Universidad Nacional Autónoma de México

Posgrado en Ciencia e Ingeniería de la Computación

DESARROLLO DE UN FRAMEWORK GENÉRICO PARA DESARROLLO DE JUEGOS PARA FACEBOOK

TESIS

Que para obtener el grado de:

Maestro en Ciencias
(Computación)

PRESENTA:

Edgar Omar Cebolledo Gutiérrez

Directora de tesis
Mat. María Concepción Ana Luisa Solís González Cosío

México D.F. 2011

RESUMEN

Funcom fue fundada en 1993 y es la compañía desarrolladora de juegos más grande en Noruega, con más de 300 empleados de más de 35 países, tiene oficinas en Suiza, China, USA y Canadá. Han producido más de 25 títulos, entre sus juegos más aclamados están "*The longest journey*" y su secuela "*DreamFall*", asi como "*Anarchy Online*" y "*Age of Conan*".

Actualmente también tiene un departamento para juegos casuales y un departamento enfocado a juegos en Facebook.

Durante el semestre 2009-2, participé en el "Proyecto de Desarrollo de Videojuegos", el cual incluyó alumnos de tres universidades: Skövde, Suecia, NITH, Noruega, y UNAM, esto dentro del marco de colaboración entre las universidades de Skövde y la UNAM. Después del Proyecto, se me invito a realizar un internado en Funcom con el apoyo de NITH y la autorización del Posgrado.

Esta tesis explica el desarrollo de un motor de juegos para Facebook realizado durante una estancia de 6 meses (Enero/2010-Junio/2010 en Funcom Oslo, Noruega), durante la cual estuve trabajando con otros dos estudiantes de NITH (Escuela Noruega de Tecnología de la Información).

Esta tesis se divide en 2 partes, la primera son los antecedentes en la cual se explica Facebook, los juegos sociales y casuales, la segunda trata sobre la metodología y tecnologías usadas para el desarrollo del framework, cómo se desarrollo éste, y cómo se usó para programar un juego casual.

CONTENIDO

INTRODUCCIÓN:

Las redes sociales se han convertido en una parte fundamental de la internet, siendo Facebook uno de los sitios más visitados en el mundo, por lo que representa un gran mercado potencial. Funcom vio el potencial de este mercado en expansión, y en noviembre del 2009 formó un grupo dedicado al desarrollo de juegos sociales.

El objetivo de nuestro grupo era desarrollar un juego para Facebook, implementando todo lo que éste requería, servidores, un motor de juegos, conectividad con Facebook y bases de datos, etc.

OBJETIVO:

El propósito principal de este trabajo es la elaboración de un framework que facilite la creación de aplicaciones con conectividad a Facebook, éste se compondrá de 2 partes, por un lado está el servidor que se encarga de comunicarse con Facebook y que debe abstraer toda la funcionalidad, éste debe ser lo suficientemente general para poder ser usado en todo tipo de aplicaciones en cualquier lenguaje, siendo fácilmente modificable para que su mantenimiento sea sencillo y un motor de juegos en AS3 que se comunique con el servidor de comunicación, y que provea de las interfaces necesarias para la programación de juegos sencillos.

Además, se desarrollará el primer juego casual hecho en Funcom, sirviendo como caso de prueba para este framework.

PROBLEMAS:

Los principales problemas investigados en esta tesis son:

Cómo se puede diseñar un framework que simplifique el desarrollo de juegos para Facebook, proveyendo a los desarrolladores de un módulo que les permita acceder a todo tipo de redes sociales sin necesidad de conectarse directamente con ellas.

Cómo desarrollar un motor de juegos que agilice el desarrollo de juegos casuales.

ESQUEMA DE TRABAJO:

La primera parte de esta tesis explica la teoría y los conceptos de frameworks y conceptos relacionados con Facebook. La segunda parte describe los métodos usados, la investigación realizada y el diseño e implementación resultantes.

Parte 1:

Capítulo 2: Facebook.

Capítulo 3: Juegos Sociales.

Parte 2:

Capítulo 4: Metodología.

Capítulo 5: Flash y AS3.

Capítulo 6: Desarrollo de un motor de juegos en AS3.

Capítulo 7: Desarrollo de un servidor de conectividad con redes sociales.

Capítulo 8: Desarrollo de un juego casual usando el framework desarrollado.

Capítulo 9: Conclusiones.

Apéndice.

PARTE 1:

ANTECEDENTES

CAPÍTULO 2: FACEBOOK.

Facebook (FB), la red social más grande del mundo (después de haber superado a MySpace en el 2008), fue creada inicialmente por Mark Zuckerburg en el 2003 quien ya era conocido por haber creado un sitio llamado “*Facemash*”, un equivalente al sitio “*hot or not*”(sexy o no) creado en Harvard. Mark fue contactado por otros estudiantes que tenían la idea de crear un sitio de citas, pero él decidió crear su propia red, de ahí el concepto evolucionó en una versión digital de FB, un libro que es impreso cada año en muchas escuelas, el cual contiene fotos e información de los alumnos. Este sitio permitiría a los alumnos mantenerse en contacto. Despues de crear Facebook, Mark abandono Harvard. Unos años después Mark fue demandado por algunos de sus ex compañeros acusando de haber robado su idea para el sitio *ConnectU* los cuales habían contactado a Mark para que lo programara, pero la demanda no prospero debido a que no había ningún acuerdo firmado.

En primera instancia Zuckerberg creó esta red social ayudado por sus compañeros en Harvard en el 2004, desde septiembre del 2006, FB es operado por Facebook Inc.

Originalmente se había inventado solo para estudiantes de Harvard, pero la red gradualmente se expandió para incluir otras universidades y después todas las universidades en Estados Unidos y Canadá. Desde entonces ha continuado expandiéndose y actualmente está disponible mundialmente a cualquiera que sea mayor a 13 años y tenga una dirección de correo electrónico.

Todavía mantiene su filosofía de ser una red social donde la gente puede conectarse con sus amigos y familiares, escribir notas sobre ellos mismos, unirse y participar en grupos, y recientemente también pueden jugar juegos dentro de FB.

Otra de las nuevas funcionalidades es el poder conectarse con sitios fuera de FB, creando una especie de cuenta “universal” que pueda ser usada dentro y fuera de FB, usando su reciente API “*Open Graph API*” [2][3][4].

FACEBOOK COMO PLATAFORMA DE JUEGOS.

Los juegos para redes sociales como FB y MySpace consisten en juegos sencillos, jugados dentro del navegador dirigidos a un mercado de jugadores “casuales”. La “*Facebook Platform*” se lanzó en el 2007, y provee a los desarrolladores con un Framework para crear aplicaciones que se conecten con FB. Esta plataforma es usada por juegos para poder conseguir información sobre tus amigos, poder mandar invitaciones o escribir en tu muro (o en el de tus amigos), mandar correos, etc. [5].

Hay obvias ventajas al desarrollar para plataformas como FB, la principal es la gran cantidad de clientes potenciales. Actualmente la compañía Zinga [6] es la que tiene más jugadores activos y la que mejor ha podido explotar las herramientas que provee FB. Zinga tiene juegos famosos como *FarmVille*, *FishVille*, *YoVille* y *FrontierVille*, los cuales suman más de 60 millones de usuarios (hasta diciembre del 2009). Zinga, a pesar de las críticas en base a su modelo de negocios, ha crecido rápidamente hasta convertirse en un nombre reconocido por muchos. Ellos tienen un proceso de desarrollo basado en métricas y constantes actualizaciones, al grado de haber lanzado diez o más versiones de un solo producto para ver cual tiene mejores resultados [6][7].

Otra compañía importante es PlayFish[8], que fue comprada en el 2009 por EA, ellos han hecho juegos muy similares a los de Zinga (Zinga ha sido criticada por plagiar los juegos de PlayFish)[9], su primer éxito fue el juego de FB "*Who has the biggest brain?*" (¿quién tiene el cerebro más grande?) [10]. Este juego fue el primero que consiguió millones de jugadores activos en FB.

Aunque tiene grandes ventajas el trabajar con FB, también tiene grandes desventajas, el principal es que FB no fue hecho con el propósito de juegos, a pesar de haber tenido cambios recientes intentando atacar este problema, a la fecha FB no tiene un framework diseñado para juegos y el otro gran problema de FB es la latencia, tener más de 400 millones de usuarios implica que las peticiones al servidor pueden tomar mucho tiempo y definitivamente no es viable tener un juego en tiempo real que dependa de conexiones a FB [11]. Otro gran problema es la mutabilidad del API de FB, en su intento de proveer funciones para todo tipo de aplicaciones, FB ha cambiado, generado y despreciado varios API y seguro lo seguirá haciendo en el futuro, con lo cual se vuelve difícil tener un código de red estable.

AMBIENTES DE DESARROLLO PARA FACEBOOK:

FB ha liberado diferentes APIs en diferentes lenguajes y depreciado otros, actualmente los APIs oficiales son los siguientes:

• GRAPH API:

Es el API más reciente de FB, donde todos los elementos son nodos conectados en una red (de ahí el nombre Graph), cada uno con un identificador único. De esta forma conseguir información es trivial, simplemente se accede a https://graph.facebook.com/id

No toda la información es pública y para conseguir mayor información, se requiere un Access_token el cual usa el protocolo OAuth 2.0. Entre sus ventajas está su facilidad de uso, conseguir información (o agregar) en teoría es sumamente sencillo. Sus desventajas es que su documentación es prácticamente inexistente, y a pesar de que la documentación específica que todas las peticiones regresan un objeto del tipo JSON, en la práctica no es así y no tiene el mismo poder que los otros APIs [17].

• OLD REST API:

Este es uno de los APIs más viejos de FB, un servidor REST, este es el API más completo que tiene FB, puede usar FQ. Aunque FB ha comentado que no piensa dejar de soportarlo, también pide a los nuevos desarrolladores que usen Graph API. Su desventaja es que muchas funciones son cambiadas o depreciadas [18].

• FQL (FACEBOOK QUERY LENGUAJE):

FQL es similar a SQL, es la forma más rápida de acceder a información en FB, la sintaxis es prácticamente idéntica a SQL(usa comandos como: SELECT, INSERT, FROM, pero está más limitado que SQL. Puede regresar objetos JSON o XML, y permite múltiples consultas en una sola llamada, reduciendo considerablemente el número de peticiones a FB [19].

• FBML(FACEBOOK MARK UP LANGUAJE):

FBML es similar a HTML, el desarrollador usa etiquetas dentro del código de su página, y FB lee e interpreta esos tags y regresa el resultado al servidor. FBML es bastante rápido, pero FB exhorta a sus usuarios a usar los otros APIs en vez de FBML, salvo para las pestañas, las cuales según FB permitirá usar otras bibliotecas para desarrollarlas en un futuro. Es probable que FB piense en depreciar esta biblioteca en el futuro [20].

• FBJS (FACEBOOK JAVA SCRIPT) SDK:

FBSJ tiene muchas ventajas, tiene funciones para usar Graph API, FQL, y FBML, es sencillo de usar, ya que es JavaScript y es soportado por la gran mayoría de los navegadores. También ofrece varias formas de autenticar y usar funciones de manera dinámica (por ejemplo, la ventana para compartir una historia o logro en un juego).

Sus desventajas son que no puede ser usado en aplicaciones o plataformas que no soporten Java Script, y comparado con FBML es más lento [21].

Adicionalmente, crearemos un servidor que se conecte con FB, las dos opciones consideradas fueron PHP y Java (Java es usado en Funcom, y PHP es el lenguaje usado en algunas bibliotecas de FB)

PHP:

PHP es un lenguaje de “scripting” principalmente usado en páginas web, tiene muchas ventajas, es soportado por la mayoría de los servidores, se puede usar dentro de etiquetas HTML y es muy sencillo de usar. Muchos sitios web utilizan PHP. Entre sus desventajas, una de las principales que enfrentamos es el no poder transferir datos entre diferentes sesiones sin guardarlas en una base de datos [15].

JAVA SERVLETS:

Un servlet es un programa hecho en Java que responde a peticiones HTTP. Un Servlet es usado para generar contenido dinámico, un servlet recibe una petición GET o POST y puede regresar cualquier tipo de contenido, aunque el más común es HTML, también puede regresar XML, JS, o cualquier otro tipo. Es más poderoso que PHP, además de tener disponible todas las bibliotecas de Java, pero es menos soportado por los servidores y la curva de aprendizaje es más alta comparada con PHP [16].

