Depto. Informática, UTFSM

Informe de Avance Transchantiago

Seminario de Desarrollo de Software

Erich Raddatz Altamirano	2273052-5
José Miguel Tobar	2273012-6
Francisco Riveros Escobar	2273036-3
Marcelo Salazar Rodríguez	2273045-2

1. Índice

2. Introducción

Transchantiago es un juego de carreras en tercera persona para dispositivos móviles, donde tomas el control de un bus para realizar tu recorrido a tiempo, sorteando diversos obstáculos y enemigos que intentaran evitar que logres tu objetivo.

2.1. Trasfondo

Transchantiago es un proyecto de un juego de carreras adaptado a la realidad nacional que se vive en chile, por lo que el jugador se identificara con el juego y está familiarizado con los conceptos manejados.

Este proyecto se realizara en el contexto del ramo "Seminario de desarrollo de software", por lo que se tiene fines académicos, no comerciales.

Esta aplicación posee cierta semejanza al popular juego Crazy Taxi (Crazy Taxi, 2000), en donde el jugador debe recoger un pasajero y llevarlo a su destino en un tiempo determinado. La principal diferencia es que en **Transchantiago** el recorrido viene dado, mientras que en *Crazy Taxi* el jugador elige el que estime conveniente. Además, en *Crazy Taxi* el objetivo del jugador es recolectar la mayor cantidad de dinero, antes que se acabe el tiempo, en **Transchantiago** el objetivo es realizar un recorrido dado obteniendo una recaudación mínima solicitada.

2.2. Propósito

El propósito de este documento es entregar un avance del desarrollo del proyecto **Transchantiago**.

2.3. Resumen Ejecutivo

Basado en un problema de acontecer nacional con el sistema de transporte público de *Santiago de Chile*, además de la petición del cliente/profesor Cesar Henandez de hacer un videojuego, nace el proyecto de **Transchantiago**. Este es un juego en 2D desarrollado en ambiente Java J2ME/Polish para dispositivos móviles, que consiste en

tomar el control de un bus para realizar tu recorrido a tiempo, sorteando diversos obstáculos y antisociales que intentarán evitar que logres tu objetivo.

Para el éxito del proyecto es necesario un buen diseño y especificación de requerimientos, es por ello que este documento integra de forma gráfica, y detallada cuando es necesario, los distintos diagramas dentro de los que destacan los casos de uso, diagramas de estado, secuencia, etc.

Básicamente existe un actor *jugador* y 5 casos de uso:

- Iniciar juego
- Recorrer nivel
- Cambiar Opciones
- Salir Juego
- Agregar Puntaje

3. Descripción General

3.1. Objetivos

El objetivo principal es la planificación y elaboración de un videojuego a ser evaluado en la asignatura *Seminario de desarrollo de software* por el cliente-profesor *Cesar Hernandez*.

A continuación se detallará la propuesta para lograr el propósito.

3.2. Planteamiento del Proyecto

Un punteo de las cosas más relevantes e influyentes en la propuesta solución ayudará a crear una mejor idea de lo que se quiere lograr.

- **Juego**, es necesario crear un juego y no existe alternativa de desarrollar un programa u otra aplicación.

- **Entretención**, la entretención es algo fundamental de un videojuego, sin ella el usuario se aburre rápidamente y no se cumple la misión de entretener y hacer pasar un rato agradable.

- ***Gancho***, se utiliza para captar la atención e interés de los usuarios y generalmente es algo con lo que la mayoría del público objetivo se siente identificado.

3.3. Solución Propuesta

Para cumplir la finalidad, se desarrollara un videojuego de celular en la plataforma *Java J2ME,* ambientado en el sistema de transporte público, de la capital de Chile, *Transantiago*. Este último servirá como gancho debido a que el videojuego es, prácticamente, una parodia del sistema de transporte tan criticado en los últimos tiempos por su ineficacia, mala planificación y demora en los recorridos.

La entretención es subjetiva, sin embargo, el ser un chofer de micro, del criticado sistema, que tiene que sortear obstáculos en las calles, transeúntes y anti sociales, le da una gota de humor y de aceptación inicial.

Esencialmente el juego consistirá en tomar el control de una micro y completar el recorrido en un tiempo establecido, pero además deberá recaudar una cantidad mínima de dinero que se logra gracias a los pasajeros recogidos en los paraderos, sin embargo en estos habrán antisociales, como también hoyos en las calles y gente cruzando calles, que intentarán impedir y agregar más dificultad al videojuego.

