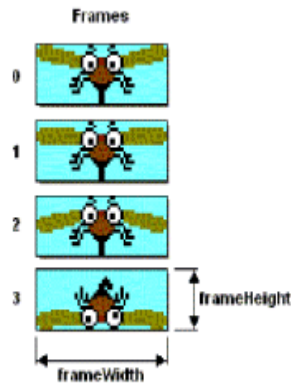
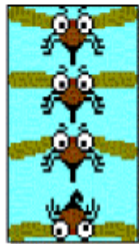
Sprite



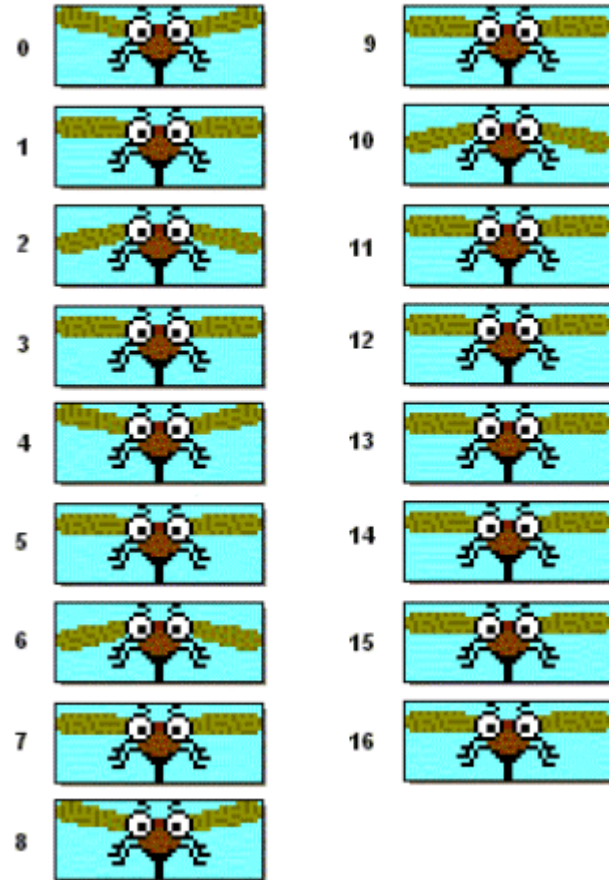
OR



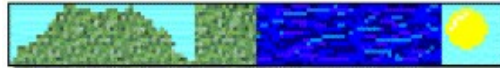
OR



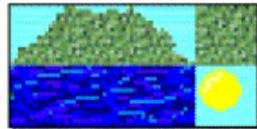
0	1	2	1	0	1	2	1	0	1	2	1	1	1	1	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---



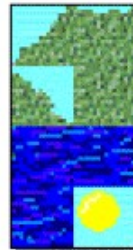
TiledLayer



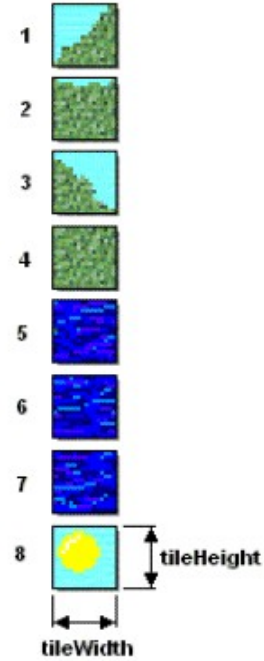
OR



OR



Tiles

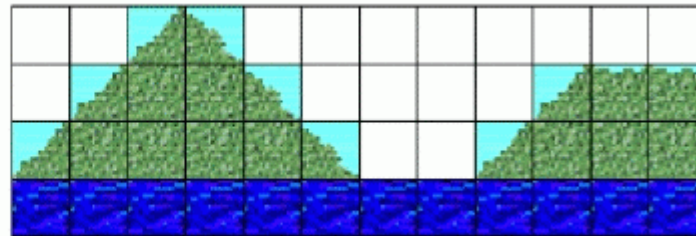


Cells

0	0	1	3	0	0	0	0	0	0	0	0
0	1	4	4	3	0	0	0	0	1	2	2
1	4	4	4	4	3	0	0	1	4	4	4
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1

Animated Tiles

-1 = 5



Mobile 3D Graphics API for JME

- Опциональный пакет JSR 184
- Не обязательна аппаратная поддержка 3D графики и вычислений с плавающей точкой
- high-level - retained mode
- low-level - immediate mode соответствует OpenGL for Embedded Systems
- Требование: 150Кб памяти, импорт сцен, текстур
- `java.microedition.m3g`

Sun & УдГУ
<http://sun.udsu.ru>

Спасибо за внимание

Андрей Гребнев
Andrey.Grebnev@Sun.COM