CAPÍTULO 3: JUEGOS SOCIALES

JUEGOS CASUALES:

Recientemente se ha hecho la distinción entre un juego casual y uno no casual, hay muchas definiciones de juegos casuales, IGDA (Asociación Internacional de Desarrolladores de Juegos) los define como "juegos que generalmente tienen controles menos complicados y en general una complejidad menor en términos de modos de juego o tiempo requerido para completar el juego", la asociación de juegos casuales los define como "juegos desarrollados para el público general y familias y que son divertidos y fáciles de jugar", ninguna definición es completamente correcta, Farmville por ejemplo es considerado un juego casual y el tiempo que se requiere es incluso mayor al de muchos juegos no casuales, y también hay juegos casuales que no son tan fáciles de jugar.

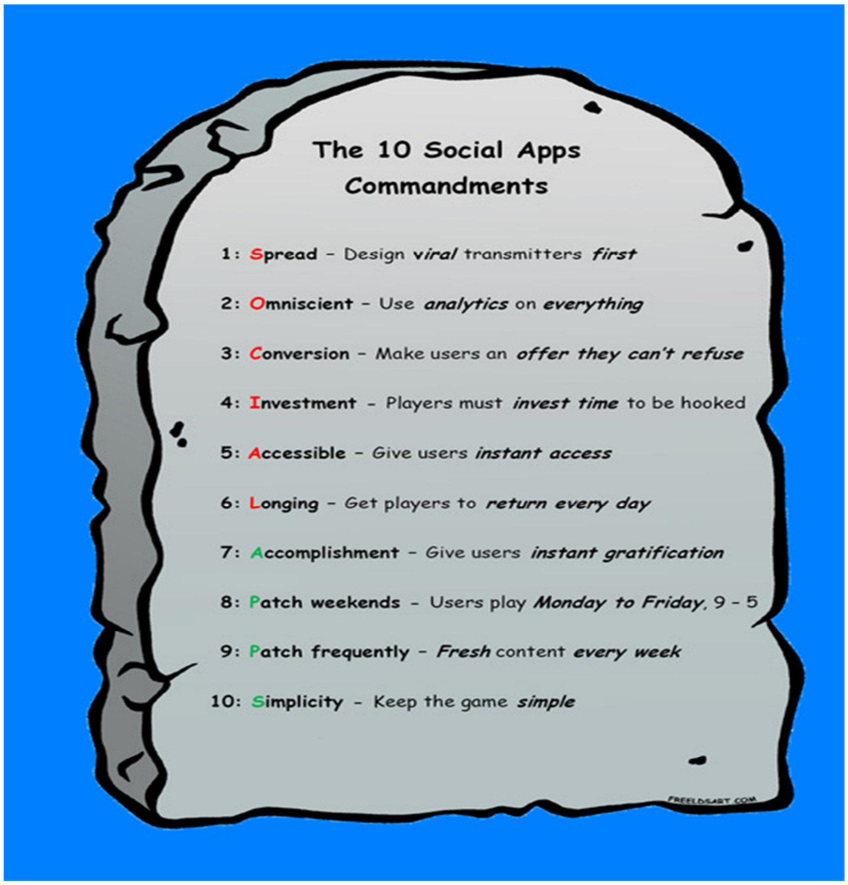
Para cada definición que usemos, siempre habrá algún juego casual que no la cumpla, pero en términos generales todos los juegos casuales comparten ciertos elementos:

- Son juegos sencillos de jugar, que no requieren una gran cantidad de botones o combinaciones.
- Cualquier persona sin importar su experiencia puede jugarlo sin problemas en cuestión de minutos.
- Están diseñados para ser jugados en poco tiempo, ya que la mayoría de los jugadores casuales no disponen de mucho tiempo.
- No requieren tanta precisión o velocidad a la hora de jugarlos, generalmente son más lentos y relajados
- La mayoría de los jugadores casuales son mujeres.

[12]

JUEGOS SOCIALES:

Con la aparición de las redes sociales ha surgido un nuevo género de juegos, los juegos sociales, *Farmville* y otros juegos de FB forman parte de esta categoría, la cual toma muchos elementos de los juegos casuales e incorpora nuevos, como los que escribió Jarle Snertingdalen (productor de Funcom):



The 10 Social Apps Commandments

1: Spread - Design *viral* transmitters *first*

2: Omniscient - Use *analytics* on *everything*

3: Conversion - Make users an *offer they can't refuse*

4: Investment - Players must *invest time* to be hooked

5: Accessible - Give users *instant access*

6: Longing - Get players to *return every day*

7: Accomplishment - Give users *instant gratification*

8: Patch weekends - Users play *Monday to Friday*, 9 - 5

9: Patch frequently - *Fresh* content *every week*

10: Simplicity - Keep the game *simple*

Fig. 1. Los 10 mandamientos de las aplicaciones sociales.

- Diseñar transmisores virales, elementos que ayuden a que el juego sea conocido por muchas personas, escribir al muro de un jugador o al de sus amigos, enviar correos electrónicos, etc.
- Usar métricas en todo, a diferencia de otros juegos aquí se intenta complacer a la mayoría, cada que se hace algún cambio al juego, se hacen varias versiones y los jugadores son redirigidos a estas diferentes versiones, después se toman métricas sobre esos grupos, y el grupo que tenga mejores métricas es el que se implementara en todos los servidores.
- Ofrecerles objetos o servicios con la intención de lograr que el jugador compre, un jugador que compra una vez, es más propenso a seguir comprando, entonces se intenta lograr esa primera compra
- Los jugadores deben pasar tiempo en la aplicación, los juegos deben hacer que el jugador tenga que dedicarle tiempo al juego, FarmVille por ejemplo tiene cosechas que maduran después de cierta cantidad de tiempo, haciendo que el jugador tenga que regresar a cultivar constantemente.
- Accesibilidad, el juego debe ser accesible tan rápido como sea posible, esto implica minimizar pantallas de carga o menús, la mayoría de los usuarios disponen de pocos minutos, por lo que el jugador debe poder empezar a jugar inmediatamente.

- Hacer que el usuario juegue diario, ya sea para tener el mejor record del día o para cosechar, o porque diario hay nuevas actividades o accesorios que conseguir o alguna otra forma.
- Gratificación instantánea, a la gente le gusta sentir que ha logrado algo, por lo que el juego debe premiar constantemente al usuario por sus logros obtenidos.
- Parchar los juegos los fines de semana, la mayoría del público objetivo es gente que trabaja y juega cuando tiene algunos minutos libres en su trabajo, por lo que se debe parchar cuando menos gente se conecta a jugar, que son los fines de semana.
- Parchar frecuentemente, los jugadores se pueden aburrir fácilmente para evitar esto hay que tener nuevos contenidos constantemente, nuevos accesorios, retos, etc.
- Simplicidad, la mayoría de los jugadores son casuales, por lo que el juego debe ser sencillo de jugar.

Mucho antes de la aparición de las redes sociales, ya existía un tipo de juego que lograba conectar a millones de personas en un mismo juego, los juegos masivamente online (MMO), estos juegos usan un único mundo persistente, esto es, todos los jugadores ven el mismo mundo, y este puede cambiar cuando el jugador no está presente, el mundo dentro del juego nunca deja de existir ni se pausa cuando un jugador deja de jugar, este siempre esta actualizándose. Otro aspecto importante es la comunicación entre los jugadores, la cual aunque no es obligatoria, es fomentada y para poder apreciar el juego o conseguir los mejores accesorios es fundamental. En muchos juegos MMO existen tareas que requieren la colaboración de varios jugadores al mismo tiempo, fomentando la comunicación. Originalmente eran exclusivos para PC, pero con la aparición de consolas de juegos con conexión a internet, han empezado a aparecer algunos juegos en consolas también. Hay todo tipo de juegos que son MMO, deportes, carreras, estrategia, rpg, etc. Actualmente el juego MMO mas popular es “world of warcraft” con más de 10 millones de usuarios activos.

Los juegos sociales tienen similitudes con los juegos MMO, pero tienen grandes diferencias, los MMO promueven el juego en equipo usando enemigos que requieren de muchos jugadores para ser derrotados, creando clanes que luchan contra otros clanes y permitiendo a todos los jugadores comunicarse entre ellos. Los juegos sociales promueven el juego en equipo premiándote con nuevos accesorios por tener amigos que también jueguen, necesitando que tus amigos te envíen accesorios que solo se pueden conseguir de esta forma e invitando nuevos amigos al juego.

Los juegos sociales pueden ser divididos en 3 categorías:

Dependientes de la red:

> Estos juegos no pueden existir fuera de una red social y requieren de las redes sociales para su funcionamiento. A pesar de que el juego se podría modificar para poderse jugar fuera de una red social, perdería su atractivo. Juegos como Farmville o FrontierVille son ejemplos de juegos dependientes de la red.

Integrados a la red:

Son juegos que pueden existir fuera de la red, pero su diseño se presta para usarse dentro de las redes sociales, generalmente son juegos basados en turnos, que aprovechan la red social para poder notificar al otro jugador que es su turno. La mayoría de los juegos de mesa como ajedrez o damas podrían ser integrados a una red social, un ejemplo de esta categoría es Scrabulous, el cual es un clon de Scrabble.

Independientes a la red:

Son juegos que no aprovechan todas las ventajas de las redes sociales, son juegos que podrían existir fuera de FB, juegos para una sola persona y que limitan su integración a escribir en muros, enviar invitaciones y tablas de records. Hay muchos ejemplos en FB de este tipo de juegos, Typing Maniac, Bejeweled, etc.

PARTE 2

INVESTIGACIÓN Y DESARROLLO

CAPÍTULO 4: METODOLOGÍA

El método de desarrollo de software que usa Funcom es SCRUM, usando PERFORCE como sistema de control de versiones, por lo que nuestro grupo usó esta misma metodología.

Para poder desarrollar el framework se requieren conocimientos de la plataforma de FB, de Servidores, Bases de Datos, AS3, orientación a objetos entre otros. Por lo que lo primero que se hizo fue investigar estas tecnologías. Las tecnologías estudiadas fueron presentadas en la primera parte. Una vez que se tuvo un mejor entendimiento de estas tecnologías, vino la fase del diseño del motor y del juego a implementar. Después de tener un diseño básico del motor, se empezó a implementar este, y el juego a implementar sirvió como casos de prueba para el motor y el servidor.

SCRUM

La ilustración de la figura 2 muestra el proceso básico de SCRUM, el cual explicaremos mas adelante, pero primero debemos explicar la terminología básica.

La reserva del producto (backlog) es una lista priorizada de los requerimientos del producto, incluyendo nuevas funciones, tecnologías, mejoras, y correcciones de errores. El contenido puede venir de todas las áreas; usuarios, clientes, programadores, diseñadores, artistas, etc. Pero solo el dueño del producto (se explicara más adelante) puede priorizar la lista, y decide que se implementará primero.

Los equipos de desarrollo toman la mayor cantidad de tareas que puedan(o crean poder) manejar durante la siguiente iteración, que en SCRUM se le llama sprint. Durante cada sprint, los equipos tienen una lista de tareas a realizar, llamada “la reserva del sprint”. Cada sprint debe terminar en una versión completamente funcional del producto.

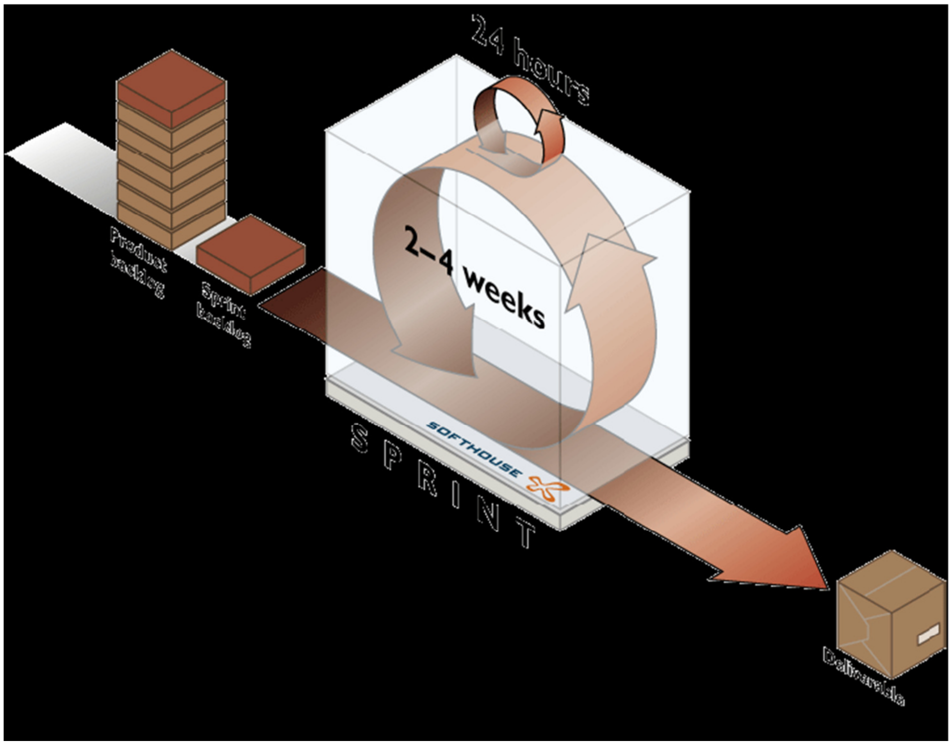


Fig. 2. El proceso de SCRUM

ROLES

SCRUM tiene 3 roles principales; el dueño del producto, el maestro, y el equipo.