3.4. Características

Las principales características del videojuego serán:

- Gráfica en 2D
- Las micros amarillas tendrán IA, por lo que seguirán caminos distintos en cada oportunidad, intentando bloquear el paso del jugador.
- La cantidad de pasajeros y su tipo será aleatorio e independiente de cada paradero.
- La velocidad con que los peatones cruzan la calle será aleatoria.
- Los hoyos del camino se ubicarán aleatoriamente.
- Se incluirá publicidad dentro del juego.

Además estarán las características genéricas, tales como

- Sonido ambiental y de efectos especiales
- Manejo mediante el joypad y teclado numérico
- Opción para configurar los sonidos.
- Y muchas más.

3.5. Tecnologías

Transchantiago será desarrollado íntegramente en *J2ME*, y utilizara la herramienta de desarrollo *J2ME Polish*.

Además, como este proyecto se realizará con fines académicos, y no comerciales, no se requiere un pago de licencias.

4. Interfaz Hombre-Máquina

4.1. Perfiles

- Novato: persona sin conocimientos sintácticos del juego y escaso conocimiento semántico de la aplicación o del uso del celular en general.

- Intermedio: persona con conocimiento semántico razonable del juego, pero que recuerda vagamente la información sintáctica necesaria para usar la interfaz

- Avanzado: son individuos con buenos conocimientos semánticos y sintácticos, es decir, personas que buscan accesos directos y modos abreviados de interacción.

El juego está orientado a tanto a hombre como a mujeres, sin exclusión de sexo, raza u credo, sin embargo, se estima que la mayoría de los usuarios sean varones sobre los 12 años de edad. Clasificar a los usuarios según sus edades en los tres perfiles es posible gracias a las tendencias y avances en el aprendizaje de los menores en lo que es tecnología, por lo que se podría inferir que los *novatos* serian individuos que tiene sobre 40 años e *intermedio/avanzado* serian aquellas personas de 12 a 40 años de edad principalmente.

Por otro lado hacer una separación de usuarios intermedios y avanzados no se hace necesario, debido a la cantidad reducida de inputs que se dispondrán, los cuales corresponden a las entradas estándar de cualquier equipo celular con *joypad* y teclado numérico, sin embargo no se descarta un eventual acceso a los menú por intermedio del teclado numérico, de esta forma acceder a las distintas opciones se lograría tanto por el *joypad* como por números asociados a una opción especifica, de esta forma una combinación numérica podría lograr que el usuario avanzado acceda directamente, como por ejemplo, al juego o al menú de mejores puntajes.

4.2. Patrones de Diseño de interfaz de usuario

Para el diseño de la interfaz usuaria se ha optado por utilizar los patrones de diseño de Little Screen Design (Little Screen Design, 2008), debido a que están enfocados, principalmente, en el desarrollo de aplicaciones móviles.

4.2.1. Game Navigation

El menú está diseñado como una estructura de árbol, el cual contiene *nuevo juego, ranking, opciones, ayuda, créditos y salir* (Ver Imagen 4).

Cuando se el juego está en curso existe un menú de pausa que tiene las mismas alternativas del menú principal, con la diferencia de que en vez de salir la alternativa de *nuevo juego* saldrá *continuar juego*.

4.2.2. Back o Atrás

Los menús del juego tienen una profundidad de máximo dos nodos (incluyendo el nodo padre) por lo que esos menús incluyen la función back y vuelven al menú principal

4.2.3. Carrousel

El bus se encuentra en el centro de la pantalla, destacándose del fondo del escenario que es la calle. Los colores utilizados para el bus es el blanco y para la vía, plomo. Esto genera un contraste entre el personaje principal del resto de la ambientación

4.2.4. Mobile Typografy

La fuente utilizada por el juego para los títulos de los menús es *unlearne*, la que varía en tamaño y diseño según la funcionalidad que cumpla.

Imagen 1 fuente unlearne, usada en títulos

Imagen 2 fuente unlearne, usada como defecto en textos

Imagen 3 fuente unlearne, usada para resaltar texto (focus)

4.2.5. Margins & Gutters

Los márgenes de la aplicación son pequeños ya que en el contexto del juego se utiliza toda la pantalla para la localización de los elementos.