- Dueño del producto

  Representa al cliente. Como ya se mencionó, éste es el que prioriza la reserva, y al final del sprint, es el responsable de evaluar el sistema.
  Para desarrollos comerciales, el dueño puede ser el anterior gerente de producto, para desarrollos internos, el dueño puede ser el gerente de producto o el del departamento de usuarios.
- Maestro

  Es responsable de asegurar que los valores, prácticas y reglas de SCRUM se cumplan. Es el que fija las reglas y hace que se cumplan.
  El líder del equipo, o del producto o el gerente generalmente asumen este rol.
- Equipo

  Los equipos en SCRUM son pequeños (entre 5 y 9 personas) y auto organizados y multifuncionales, llevan a cabo el diseño, desarrollo y pruebas entre todos. El equipo tiene la autoridad de hacer lo que sea necesario para que se cumpla la meta del sprint. Los miembros pueden interrogar a otros, traer ayudantes, leer libros, internet o lo que sea necesario (que este dentro del presupuesto) para completar la meta.

No hay roles ni títulos dentro del equipo y no hay descripciones del trabajo, únicamente realizarlo lo mejor posible.

EL PROCESO

REUNION DE PLANEACIÓN DEL SPRINT

Durante la reunión de planeación del sprint, los clientes, usuarios, gerentes, el dueño del producto y el equipo determinan la funcionalidad y meta del próximo sprint. La reunión consiste en dos reuniones diferentes. Primero los equipos se reúnen con el dueño, gerente y usuarios para discutir que funcionalidad se desarrollará en el siguiente sprint. En la segunda reunión, el equipo trabaja por sí solo para discutir cómo se implementará esta funcionalidad durante el sprint.

REUNIONES DIARIAS

Cada equipo se reúne diariamente durante quince minutos. El maestro de SCRUM es el encargado de que se lleve a cabo esta reunión correctamente, evitando que dure más de lo debido y asegurándose de que todos hablen brevemente. Otros integrantes pueden asistir a la reunión, pero están ahí como invitados y no pueden interferir en ésta.

Cada integrante del equipo debe responder brevemente tres preguntas:

- ¿Qué has hecho desde la última reunión?
- ¿Qué harás después de esta reunión y antes de la próxima?
- ¿Qué problemas tuviste en tu trabajo?

Las primeras dos preguntas dan un breve reporte de progresos y permite que el equipo se sincronice. Si un miembro identifica algo que no le permite avanzar, es deber del maestro remover ese impedimento. Tales impedimentos pueden ser un servidor que está caído, o que un gerente le pidió hacer alguna otra actividad.

REUNIÓN DE EVALUACIÓN DE SPRINT

La reunión de evaluación es una reunión informal de cuatro horas de duración, donde el equipo presenta lo que logró realizar durante el sprint.

La administración asiste a la reunión para ver lo que el equipo ha realizado con los recursos que le fueron asignados. Los clientes asisten para ver si les gusta lo que el equipo desarrolló y el dueño asiste para ver que funcionalidad se agregó.

Puesto que es una reunión informal, nadie debe preparar nada para ésta, por lo que no deben prepararse presentaciones en power point o similares. Esta evaluación es una reunión de trabajo donde todos deben entender el incremento en el producto, pues ésta es la información necesaria para la próxima reunión de planeación.

REUNIÓN DE RETROSPECTIVA DE SPRINT

Después de cada evaluación, pero antes de la próxima planeación, se lleva a cabo una reunión de retrospectiva donde todos los miembros reflexionan sobre el sprint anterior. El propósito de esta reunión es mejorar el proceso contínuo.

VENTAJAS

CORTAS ITERACIONES

Al usar cortas iteraciones y reuniones diarias, el equipo siempre sabe dónde está y a dónde va. Por ejemplo: todos los días saben que funciones faltan, qué tanto se ha progresado en las tareas pendientes y los problemas se detectan diariamente.

NO HAY PÉRDIDAS DE TIEMPO

Debido a que siempre se trabaja en las funciones más importantes, SCRUM asegura que el equipo no perderá tiempo desarrollando funciones que nadie usará. Como resultado la velocidad de desarrollo se incrementa.

MULTIFUNCIONALIDAD

SCRUM intenta evitar la barrera entre los diferentes roles y hacer que todos trabajen conjuntamente hacia la meta. Todos son responsables del producto y trabajando juntos el equipo junta sus habilidades para crear mejores productos.

VALORES IMPORTANTES EN SCRUM

Al usar SCRUM, al ser equipos auto organizados y multifuncionales, tienen el poder de resolver las situaciones complejas entre ellos mismos. Esta libertad, junto con la creatividad que surge de ésta, es uno de los beneficios principales de SCRUM. El equipo se hace más responsable del producto si no se les dice todo el tiempo qué deben hacer.

El maestro de SCRUM es un líder, no un jefe. Así, en muchos casos será necesario el cambio de controlar a facilitar. Esto es probablemente lo más difícil en muchas organizaciones, cambiar la forma de pensar.

SCRUM EN LOS PROYECTOS EN FUNCOM

Funcom usa SCRUM como metodología de trabajo, pero cada equipo tiene algunas variantes a la metodología SCRUM. A continuación explicaré cómo llevaba a cabo la metodología mi equipo de trabajo.

PLANEACIÓN DEL SPRINT

Cada inicio de sprint, nuestro equipo se juntaba con una diseñadora y un productor de Funcom que fungían como facilitador y dueño respectivamente. Durante esta reunión, se discuten qué tareas se desarrollarán en este sprint y quién las llevará a cabo. Siempre dándole prioridad a las tareas más importantes para el funcionamiento del sistema, cada tarea es evaluada por todos los miembros del equipo y se le asigna una dificultad a cada tarea.

Al final de cada sprint, también se tenía una reunión con los mismos integrantes, durante éstas se revisaban las tareas asignadas al sprint, se revisan cuáles se cumplieron y cuáles no. Las tareas cumplidas deben ser mostrables para considerarse cumplidas (y deben funcionar como es debido), y los encargados de la tarea que no se cumplió deben explicar el por qué no se cumplió esa tarea. También se aprovechaba para discutir el diseño del juego.

REUNIÓN DIARIA

En las oficinas teníamos un pizarrón dividido en 4 áreas, reserva, tareas actualmente realizándose, tareas que requieren probarse y tareas completadas. Cada tarea tenía un post-it asociado. Al inicio de cada día, todo el equipo se reunía y cada uno explicaba qué había hecho el día anterior, qué complicaciones tuvo y qué pensaba hacer ese día, en caso de que empezara una tarea nueva, o considerara que la tarea estaba terminada o que faltaba ser probada, el post-it respectivo se movía en el pizarrón.

DEMOS

Al final de cada iteración, durante la reunión de fin de sprint, todas las tareas que se habían completado debían ser mostrables, esto servía como motivación para el equipo, pues aparte de terminar la implementación de la tarea, debía incorporarla de alguna manera al juego que estábamos desarrollando. No necesariamente debía incorporarla donde le correspondía en el juego, simplemente que fuera mostrable y probable.

POKER DE PLANEACIÓN

Durante las reuniones al inicio de cada sprint, se evaluaba la dificultad de las tareas entre todos los integrantes del equipo. Cada miembro del equipo tenía un conjunto de cartas con valores numéricos. Por cada tarea, el responsable de esa tarea explicaba en qué consistía y todos los integrantes del equipo ponían una carta boca abajo, con el valor numérico que le daban a la dificultad de esa tarea. Cuando todos ponían su carta boca abajo éstas se levantaban, si algún valor es mayoría, esa es la dificultad que se le asocia a esa tarea. Si hay mucha discrepancia en los valores, el que asignó mayor y menor valor explican el por qué de su razonamiento y de nuevo se vuelve a hacer una valoración.

Todos los integrantes participan en la valoración de todas las tareas, esto ayuda a que todo el equipo tome conciencia de las tareas de los demás integrantes, fomentando la integración. El valor numérico de las tareas es único a cada equipo y no puede ser comparado con otros equipos, pues cada equipo valora diferente cada tarea, la valoración sirve para que el equipo pueda saber sus alcances y limitaciones dentro de un sprint.

CAPÍTULO 5: FLASH Y AS3

AMBIENTES DE DESARROLLO:

Una parte fundamental de nuestro framework es el motor del juego, el cual debe poder ser embebido en una página de internet. Hay muchas alternativas que pueden ser embebidas en páginas web, la más común y la que la compañía quería usar es Action Script 3(Flash) [3], ya que Flash es usado por otros juegos en FB y la gran mayoría de la gente tiene instalado el plug-in de Adobe para Flash. A continuación explicaremos los ambientes usados.

ACTION SCRIPT 3:

Action Script es el lenguaje oficial para Adobe Flash. Empezó siendo una herramienta para animaciones pero ha crecido mucho más allá de su propósito inicial. Action Script 3 es un lenguaje orientado a objetos similar a C# o Java.

Las aplicaciones pueden ser desarrolladas usando solamente código, o combinándolo con herramientas graficas de desarrollo.

Las aplicaciones flash son compiladas en archivos con formato .swf. Este archivo contiene 2 partes, la primera es Action Script bytecode(ABC). La segunda parte son objetos multimedia adjuntados como imágenes, video o audio. Los archivos son compilados “just in time” a código maquina por el interprete de flash. Hay diferentes tipos de intérpretes, (aunque todos tienen una maquina virtual que interpreta AS3), Flash lite por ejemplo es usado en dispositivos portátiles como celulares. Flash player es usado en páginas web y Adobe AIR es usado para aplicaciones de escritorio.

AS3 tiene varias propiedades útiles, muchas similares a otros lenguajes orientados a objetos. Flash usa un modelo de ejecución de un único hilo de ejecución con verificación de tipos en tiempo de ejecución y en tiempo de compilación (opcional). También puede lanzar excepciones y soporta funcionalidades básicas de orientación a objetos como herencia, interfaces, manejo de eventos y recolector de basura (pero no tiene clases abstractas).

Otra función útil es el uso de paquetes. Los paquetes facilitan el crear una jerarquía u orden de las clases en un proyecto. Los nombres de las clases dentro de un paquete siempre son precedidos por el nombre del paquete. También soporta XML nativamente, característica que fue usada por nuestro motor de juegos.

Existen varios frameworks de desarrollo para Flash, nosotros usamos el que es soportado por Adobe, Flex.

FLEX:

Flex permite programar usando AS3, pero también se puede usar MXML (una extensión a XML). Facilita el crear una aplicación proveyendo bibliotecas para mostrar mensajes, botones, y demás elementos comunes a interfaces como checkboxes, botones de radio, etc.

Flex puede ser usado en varios ambientes de desarrollo, nosotros lo usamos con IntelliJIdea, aunque para el nivel de personalización que queríamos casi no se usaron los componentes de Flex, pues proveen de botones y elementos para interfaces, pero no del tipo de los que son usados comúnmente en juegos.

CAPÍTULO 6: DISEÑO DEL MOTOR DE JUEGOS

Ya que sabemos qué tecnología usaremos para programar el motor, se puede empezar a diseñar éste. A pesar de que se puede diseñar un motor de juego desde antes de saber qué tecnología se usará, generalmente no es recomendable, pues no se aprovecharían las herramientas o paradigma de la tecnología a usar. Por lo que primero empezamos investigando la tecnología a usar y después lo diseñamos tomando en cuenta en qué se va a desarrollar.

MOTORES DE JUEGO:

Un motor de juego es un sistema diseñado para la creación de juegos, provee de interfaces para poder mostrar objetos en pantalla, detección de colisiones, sistemas de física, etc.

En sus inicios todos los juegos eran programados sin usar motores, esto implicaba tener que programar rutinas similares para cada juego. Pero dado que el hardware era muy limitado, se tenía que optimizar al máximo el código específico de cada juego, por lo que no era viable usar motores de juego, pero a mediados de la década de 1980, conforme la tecnología se empezó a volver más estándar y la capacidad de cómputo se incrementaba, se empezaron a reusar partes de código hasta poder separar la parte específica al juego y la parte encargada de proveer interfaces. Uno de los juegos pioneros fue Doom[22], cuyo código fue usado por otras personas para crear nuevos juegos cambiando las gráficas, el diseño de los niveles y personajes, pero manteniendo la misma forma de jugar. En la actualidad hay compañías que se dedican a desarrollar motores de juego como Unreal[23], vendiendo licencias a los otros desarrolladores, a pesar de que estas licencias en su mayoría tienen un alto costo, es benéfico para las compañías pues se pueden enfocar a programar específicamente el juego lo cual acelera los tiempos de desarrollo de manera sustancial.

Cada motor de juegos debe satisfacer ciertas necesidades, por lo que no hay un motor de juegos óptimo o ideal para todo tipo de juegos. La compañía quería poder crear juegos muy sencillos en AS3, pero también muy rápido, idealmente, programar un juego nuevo debería tomar entre 1 y 3 días. Adicionalmente, todos los juegos tendrían interfaces similares. Se quería que en la parte inferior, similar a otros juegos, se mostrara una barra con los mejores jugadores (de entre tus amigos que han usado la aplicación), mostrando su imagen, nombre y puntaje. Se quería una opción para poder invitar a tus amigos a usar la aplicación, y una forma de retar a uno de tus amigos a competir por el máximo puntaje entre los dos, similar a juegos como “*Typing Maniac*” [24].

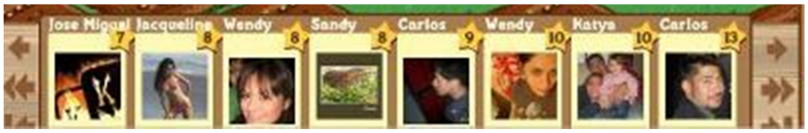


Fig. 1. Barra de amigos con nombre, imagen y nivel en Farmville.

FUNCIONALIDAD BÁSICA DEL MOTOR:

Para este motor tenemos varias necesidades:

- Debe poder conectarse con un servidor y enviar y recibir información.
- Debe tener una barra con los mejores jugadores, su foto y su puntaje (similar al de otros juegos en FB).
- No requiere poder cargar contenido dinámico o descargable.
- El tiempo de carga inicial debe ser tan rápido como sea posible. A pesar de que este punto es importante en cualquier juego, en juegos para FB es todavía más importante, pues la mayoría de los jugadores no tienen mucho tiempo libre, por lo que el juego debe ser jugable casi inmediatamente, incluso sacrificando elementos que no sean primordiales. Farmville por ejemplo carga lo mínimo para jugar, y algunos elementos los carga mientras el jugador ya está jugando (avatares, iconos, sprites).
- Sera usado para juegos sencillos y que deben ser desarrollados en cuestiones de días.

Con estas necesidades, fue que empezamos el diseño de un motor de juego en AS3, la fig. 3 muestra un diagrama de clases simplificado

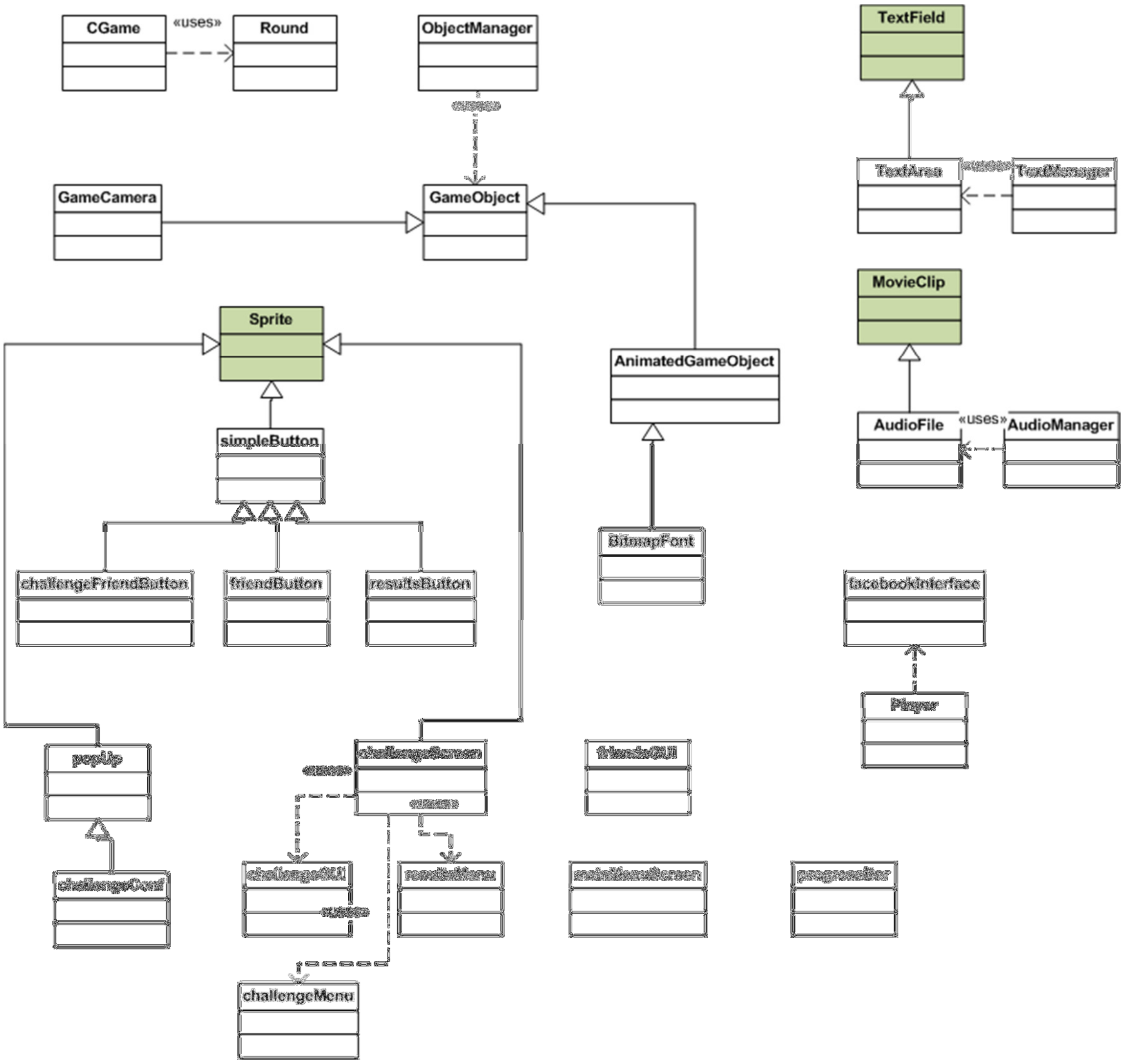


Fig. 3. Diagrama de clases simplificado.

Como se mencionó antes, un motor debe proveer interfaces al programador para que éste no tenga que preocuparse en algo que no sea específico al juego a desarrollar.

A pesar de haber muchos motores de juegos hechos en AS3, se decidió crear un motor nuevo, porque los otros motores proveen demasiadas funcionalidades que no se usarían y que terminarían alentando a los juegos, el motor que hicimos fue hecho pensando en simplicidad y velocidad y no incorpora algunas funcionalidades básicas de motores de juegos como detección de colisiones y física(de hecho hay muchos motores que no los incorporan). En caso de ser necesaria esa funcionalidad es sencillo incorporarla al motor, ya que un motor de física solamente tendría que crear una clase que extienda a GameObject, y que contuviera un Sprite y un objeto físico, en el método update del ObjectManager se agregaría la funcionalidad de actualizar el motor de física y eso sería todo.

Todos los elementos multimedia del juego deben ser descargados de un servidor, esto con el fin de poder acceder al juego lo más pronto posible, mientras más archivos adjuntos tiene una aplicación, mayor tamaño será y tomará más tiempo descargarlo, además nos facilita el poder cambiar estos archivos en un futuro. El motor lee todos los elementos gráficos de los menús de un archivo XML. Esto en muchos casos acelera el proceso de desarrollo de nuevos juegos en gran manera, ya que una vez teniendo un juego con un menú funcionando, crear uno nuevo es tan sencillo como copiar el archivo XML y cambiar la dirección de las imágenes(y su posición si el tamaño es distinto). Esto sumado al servidor del juego, lograría que se pueda tener el menú principal del juego con retos e invitaciones y la barra del fondo mostrando a tus amigos en unos cuantos minutos. Haciendo que el programador se enfoque exclusivamente al juego a desarrollar.

La desventaja de no incluir los elementos multimedia en el juego es que al iniciar el juego a lo mejor no estarán todos los elementos gráficos en pantalla por unos segundos en lo que son cargados estos elementos, pero en juegos sociales, es muy importante que el jugador empiece a jugar lo más rápido posible, ya que el jugador se puede desesperar fácilmente.

Farmville hace esto, carga lo más básico del juego al empezar, y después una vez cargado el juego empieza a descargar las imágenes del juego que no son tan importantes como tu avatar o tus cosechas.

ARQUITECTURA

Para la arquitectura del motor, consultamos diferentes motores de juego y nuestro enfoque es similar a la arquitectura de XNA[25], basada en componentes, en nuestro caso llamados objetos de juego (Game Object), donde tenemos una clase que contiene todos los componentes del juego, y es la encargada de pintarlos y actualizarlos. Así, para crear un nuevo elemento, basta con hacer que extienda a Game Object, y sobrecargue la función Update y Draw.

Para los objetos que formarán parte del juego, tenemos soporte para Sprites animados, textos, audio, una cámara en 2D, fondos, textos hechos de imágenes y soporte para rounds o niveles. También tenemos soporte para barras que se mueven de lado a lado, que aunque no son usadas por todos los juegos, pueden ser útiles para muchos juegos sencillos.

Tenemos clases dedicadas a la comunicación con FB, éstas se comunican con un servidor usando un protocolo REST, reciben la información en XML, y guardan esa información en objetos del tipo Player. (Facebook Interface, Player).

Tenemos clases dedicadas a los menús o pantallas que todos los juegos usarán, como la pantalla para retar a algún amigo, la pantalla de resultados, la barra de progreso del juego, el menú principal y otras.(Challenge Conf, Challenge GUI, Challenge Screen, FriendsGUI, ResultsMenu, ChallengeMenu, MainMenuScreen, ProgressBar)

Tenemos clases usadas para los botones, (ChallengeFriendButton, FriendButton, ResultsButton).

La figura 3.1 muestra un diagrama con los diferentes componentes usados.

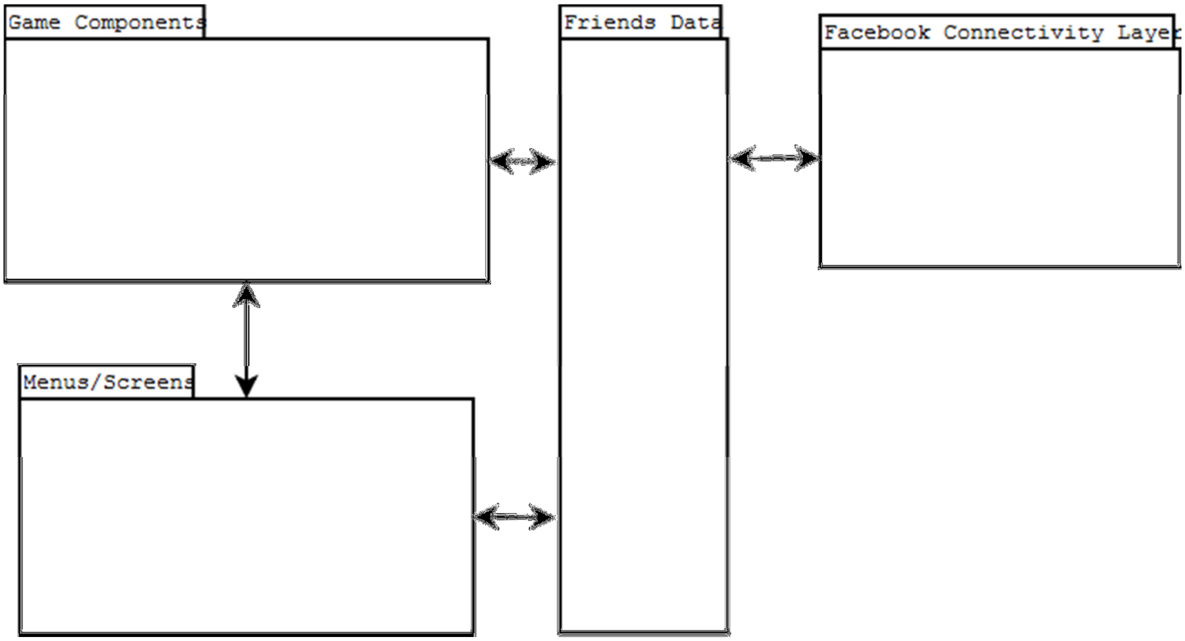


Fig 3.1. Diagrama de la Arquitectura

CAPÍTULO 7: DISEÑO DEL MÓDULO DE COMUNICACIÓN CON FACEBOOK

Ya que se tiene el motor de juego, la otra funcionalidad importante del framework es conectarse con FB, por lo que se diseñó un módulo encargado de comunicarse con éste.