4.3. Prototipos de Interfaz

Imagen 4 Screenshots de los menús de Transhantiago

Imagen 5 Screenshot del video juego

5. Diseño del Sistema

5.1. Patrones de Diseño

El video juego no fue programado siguiendo un patrón de diseño en especial, debido a que se adopto el método ágil denominado *programación extrema* (Wikipedia - Programacion Extrema, 2008), cuya escancia radica en que pone más énfasis en la adaptabilidad que en la previsibilidad, además de su capacidad de adaptarse a los cambios en los requerimientos. Es por lo anterior, que en ningún momento de la planificación se decidió utilizar un patrón de diseño

5.2. Diagramas de Secuencia

5.2.1. Diagrama de secuencia de sistema – Iniciar Juego

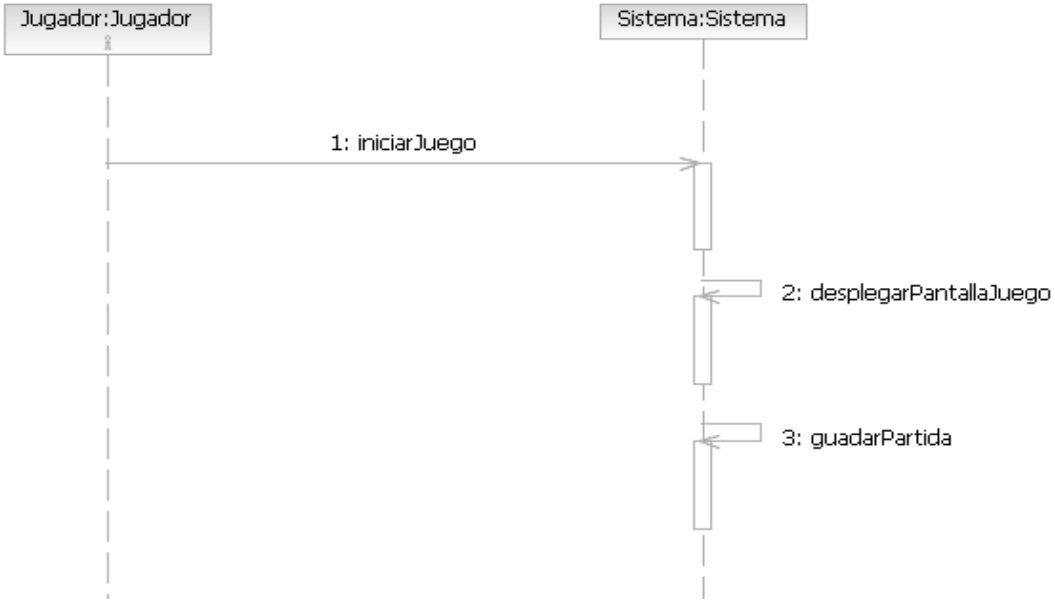

Imagen 6 diagrama de secuencia, iniciar juego

5.2.2. Diagrama de secuencia de sistema – Recorrer Nivel

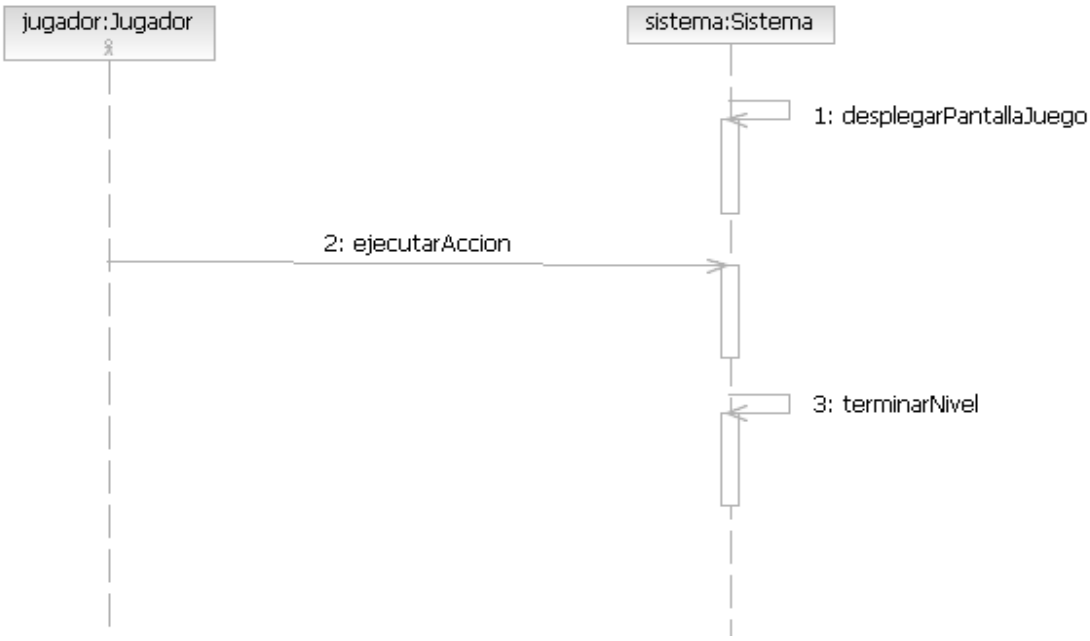