Uno de los principales problemas a la hora de hacer un programa que se comunique con FB, es que el API de FB es altamente mutable, FB siempre está haciendo cambios a sus entornos de desarrollo, algunos cambios son menores, pero aún así obligan al programador a cambiar su código constantemente.

MÓDULO DE COMUNICACIÓN CON REDES SOCIALES (SOCIAL NETWORK WRAPPER SNW)

Con esto en mente, el módulo de comunicación con redes sociales (pues no solo debe comunicarse con FB, sino también con otras redes sociales en un futuro), debe poder ser ligero, fácilmente modificable, debe poder ser usable por todo tipo de aplicaciones creadas en todo tipo de lenguajes (no solo AS3, también C/C++ y Java especialmente) y la aplicación debe poder realizar las mismas funciones a pesar de que el módulo cambie, es decir, si una red social cambia la forma de acceder o escribir información, el único código a modificarse debe ser el del módulo, jamás el de la aplicación.

Todo esto es con la finalidad de poder tener muchas aplicaciones conectadas a redes sociales y si una red social cambia su API, las aplicaciones no deben ser modificadas, solamente el módulo. Este es un enfoque similar al mostrado por Zinga en la GDC 2010 [26].

Hay varias formas de ofrecer servicios web, (SOAP, RPC, REST, etc.), para este módulo utilizamos una arquitectura REST sin usar SOAP, donde las respuestas están codificadas en XML (se consideró usar JSON, pero XML es más soportado actualmente que JSON), la razón de esto es que no queremos tener que implementar protocolos complicados como SOAP, usando una arquitectura REST, simplemente los programas envían su petición usando GET o POST, y reciben la respuesta en formato XML. Debido a que muchas aplicaciones en diferentes lenguajes se conectaran a este modulo, se decidió usar un protocolo simple y sencillo (FB tiene un API similar llamado "*Old REST API*").

También, al no querer que la aplicación se tenga que modificar en caso de que alguna red social modifique su API, tenemos que crear un conjunto de campos a los cuales se puede acceder (como nombre de usuario, identificador, fecha de nacimiento, etc.). Y el módulo necesita procesar esos campos para acceder a la información necesaria. Esto es pensando a futuro, que algunos campos pueden dejar de ser accesibles o cambiar la forma de acceder a ellos en el futuro, entonces el servidor leería esos campos y haría lo necesario para poder regresarlos. También hace más independiente al servidor de las redes sociales.

Toda la funcionalidad que no es específica a juegos está en este módulo, pero la arquitectura usada consta de 3 elementos, el módulo de conexión a redes sociales (SNW), un servidor que provee al juego de varios servicios como acceso a bases de datos, conectividad con otros jugadores, etc. (ver capitulo 3), y el juego que puede estar desarrollado en cualquier lenguaje. La Fig. 4 muestra la arquitectura de este módulo.

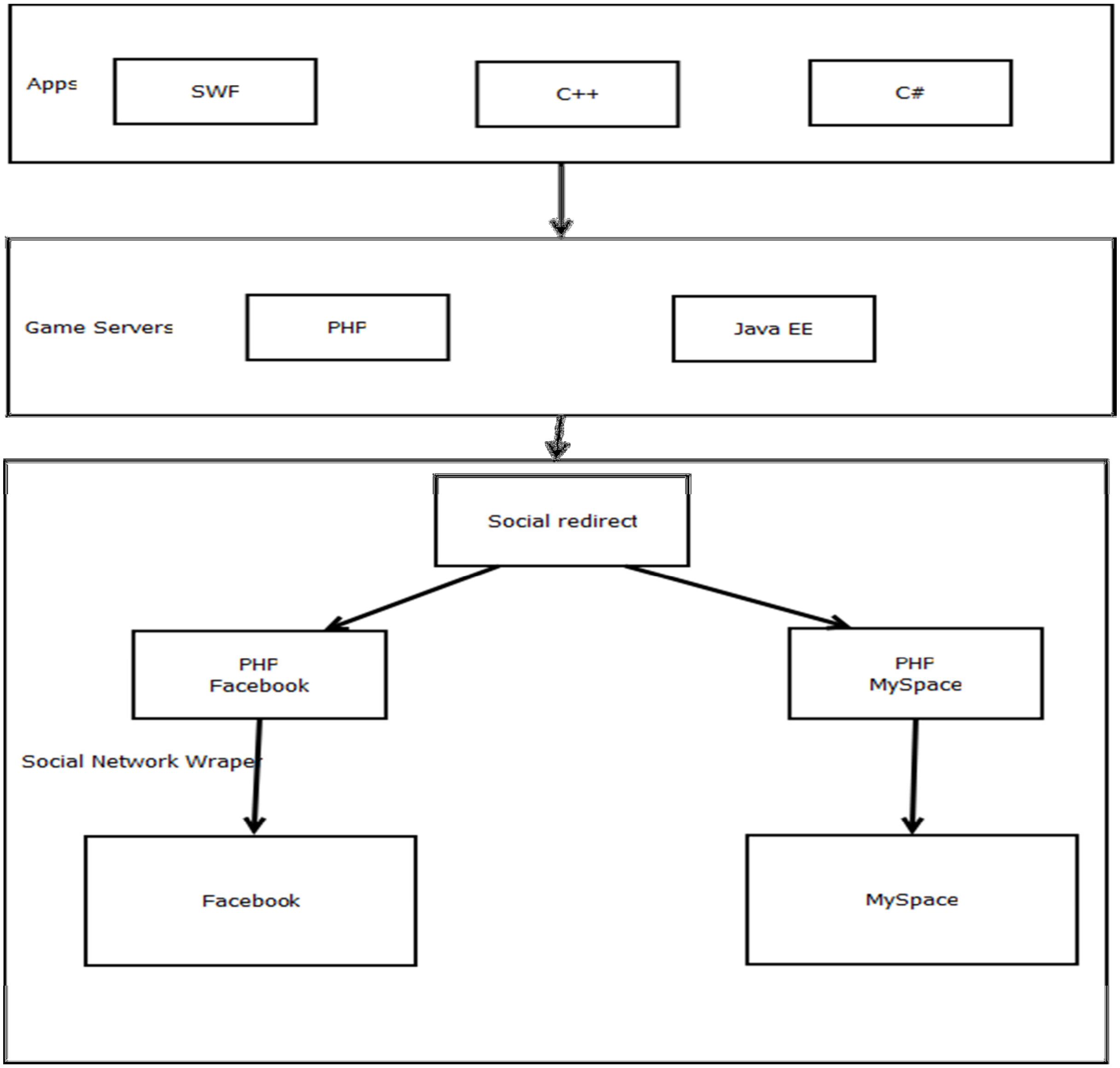


Fig. 4. Arquitectura del Módulo de comunicación con redes sociales y su relación con los otros módulos.

El módulo de PHP Facebook dentro del Social Network Wrapper(SNW) esta implementado en PHP, usando FQL, Old REST API y Graph API(se le dio prioridad a FQL por ser de las opciones mas rápidas, pero para el resto de la funcionalidad fue necesario usar los otros APIs).

Este módulo (SNW) puede ser usado por aplicaciones creadas en cualquier lenguaje (como se muestra en el módulo Apps).

SERVIDOR DEL JUEGO

El módulo intermedio (Game Servers) es opcional, un juego podría conectarse directamente con SNW, pero este es rara vez el caso. Con el fin de tener mayor modularidad, nuestro motor de juegos se conecta con un servidor, este servidor recibe las peticiones del juego y las que corresponden las reenvía al SNW, pero algunas de esas peticiones son para leer o escribir en la base de datos, (por ejemplo, para acceder al máximo puntaje). Además la comunicación entre el servidor y SNW es usando XML, pero si se tiene un juego cuyo lenguaje no soporta XML, el servidor del juego podría servir como traductor.

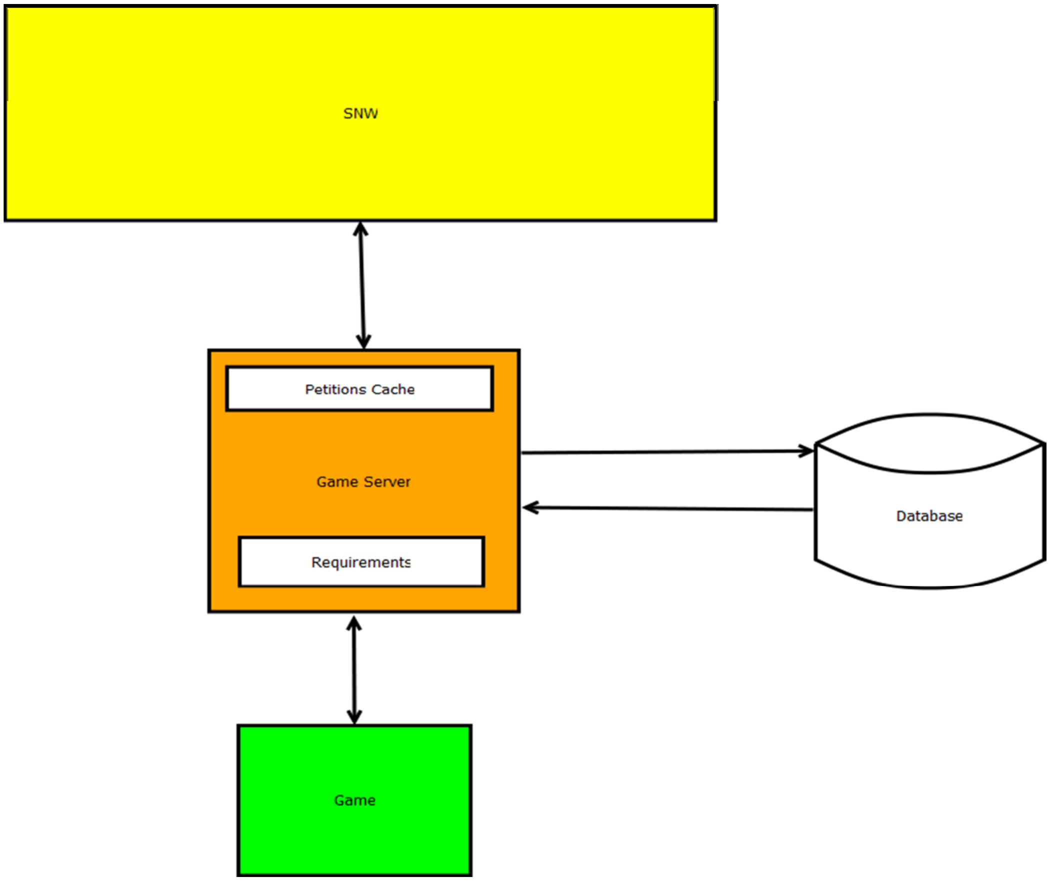


Fig. 5. Zoom al servidor del juego y sus relaciones con los demás elementos.

SUBMÓDULO DE REQUERIMIENTOS Y CACHÉ DE PETICIONES

Otro punto importante que lleva a cabo el servidor es el submódulo de Requerimientos y un Caché de Peticiones. Conectarse a FB es tardado y en menor medida, conectarse a la Base de Datos(BD) también, por lo que se deben minimizar las peticiones al SNW y a la BD, el caché justamente hace eso, si se hace una petición, guarda los resultados y en caso de que se pida la misma información más adelante, regresa su copia (esto no funciona si se quiere un juego que este siempre actualizado, pero para la mayoría de los casos no es importante que este actualizado todo el tiempo, o incluso si el servidor no responde, pero hay una copia anterior, se podría usar ésta).

El submódulo de requerimientos checa si se cumplen los requisitos para pedir algo a FB o a la base de datos (por ejemplo, si se quiere escribir un nuevo record, el submódulo debe checar si realmente es un nuevo record y si se debe o no escribir a la BD).

Un submódulo que se consideró pero no se realizó es el de verificación, (el cual podría estar contenido en el de requerimientos), y consiste en un módulo dedicado a verificar que el estado del juego y las peticiones sean correctas. Esto con el fin de evitar trampas en el juego, este submódulo consiste en algunas heurísticas e información muy específica a cada juego.

En la Fig. 6 se muestra un diagrama de secuencia de una aplicación en FB, el usuario entra a la página del juego dentro de FB, éste verifica que el usuario haya dado permisos a la aplicación, si no los tiene, FB le pedirá que autorice acceder a esa información, si no acepta el usuario, es regresado a FB, si el usuario acepta o si ya tenía permiso la aplicación, FB generará un "accesss_token" usando OAUTH 2.0 que enviará al servidor del juego, este servidor se puede comunicar con el SNW enviando este "access_token" y éste ya puede realizar peticiones a FB.