Imagen 7 diagrama de secuencia, recorrer nivel

5.2.3. Diagrama de secuencia de sistema – Cambiar Opciones

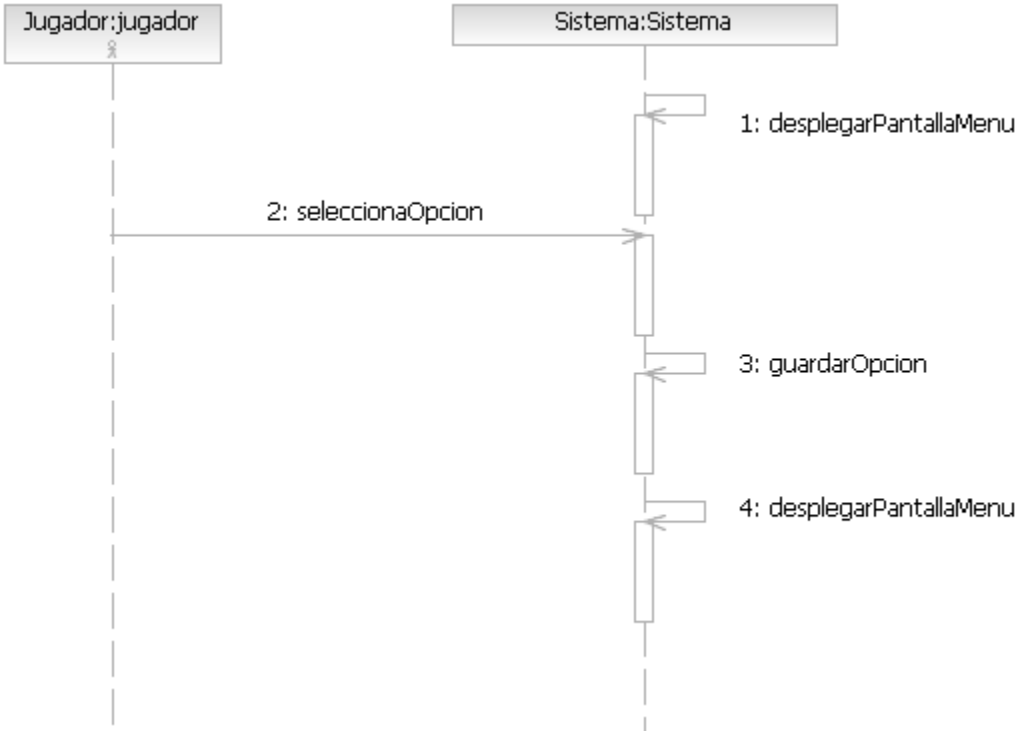

Imagen 8 diagrama de secuencia, cambiar opciones

5.2.4. Diagrama de secuencia de sistema – Salir del juego

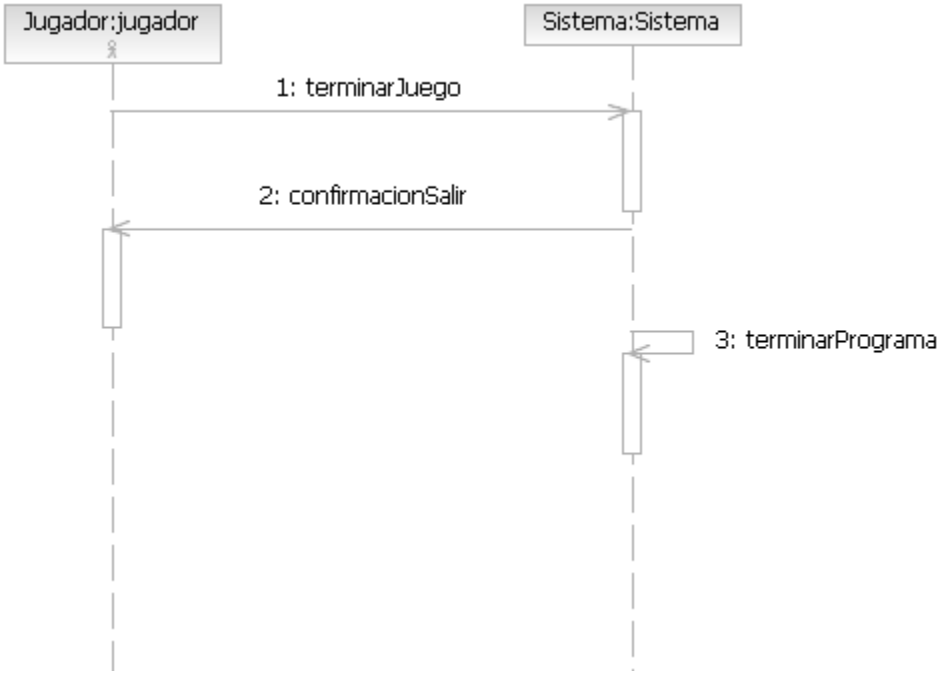

Imagen 9 diagrama de secuencia, salir del juego

5.2.5. Diagrama de secuencia de sistema – Agregar Puntaje

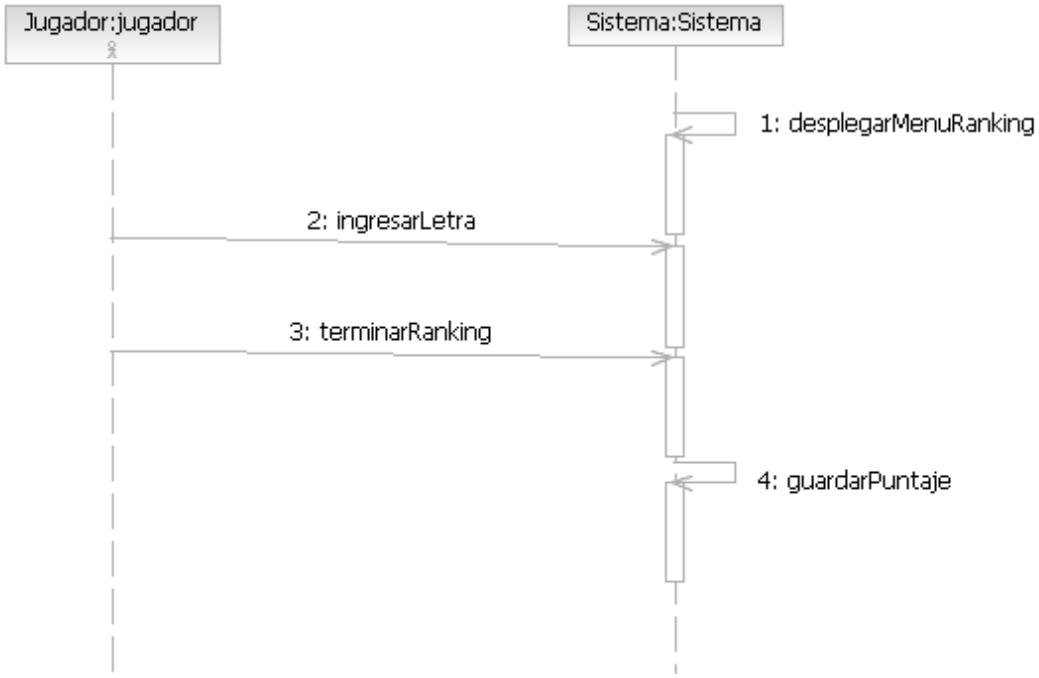

Imagen 10 diagrama de secuencia, agregar puntaje

5.3. Diagrama de Clases

El diagrama de clases puede ser observado en el Anexo 1.

5.4. Modelos de Datos

Para el desarrollo de esta aplicación no será necesario un modelo de datos, sin embargo hay datos que deben ser almacenados y para ello se uilizará *Record Management System* (Record Managment System, 2008) , con la siguiente estructura:

Tipo	Variable	Descripción
Int	sonido	volumen del sonido
Bool	Vibración	Vibración del celular
String	Jugador1	Nombre del jugador 1
Int	Puntuacion1	Puntuación del jugador 1
…	…	…
String	Jugador5	Nombre del jugador 5
Int	Puntuacion5	Puntuación del jugador

Tabla 1 Tabla de registros

5.5. Protocolos de Comunicación

El proyecto no requiere de protocolos especiales ya que todo radica en el mismo celular y no utiliza la arquitectura cliente servidor.

6. Implementación

6.1. Descripción por componentes

El proyecto transchantiago esta divididos en 5 elementos software.

- Transchantiago-Screens : Corresponde a las pantallas del menú opciones. Existe una clase por cada pantalla desplegada al usuario. AboutScreen, HelpScreen, HighScoreScreen, MainMenuScreen, SetingScreen y SplashScreen.

- Transchantiago-Audio: Contiene las clases que manejan el dispositivo de Audio y los sonidos del juego.

- Transchantiago-Game: Contiene Midlet principal y clases auxiliares de utilidades.

- Transchantiago-Data: Datos almacenados en un archivo RMS. Clases que abstraen la entrada y salida de datos almacenados en el teléfono.

- Transchantiago-Map: Contiene las clases que maneja los tiled layer y sprites de los elementos del juego. Se encarga de la carga y distribución de los tiles en el mapa.

6.2. Otras consideraciones

- Hojas de estilo. El diseño y manejo de texto y menú se realiza con **J2ME Polish**, su utilización es bastante simple y funciona de manera similar a como lo hacen los sitios web. Primero se deben agregar los paquetes relacionados al framework, posteriormente se deben definir las hojas de estilo, archivos .css, que contendrán toda la información del objeto que se desea alterar. Por último es necesario relacionar el objeto con el estilo que se desea aplicar, tal como se señala a continuación.

```
.especial {
      font-color : green;}
```

Tabla 2 Hoja_de_estilo.css

```
//# Style especial

StringItem PruebaString = new StringItem(null,"texto string");
```

Tabla 3 Archivo.java

- Creación de los mapas. La creación de los mapas se realizo gracias a la herramienta *Netbeans Game Builder* (Netbeans Game Builder, 2008)y para la edición de imágenes se utilizaron las herramientas de Fireworks 8.0 y Photoshop CS3. Una vez definidas las imágenes que se van a utilizar, estas se integran en el *game builder* y se manejan mediante *drag&drop.*

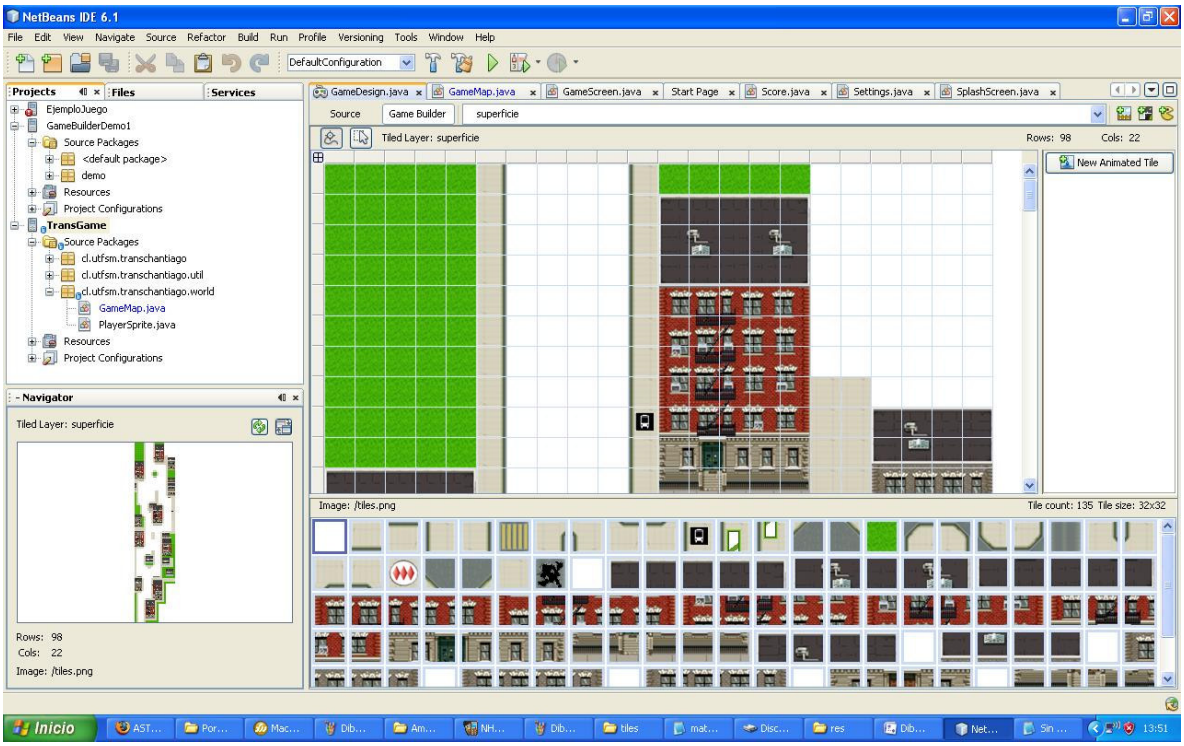

Imagen 11 Diseño del mapa en *Netbeans Game Builder*

- Documentación. Java en general posee harta documentación. Existe un sin número de libros que apoyan y facilitan la creación de aplicaciones en dicha plataforma, además existe gran cantidad de foros de discusión dedicados a la ayuda de los desarrolladores.