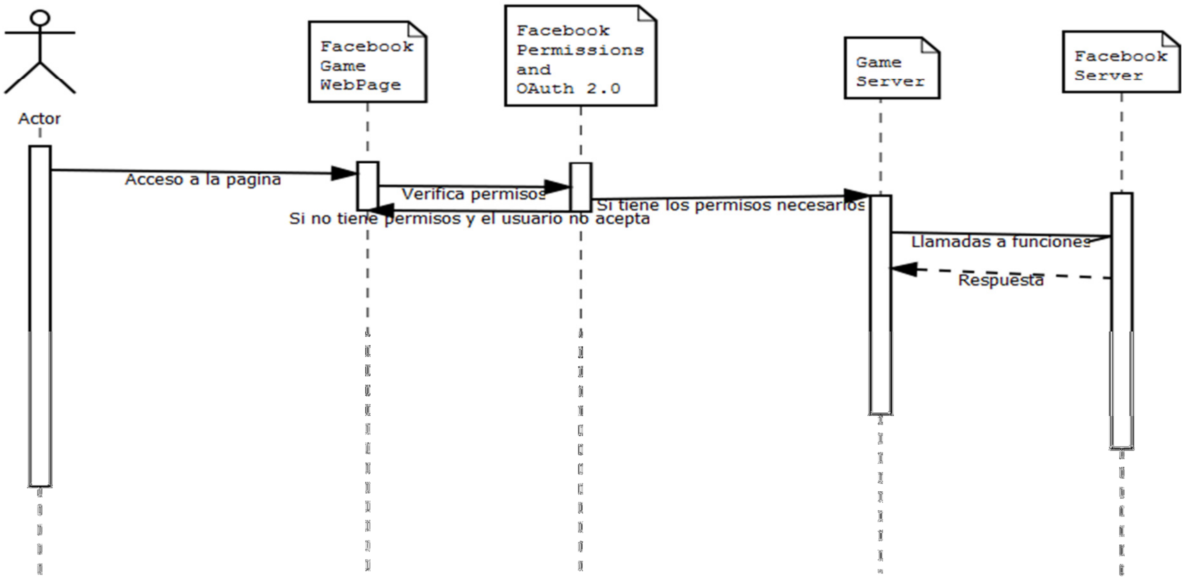


Fig. 6. Diagrama de secuencia de una aplicación en FB.

CAPÍTULO 8: DESARROLLO DE UN JUEGO UTILIZANDO EL MOTOR DE JUEGO Y EL MÓDULO DE COMUNICACIÓN.

El juego que Funcom (FC) quería desarrollar era un juego casual, pequeño y sencillo, pero que entretenga al jugador aunque sea por un corto tiempo y que motive al jugador a recomendarlo a sus amigos. El gameplay podría ser usado por otros juegos para crear juegos similares. Este es el primer juego desarrollado por FC para Facebook.

Este juego sirvió para probar a FB como plataforma de juegos, poder probar los pros y contras y por lo tanto no requiere soportar expansiones ni contenido descargable, se trata de tener un juego que sea lo más sencillo posible, pero que entretenga.

Además fue nuestro caso de prueba para el motor y servidor desarrollados. Este se programó a la par que estos, siempre teniendo en mente que código debía ir en el motor y que código en el juego. Esta distinción es muy importante, ya que un problema común es mezclar ambos códigos y terminar con un motor que no es lo suficientemente abstracto y por lo tanto no es usable para otros juegos.

Nuestro equipo contaba con un diseñador, además de tener la ayuda de otra diseñadora de FC y un productor, siendo todos relativamente nuevos en el diseño de juegos casuales, el diseño del juego sufrió muchos cambios durante el proyecto.

Nuestra diseño inicial fue crear un juego basado en Conan, ya que es la franquicia más importante y reconocida de FC, donde los jugadores crearían un avatar y competirían en ver quién es el más fuerte, usando un juego usado en ferias donde el jugador tiene que golpear con un mazo una superficie, la cual al ser golpeada aplica fuerza a una bola y ésta se mueve hacia arriba, qué tanto se mueve depende en la fuerza aplicada, así, a mayor fuerza, mayor altura.

FACTOR DIVERSIÓN

Basados en la tesis de Charles A. Butler [27], el equipo hizo un intento de clasificar y definir el factor diversión en juegos sociales y en nuestro juego. Esto es importante porque la audiencia de este juego es cualquiera que use FB, y siendo este un grupo tan grande y diverso, debemos entender que hace que la gente juegue y que se mantenga jugando, incluso si no acostumbran jugar otro tipo de juegos.

Empezamos con la categorización clásica de tipos de juegos hecha por el antropólogo Roger Caillois (Butler, 2009):

1. Agon: competencia y desafíos competitivos.

2. Alea: sumisión a la suerte.
3. Mimicry: juegos donde se adopta un rol.
4. Ilinx: vértigo y sensaciones físicas.

Mientras que la mayoría de los juegos exitosos en FB incluyen elementos de Mimicry y Alea, el elemento más prominente es Agon, pues el objetivo siempre es competir en contra de sus amigos en ver quién tiene el mejor puntaje, la mejor granja, etc. Y esto se puede observar en la barra de Rankings situada todo el tiempo en el fondo de la pantalla. El objetivo de estos juegos es llegar al primer lugar (y mantenerse ahí). El juego que desarrollaremos también caerá en la categoría Agon, similar a los demás juegos exitosos en FB como Farmville o Mafia Wars entre otros [28].

Al ser un juego casual, necesitamos tomar en cuenta las características de estos (explicados en el capítulo 3). Por lo que nuestro juego es sencillo de jugar (solo requiere 2 clics), no requiere mucho tiempo, cada nivel del juego consiste en 2 clics y pierde al perder todas sus vidas, aunque los primeros niveles son muy sencillos, la dificultad va aumentando hasta que el jugador eventualmente pierde.

También queremos que siga algunos de los 10 mandamientos (capítulo 3), como transmisores virales, hacer que el jugador pase tiempo en la aplicación, cargar tan rápido como sea posible, y recompensar al jugador inmediatamente. Y debido a que queremos que sea sencillo de programar será un juego independiente a la red.

ESTADO DE FLUJO

Lo anterior es importante porque nos ayuda a saber qué tipo de juego desarrollaremos y cómo es que será divertido para el usuario, pero en vez de intentar categorizar qué es divertido, usaremos los principios de “flujo” (flow) definidos por Mihaly Csikszentmihalyi.

Queremos que el jugador disfrute el juego, al grado de entrar al estado de flujo (un estado de concentración o completa absorción con la actividad y situación actual, durante la cual, las preocupaciones temporales son ignoradas, tiempo, comida, etc.)

Para poder entrar a ese estado, se requieren ciertos requerimientos:

- Una actividad que requiere habilidades.
- La mezcla o unión de acción y conciencia.
- Metas claras y retroalimentación.
- Concentración en la tarea actual.

- La pérdida de conciencia.
- La transformación del tiempo.

De esta lista podemos concluir que nuestro juego debe retar al jugador, pero éstos deben ser accesibles, si el reto es muy alto o muy bajo, se produce aburrimiento o frustración. También queremos que el jugador sienta que va mejorando, metas claras y retroalimentación al jugador son básicos en el diseño de juegos, especialmente cuando se enfocan a jugadores nuevos y finalmente queremos hacerlos sentir un grado de control mayor al real.

El juego está dividido en niveles, cada nivel siendo ligeramente más difícil que el anterior, hasta llegar a un punto donde deja de ser más difícil (para no llegar a un punto donde sea imposible ganar).

Los primeros niveles son bastante sencillos para que los jugadores principiantes puedan acostumbrarse al sistema de juego, pero para no aburrir a los más expertos, introducimos un sistema de bonus, si el jugador logra 5 golpes perfectos, consigue una vida extra, que le será muy útil en los niveles más avanzados, haciendo que los jugadores expertos se concentren en estos primeros niveles en vez de solo ser niveles de paso.

Además el jugador puede retar a sus amigos, compitiendo por el máximo puntaje, esto obliga al jugador a intentar conseguir golpes perfectos la mayor cantidad de veces, y con estos retos, promovemos el juego, ya que el jugador querrá retar a sus amigos, escribiendo en sus muros que han sido retados y además ofrece un reto diferente, en el modo de juego solo, uno puede conseguir un máximo puntaje después de reintentar muchas veces, pero cuando se reta a un amigo, el puntaje que logre en esa partida es contra la que competirá su amigo. Además agrega una segunda tabla de puntajes, el jugador no solo compite por el mejor puntaje entre sus amigos, sino también por la mayor cantidad de victorias y menor cantidad de derrotas en duelos.

En cualquier tipo de juego la retroalimentación es muy importante, el jugador debe ser capaz de saber qué está pasando de manera sencilla, esto se logra de muchas formas en juegos, animaciones, partículas, sonidos, textos, etc. Pero en el caso de juegos casuales es todavía más importante esta retroalimentación, pues estamos tratando con mucha gente que jamás ha jugado antes, por lo que no podemos asumir ningún tipo de conocimiento de su parte y tenemos que mostrarle retroalimentación a cualquier evento del juego, para evitar que se pierda en el juego, o que crea que no está funcionando correctamente, además de que son una forma de llamar su atención y lograr que usen la aplicación durante más tiempo.

En la parte de la retroalimentación, usamos textos animados para indicarle que tan bien lo hizo, además de que Conan y el pict también reproducen una animación basada en el desempeño.

Como se mencionó en el capítulo anterior, también se creó un servidor que funcione como intermediario entre el SNW y el juego. Para este juego programamos un servidor que recibe una petición del juego (peticiones como escribir en tu muro, actualizar un máximo puntaje, conseguir los amigos del usuario, conseguir los puntajes de tus amigos, etc.) y el servidor evalúa cada petición y se conecta a la base de datos o al SNW o a ambos dependiendo la petición. Este servidor es independiente del SNW, por lo que cuando el SNW soporte otras redes sociales, este podrá ser usado con mínimos cambios.

Este juego requiere 2 tablas en la base de datos:

- La información de cada jugador, donde se guarda su puntaje máximo, sus victorias y derrotas en los duelos (hay cierta información relacionada con FB que por políticas de seguridad de ésta no se pueden guardar).
- Todos los duelos efectuados contiene el identificador de ambos jugadores, los puntajes de éstos (el del retado es -1 si no ha aceptado el duelo) y la fecha en la que el duelo fue solicitado (esto con fin de poder eliminar retos que lleven mucho tiempo sin ser aceptados).

Los amigos de cada jugador no pueden ser guardados, por lo que al inicio de la aplicación ésta tiene que pedir la lista de amigos a FB, después con esa lista se conecta al servidor del juego para pedirle los puntajes de esos amigos, y si el jugador selecciona la opción de duelo, se efectúa otra petición para obtener la lista de todos los duelos relacionados con él.

En las siguientes páginas se puede ver una versión previa del juego y la versión final del juego.