7. Aspectos Técnicos

El dispositivo donde se llevaron a cabo las pruebas es un **Nokia 5300**, sin embargo el juego se puede ejecutar en cualquier celular que cumpla con requerimientos mínimos siguientes:

- Soporte Java
- Soporte MIDP 2.0 y CLCD 1.1
- Heap size 512[KB]
- Optimizado para resolución de pantalla 240x320

Las especificaciones anteriores se aplican para cualquier dispositivo móvil y eso no excluye a que el video juega pueda ser corrido con un emulador en algún computador.

El desarrollo del video juego se realizó con:

- Java Wireless Toolkit
- Eclipse/Netbeans
- J2ME Polish
- Netbeans Game Builder
- Macromedia Fireworks 8.0 – Photoshop CS3

8. Pruebas Unitarias e Integradas

Para comprobar el correcto funcionamiento de un módulo independientemente del funcionamiento de otros se deben realizar lo que hoy en día llamamos pruebas unitarias, sin embargo esto no basta para asegurar un buen funcionamiento del sistema, es por ello que esta técnica debe combinarse, una vez terminada las pruebas unitarias, con otras denominadas de Integración, cuyo fin es asegurar el correcto funcionamiento, integración y/o sinergia entre las distintas unidades o módulos estudiados en la fase previa.

Debido al reducido tiempo declarado anteriormente, las pruebas de calidad llevadas a cabo se basaron principalmente en *netbeams debugger*, lo que permitió, en gran parte, solucionar la mayoría de los problemas detectados de manera eficiente y sin mayores complicaciones. Es importante mencionar, que lo anterior se pudo llevar a cabo gracias a que el tamaño del video juego, *versión beta*, es pequeño en comparación con una versión completa de cualquier otro video juego, los que integran mayor funcionalidad y complejidad en y entre los componentes, es decir, que *nebeams debugger* se adapto perfectamente a las necesidades del equipo de trabajo. Por otro lado, no se descarta la opción de integrar una técnica de prueba unitaria *más refinada* como lo son *JUnit* y *TestNG* por mencionar algunos, en una fase posterior a la creación del documento y anterior a la entrega del video juego (Wikipedia - Prueba Unitaria, 2008).

Para las pruebas unitarias se tomo en cuenta lo siguiente

- **Microbús (personaje principal)**, básicamente se probó que el comportamiento fuera ad-hoc con el input entregado, es decir, si se apretó a la *joystick* a derecha, o bien, el *número 6* la micro doblara a la derecha. Además, se probó la colisión con distintos objetos.

- **Obstáculos**, inicialmente se probó que los obstáculos aparezcan en la superficie y que colisionen con objetos que traten de superponerse.

Una vez que las unidades tenían su funcionalidad básica, se procedió a probar como era su comportamiento con los otros componentes. Se trabajo en pares, por ejemplo bus con obstáculos, en donde si un bus pasa por sobre un hoyo este último debería ver alterado su curso provocando una vibración en el celular.

9. Conclusiones

Antiguamente para crear un videojuego existían limitadas técnicas y diseños para la implementación y diseño de la aplicación, sin embargo hoy en día en base a la experiencia lograda en el transcurso del proyecto más el conocimiento adquirido, gracias a las exposiciones de los demás equipos, es totalmente cierto decir que las herramientas y tecnologías para construir un video juego son muchas y muy variadas, lo que facilita la creación de estos según el objetivo que se quiere lograr y las habilidades de los integrantes del grupo de desarrollo. Por otro lado, a pesar de la evolución de la tecnología en celulares, se siguen teniendo problemas básicos a la hora de programar una aplicación como son resoluciones de pantallas y keyboard-joystick, por solo mencionar algunos, por lo que aún queda por desarrollar y estandarizar en esta área.

Otra cosa importante a destacar es el avance en la ubicación de los tiles en la creación de los mapas, ya que antiguamente se hacía a mano y prácticamente a prueba y error, ahora es factible hacerlo gracias a herramientas como *Netbean Game Builder*.

A pesar de los inconvenientes y el reducido tiempo, el equipo de trabajo ha quedado satisfecho con lo logrado y muy conforme con la experiencia adquirida.

10. Anexo 1

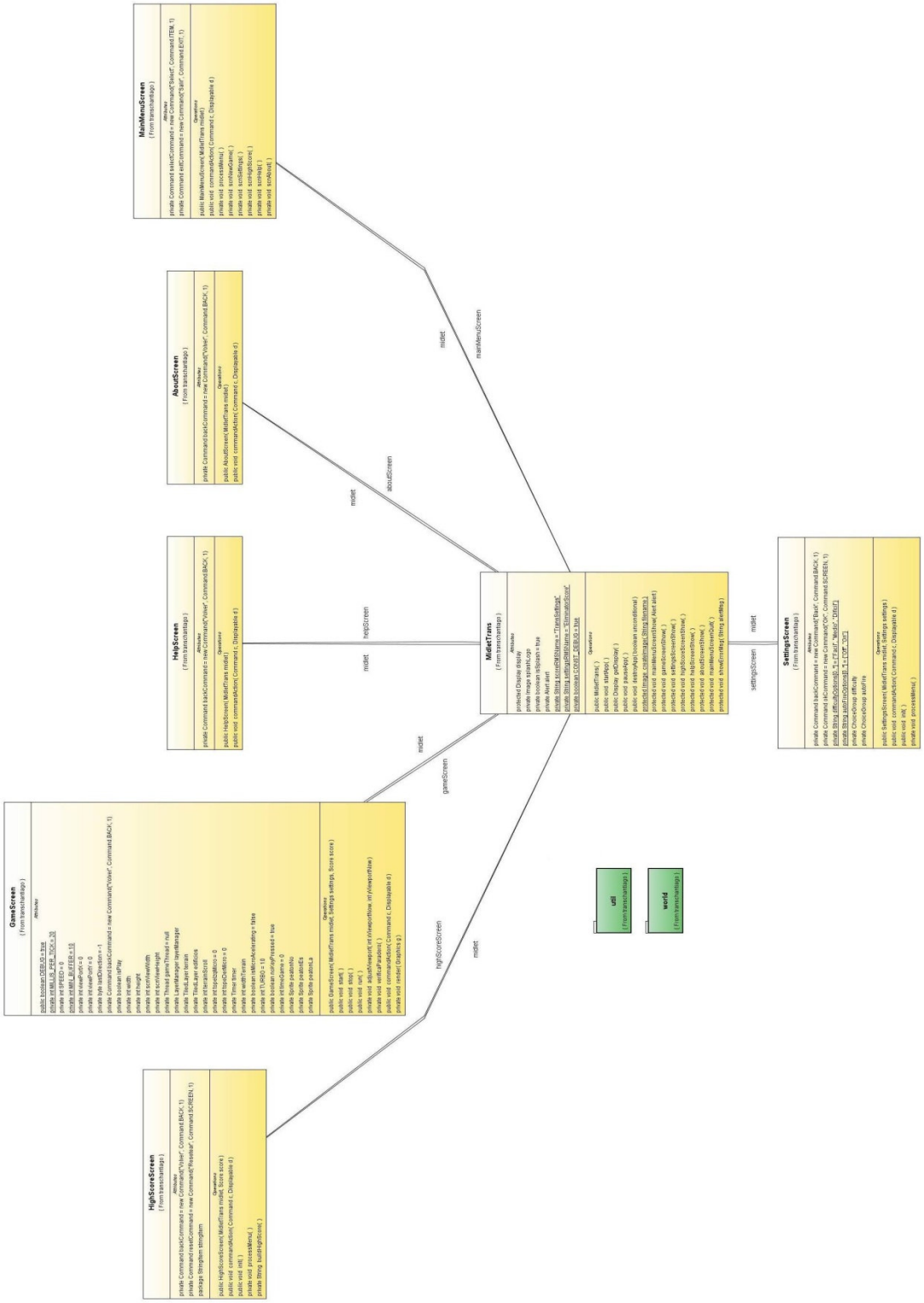

Imagen 12 Diagrama de clases de Transhantiago

11. Bibliografía

Crazy Taxi. (24 de 06 de 2000). Recuperado el 07 de 29 de 2008, de http://www.mobygames.com/game/crazy-taxi

Little Screen Design. (29 de 05 de 2008). Recuperado el 29 de 07 de 2008, de http://patterns.littlespringsdesign.com

Netbeans Game Builder. (11 de 03 de 2008). Recuperado el 29 de 07 de 2008, de http://wiki.netbeans.org/CreatingJavaMEGamesWithGameBuilder

Record Managment System. (02 de 06 de 2008). Recuperado el 29 de 07 de 2008, de http://en.wikipedia.org/wiki/Records_management

Wikipedia - Programacion Extrema. (18 de 05 de 2008). Recuperado el 29 de 07 de 2008, de http://es.wikipedia.org/wiki/Programaci%C3%B3n_Extrema

Wikipedia - Prueba Unitaria. (27 de 04 de 2008). Recuperado el 29 de 07 de 2008, de http://es.wikipedia.org/wiki/Prueba_unitaria