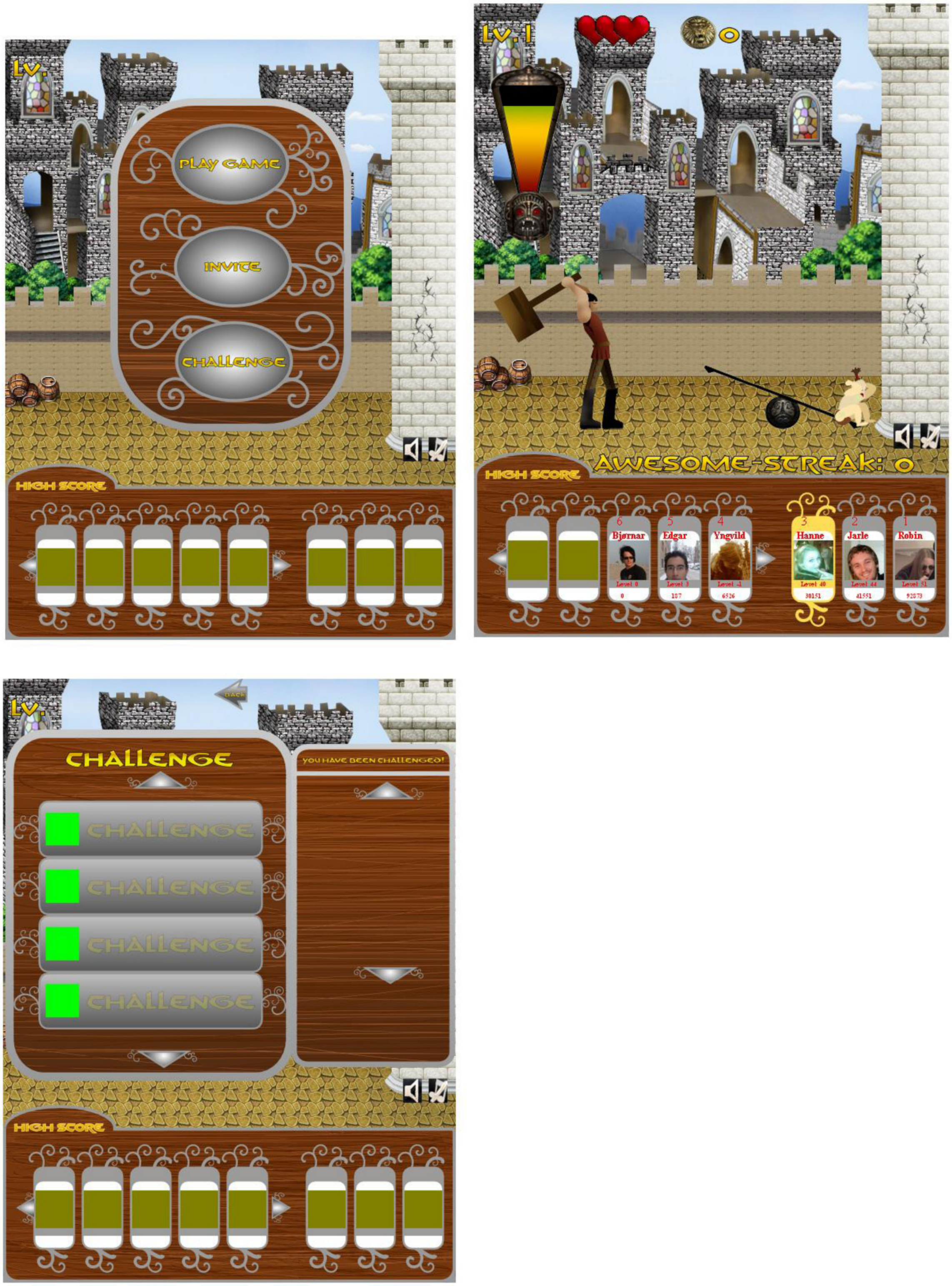


Fig. 8. Imágenes de una versión beta del juego.

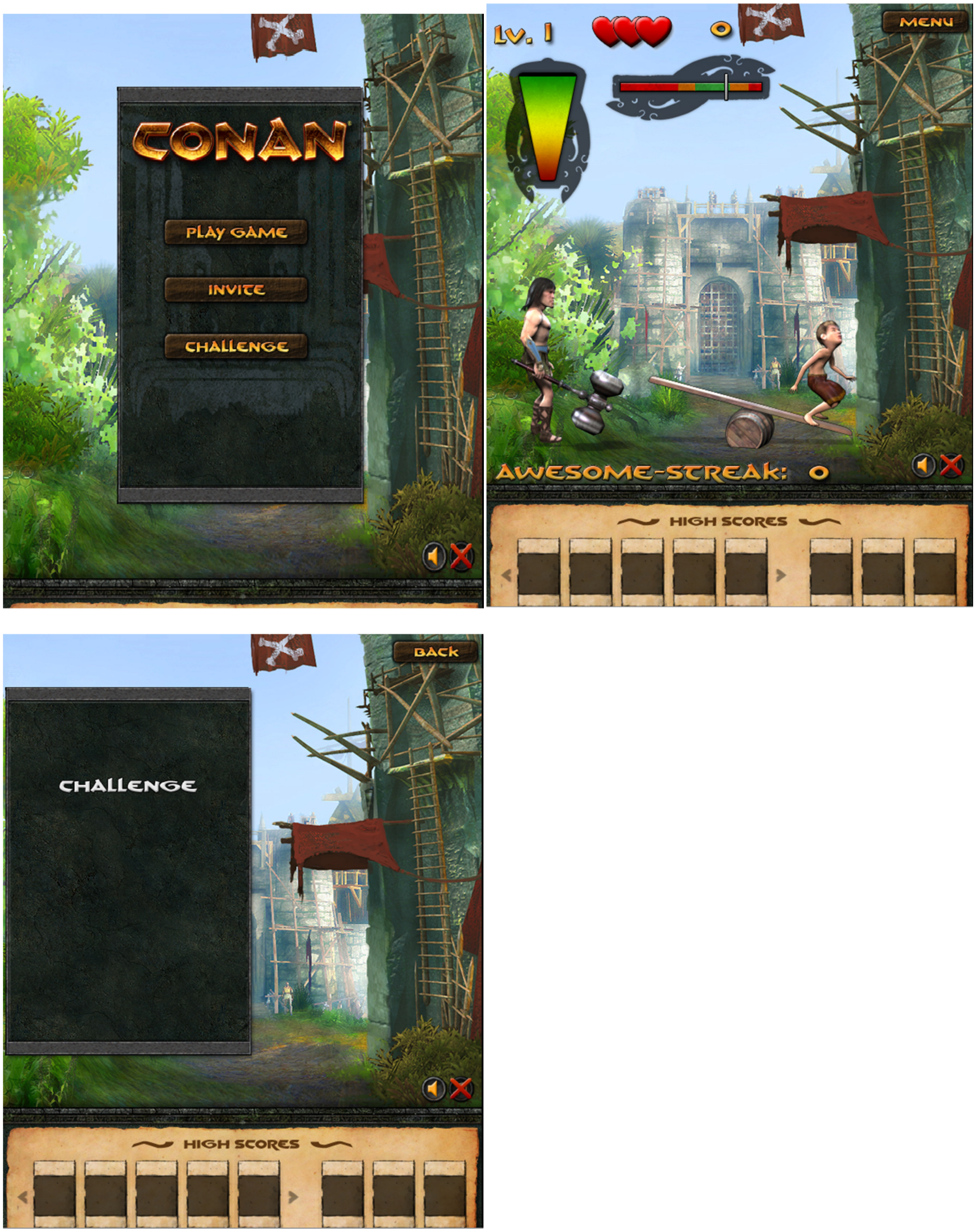


Fig. 9. Imágenes de la versión final del juego.

CAPÍTULO 9: CONCLUSIONES.

Llegar a un diseño sencillo generalmente es una tarea difícil, el motor de juego pasó por varias etapas donde se cambiaban las clases y sus relaciones, y se trataba de hacer que funcionara lo más rápido posible sin perder generalidad.

El módulo de conectividad con FB fue hecho con Java Servlets, después se reimplementó en PHP cambiando las bibliotecas de comunicación varias veces. También se usó JSON para la comunicación entre servidores, implementando un parser en el servidor del juego que convertía JSON a XML que finalmente no se usó porque el servidor de FB usa XML para sus respuestas.

El desarrollo de un juego siempre involucra cambios, a pesar de que la idea final es muy sencilla, primero se consideraron e implementaron sistemas de bonus, personalización de avatares, rounds dentro de niveles, diferentes tipos de retos y tablas de puntajes para los diferentes niveles y otros, pero al final terminó siendo demasiado complicado para el jugador casual. Como usuario de juegos de video con una vida de experiencia, a veces es difícil pensar como alguien que no le dedica el mismo tiempo, que si no logras llamar su atención en los primeros segundos jamás volverá a jugar. El proceso del diseño del juego sufrió muchos cambios hasta llegar a esta idea simplista, pero que mantiene el elemento del reto y desafío a tus amigos usando una tabla única de puntaje y un sistema único de retos. Creo que logramos aislar lo más importante del juego y crear un juego sencillo, pero que tenga reto para jugadores con más experiencia, y a juzgar por las reacciones en las oficinas de FC creo que el objetivo se logró.

Comunicarse con FB no es tarea sencilla y requiere mucho mantenimiento al código debido a la naturaleza mutable de los entornos de desarrollo de FB. La latencia y errores fueron otro problema, muchas veces se cambiaba código pensando que había cometido un error, solo para darme cuenta varias horas después (y muchas líneas de código) que el código original estaba bien, simplemente FB no respondió en ese momento. Pero al final se logró implementar un servidor que pudo funcionar para aplicaciones hechas en C, en Java y en AS3, y que con el mantenimiento apropiado puede seguir funcionando sin necesidad de cambiar el código de los programas que lo usan. (y en teoría funcionará para cualquier lenguaje que pueda hacer una petición GET o POST).

Trabajar en una compañía profesional de juegos de video me ayudo a comprender el proceso detrás de un juego, no solo la parte técnica de programación, sino también en el aspecto gráfico, de diseño y producción. También tuve la oportunidad de convivir con personas de otras áreas de la empresa y aprender un poco sobre cómo es su trabajo, que aunque no siempre eran parte del área técnica del juego, de todas formas son parte fundamental de este.

Durante mi estancia en NITH tome un curso sobre Diseño y prototipos de juegos, participe con otros estudiantes en el proceso de selección para Dare to be Digital [31], y actualmente trabajo con un diseñador egresado de NITH en la compañía DeRailGames [32].

REFERENCIAS

[1] Funcom Company Info.
http://www.funcom.com/wsp/funcom/frontend.cgi?func=publish.show&func_id=1007&table=PUBLISH&item=1007

[2] At Last -- The Full Story Of How Facebook Was Founded
http://www.businessinsider.com/how-facebook-was-founded-2010-3#we-can-talk-about-that-after-i-get-all-the-basic-functionality-up-tomorrow-night-1

[3] Who will be Facebook's next 500 million?
http://news.cnet.com/8301-13577_3-20011158-36.html

[4] Facebook Developers Introduction
http://developers.facebook.com/docs/api

[5] Facebook Platform
http://www.facebook.com/platform

[6] Zinga website
http://www.zinga.com

[7] Zynga Hits 100 Million Unique Visitors Per Month
http://www.zynga.com/about/article.php?a=20091123

[8] Playfish website
http://www.playfish.com/

[9] Playfish vs. Zinga: The never ending copying battle
http://www.thefacebookforum.net/zynga-f117/playfish-vs-zynga-the-never-ending-copying-battle-t1898.htm

[10] Who has the biggest brain? web site
http://www.facebook.com/whohasthebiggestbrain

[11] Platform Live Status
http://developers.facebook.com/live_status

[12] Casual game definition
http://en.wikipedia.org/wiki/Casual_game

[13] Action Script Technology Center
http://www.adobe.com/devnet/actionscript.html

[14] Adobe Flex
http://www.adobe.com/es/products/flex/

[15] PHP website
http://www.php.net/

[16] Java Servlet Technology
http://www.oracle.com/technetwork/java/index-jsp-135475.html

[17] The OAuth 2.0 Protocol
http://tools.ietf.org/html/draft-ietf-oauth-v2-10

[18] Old REST API
http://developers.facebook.com/docs/reference/rest/

[19] Facebook Query Language
http://developers.facebook.com/docs/reference/fql/

[20] Facebook Markup Language
http://developers.facebook.com/docs/reference/fbml/

[21] JavaScript SDK
http://developers.facebook.com/docs/reference/javascript/

[22] Doom
http://es.wikipedia.org/wiki/Doom

[23] Unreal Technology
http://www.unreal.com/

[24] Typing Maniac web site
http://www.facebook.com/apps/application.php?id=66258690527

[25] App Hub
http://create.msdn.com/en-US/

[26] Mahajan, Amitt (2010). Rapidly developing Farmville: how we created and scaled a #1 Facebook game in 5 weeks, GDC 2010

[27] Butler, Charles (2009). Effect of Success versus Other Players on the Perception of Fun, Guildhall at SMU.

[28] Top 10 Most Addictive Games On Facebook.
http://www.tricksdaddy.com/2009/12/top-10-most-addictive-games-on-facebook.html

[29]Bergström, Andreas (2008). Software Configuration Management in Scrum Projects, Lund University.

[30] Brathwaite, Brenda (2008). Challenges for Game Designers, Charles River Media

[31] Dare to be Digital
http://www.daretobedigital.co.uk/

[32] DeRailGames
http://derailgames.no/

APÉNDICE A: REPORTE ENTREGADO EN NITH

NITH

Norges Informasjonsteknologiske Høgskole

Tilgjengelighet:

□ *FRI*

X BEGRENSET

Fagbetegnelse:

PJ600

Gruppenummer (dersom besvarelsen leveres i gruppe):

3

Tittel norsk:

Conan Strongest Man Challenge Game

Tittel engelsk (Hovedprosjekt):

Conan Strongest Man Challenge Game

Innleveringsdato:

19.05.10

Semester:

Spring 10

Oppdragsgiver:

FunCom

Antall sider: 61

Antall ord: 15392

Sammendrag (fra rapporten):

Our task was for this project was to create a flexible challenge system for Facebook games, prototyping one small challenge game with it.
FunCom wanted a result that would allow them to create a variety of small challenge-type games by keeping the same code and just altering the graphical elements.
The group has accomplished this and while the current graphical elements do not meet with FunCom's standards, both the group and the employers are satisfied with the prototype, which runs smoothly, connects with Facebook and has a simple and fun gameplay.
The current status of the game is that a team of professional graphical artists will evaluate and possibly polish it so that it can be seriously considered for release on Facebook sometime in the future.

Arbeidet avsluttet: 19.05.10

Navn:	Studentnummer	Signatur
Hanne Lien	245419	______________________________
Robin Berg Pettersen	802974	______________________________
Edgar Omar Cebolledo Gutierrez	700006	______________________________

*Studentenes signatur er også en bekreftelse av at gruppen har gjort seg kjent med, og fulgt,
NITHs retningslinjer for intellektuell redelighet (tilgjengelig på intranett).*

Foreword

This report is written as part of the final Bachelor Project done by two students from NITH and one exchange student from UNAM (Mexico City, Mexico). The project group consists of Robin Berg Pettersen (Game Programming, NITH), Hanne Lien (Game Design, NITH) and Edgar Omar Cebolledo Gutierrez (Game Programming, UNAM).
As required by NITH the group came in contact with the employers by their own efforts and on their own initiative.

Our project was completed together with FunCom's social games division in FunCom's main office building located at Skøyen outside of Oslo. The group was tasked with producing the first game to be published on Facebook by FunCom and the freshly started social games division.

The group would very much like to thank all members of this division for making us feel welcome at FunCom, as well as for all their help and guidance during the project period. Thank you to Jarle and Yngvild for being such fantastic project managers and for helping us manage time and prioritize tasks, and to Jørgen and Charles for being our programming gurus with apparent infinite knowledge about their intricate field of expertise.
We would also like to thank our internal supervisor at NITH, Faltin Karlsen, who has been a source of support, advice and constant enthusiasm.
The group would also like to thank Truc Trung Duong, a fellow 3rd year game design student, who was kind enough to help us out in a pinch by producing excellent graphics for us, even though he was already busy working on another project together with his own group.

This report is primarily meant to be read by NITH's internal sensors. Prior knowledge to programming or game design should be superfluous, but is recommended. For the sake of your convenience, a Nomenclature can be found in a separate appendix, explaining certain terms.

Signed:

__________________ ____________ ____________________

Robin Berg Pettersen Hanne Lien Edgar Cebolledo Omar

Index

Contents

1.0 Introduction

1.1 Presentation of the group

Hanne Lien and Robin Berg Pettersen, both 3rd year students at NITH within the respective fields of game design and game programming, had already decided to work together on the bachelor project and was on the look-out for a 2nd programmer. They had previously worked together on various school related assignments and were confident in each other's skills and dedication. It was thanks to Hanne's fellow classmate Bjørnar Herstad, that the group was put in contact with the Edgar Omar Cebolledo Gutierrez, a game programmer from UNAM in Mexico. He had already agreed to enter an exchange program and come to Norway in order to work with Bjørnar – we were originally supposed to work together as a 4 man team – but when Bjørnar was offered to work on a different project at FunCom, he accepted and the team finally ended up being 3 people.

Edgar has been working primarily on the server and databases and making the necessary Facebook connections, Robin has focused on implementing gameplay and graphics, while Hanne has been responsible for the design, documentation and temporary 2D graphics.

We would like to offer Bjørnar Herstad our sincerest gratitude for giving us the opportunity to work at FunCom!

The group has also been fortunate enough to receive the help and support from Truc Trung Doung, another fellow-classmate of Hanne, who has been of tremendous value to us, helping the group out with GUI and menu graphics while he was working on his own bachelor project here at FunCom.

1.2 Presentation of the employers

FunCom was founded in 1993 and is currently the largest game development company in Norway, with over 300 employees from 35 different nations and offices in Switzerland, China, the USA and now recently Canada as well, in addition to their main office in Oslo. They have produced over 25 game titles across several different gaming platforms, most notably the award winning adventure game "the Longest Journey" and its sequel, "Dreamfall", as well as the two Massive Multiplayer Online (MMO) games "Anarchy Online" and "Age of Conan". With these they have asserted their position internationally and they are currently one of the world's most experienced developers of MMO-games.

In 2005 they also became a publicly listed company on the Oslo Stock Exchange.

In addition to MMO games, FunCom also has a department for Casual Games and a brand new department for Facebook-games, the latter in which our project group have worked in as interns.

The project group came in contact with FunCom at the last moment through a fellow NITH 3rd year student, Bjørnar Herstad, who was also hired by the company for this project as a Design intern and research assistant on FunCom's most recent MMO title, “The Secret World” (Release date unconfirmed). The group's lead designer has been in contact with Jarle Snertingdalen, the group’s internal project supervisor at FunCom, since the end of November 2009 and started working on the pre-production of the first assignment.

1.3 Presentation of the project

FunCom, while commonly associated with big titles and similarly big productions, saw the potential in the growing market of smaller, browser-based online games applicable to social networking platforms such as for instance the highly popular Facebook, and in November 2009 they put together a small group of experienced designers and programmers to form a Social Games Division. The goal of this division is to work towards the publication of one or more such games.

The primary goal of this young division is first and foremost to get a small game published on Facebook to test its potential, and so the project group, consisting of Hanne Lien as lead-designer and graphic artist and Robin Berg Pettersen and Edgar Omar Cebolledo as programmers, has been tasked with the development of their first release. The group will be working as a semi-independent part of this division, providing all design, place-holder graphics and code required to completing the game with the help and guidance of the talented people of the Social Games division.

1.3.1 Project definition

The game the group has been asked to make is supposed to be small and uncomplicated, while still providing the player with an instant-fun/instant-satisfaction type of gameplay-experience that they will want to play for a short while and recommend to their friends. With its envisioned narrow scope in mind, the project may include more than one game, provided the first one is completed quickly and completed well. While the gameplay may be re-used at a later time to create similar or semi-similar types of games in the future, this first game is meant primarily to be a test of using Facebook as a gaming platform and is thus not meant to have particularly long-lasting gameplay, and no further expansions such as release of new content is planned or intended at this point.

“How can we make a Flash-based game for a social networking site that is small in size and has a type of gameplay that is at once instantly understandable and instantly enjoyable and that will make players want to play it again and again and recommend it to their friends as well?”

Our specific task was to make the game fit the theme of a Conan Strongest Man Challenge Game. The idea for the gameplay was initially to have players create their own avatar that would compete to prove he/she was the strongest of all, by making use of the familiar concept of having a strength meter such as one might find at a carnival or a fair where the point is to hit the meter as hard as you can with a sledge hammer to make a ball or some similar object go as far up as

possible.
Several possible variations were suggested that might fit with the planned core gameplay mechanics (e.g. bow and arrow, throw a Pict etc.), but these were all eventually discarded and the group decided to go through with the original idea with the hammer and the strength meter in a Conanesque setting.

1.3.2 Target platform

For the past couple of years the popularity of online social platforms has sky-rocketed and expanded vastly in both size and number of active users. But also in terms of diversity has this media developed. Users can now not only expand their social network and add their friends, post pictures and send each other messages; they can also play games within the Facebook framework [appendix 4]. This endeavor has proven to be extremely successful and users worldwide have become completely addicted to such games as FarmVille and CaféWorld, both produced by the social games company Zynga.

These games are free to download, easy to learn and easy to play and awards players for inviting their friends to join, thus ensuring that word of their game gets spread around. In addition players can use their real money to purchase in-game cash, which can be used to buy special items and other content. This way of making money from a game is called micro-transaction and is rapidly becoming a standard within the online gaming community.

1.3.3 Vision, main goals and sub-goals

The group’s vision for this project is to develop a game that not only meets with the employer’s given requirements but which can also help the newly founded FunCom’s Social Games Division gain a wider perspective on the pros and cons to developing games for this particular platform. If the group can make Facebook users begin to associate FunCom with enjoyable, well-made browser games then they will have reached their goal.

Our main goal for this project is to create a product that lives up to our employer's expectations, meets their requirements as specified and that is of such quality that it can be released on Facebook within the set time frame. Additionally we aim to have all the required documentation ready to be handed over to NITH and FunCom respectively, with May 19th 2010 being the final deadline for delivery in both cases.

In order to accomplish what is expected from us, the group has focused to working effectively and as independently, finishing our tasks quickly but never sloppily and solving the problems we encounter internally within the group as far as this is possible. While working towards developing a great game, as should be the goal of all game developers, the group will keep in mind what the game is meant to be and thus not make it any more or less elaborate than it was ever planned to be.

Our sub-goals include:

- Keeping it simple! The group will take great care to avoid feature-creeps, in accordance with FunCom's wishes to keep it as simple as possible

- Creating the required content for the game

- Implementing the required functions to make the game play as it is supposed to

- Testing and tweaking the game after completion to ensure that what the group has made is of the desired quality

- Mastering the tools required to complete the game as planned. This includes Adobe Flex, which is a Flash-based programming toolkit, and Photoshop CS3, which is a tool for working on pictures but can also be used for making them

- To give each group member a clear insight into the various stages and sides of the development process as well as the opportunity to participate and further expand their own knowledge and abilities

- Keeping the flow of information constant and relevant, using the tools Skype for chatting, Social Wiki for updating the design documents and WinSCP to share files between computers

1.3.4 Limitations

As previously stated the group will not make the game bigger or more complex than it needs to be. With this focus in mind, the group will limit the scope of their project in the following ways:

- There will be no extra content generated for the game after it has been released

- The game will not be expanded in any way after it has been released

- The game will be developed for release on Facebook using the Adobe tool Flex

- The group is responsible only for producing the game itself and has nothing to do with other aspects such as promoting it or putting up a web-site for it, should there turn out to be a need for either of these

1.4 Relevance to scientific research

Since the group will be working on making casual games to be published on social networking platforms such as Facebook, which is a relatively new platform as far as gaming is concerned, there is little documented research to base the technical part of our work on.

1.4.1 Casual games

The casual games sector is one of the fastest growing segments in the gaming industry today, but currently very little research has actually been done on the subject. One of the few articles out there about this particular genre is “Casual Games Discussion” [1] written by a Finnish team of 4: Jussi Kuittinen, Annakaisa Kultima, Johannes Niemelä and Janne Paavilainen. The goal of their research was to gain deeper insight into what it is that makes a game ‘casual’ and how the term ‘causal’ may apply to any number of game-related subjects that are not to be confused with one another, such as ‘casual gamer’ vs. ‘casual gaming’. To briefly sum up what makes a game *casual*, these genre include the games that are often relatively small in size, have a type of gameplay and controls that are easy to master, they are not very complex and generally require very little time and dedication from the player in order to get the full experience. Famous examples include the mobile game Snake, Minesweeper, FarmVille and Peggle.
The description makes these obvious candidates for distribution through websites, where people are used to simply moving on to the next link the second they get even remotely bored, and social networking sites have the additional benefit of offering a way for games to spread. While online distribution of browser based games might be the cheapest and possibly most effective solution, casual games are also vastly popular on other platforms such as mobile phones and consoles with online shops (e.g. Window Live and the PS3 Store).

1.4.2 About games as Fun

Based on the thesis written by Charles A. Butler [6] the group has made an attempt to classify and define the fun-factor in both social games in general and our game specifically. The reason why this is relevant is that the group is aiming for an audience that consists of more or less ‘everyone’, because ‘everyone’ is on Facebook, and therefore it is important to be conscious of what it is that makes these people play and continue to play when most of them don’t consider themselves to be gamers at all.

To start with the classic categorization of types of games by anthropologist Roger Caillois (Butler 2009):
1. Agôn : competition and competitive struggle
2. Alea: submission to the fortune of chance
3. Mimicry: role-playing and make-believe
4. Ilinx: vertigo and physical sensation

While many of the currently most successful Facebook games (FarmVille, MafiaWars, etc.) include minor elements of both Mimicry and Alea, the most prominent common element seems to be Agôn, because of how people can compete against their friends and fellow players about who gets the highest score or has the biggest farm, all of which is then displayed on a leader board (high score list). Everything you do in the game is for the purpose of moving higher up on that board, past your friends. This will also be the case with the game the group is making.

Knowing this it is important to establish how this type of game will appear as enjoyable to the players. It is, of course, as good as impossible to produce a satisfying definition of what 'fun' is, as there are so many ways to be entertained. Laughing at a joke is not the same type of fun as you have attending a fun party, for instance, or when you're driving really fast. So rather than categorize fun, the group has chosen to consider the principles of flow, as defined by Mihaly Csikszentmihalyi.

As game designers we want the player to enjoy himself so much that he enters a state of flow, and Csikszentmihalyi has specified the following as requirements for entering such a state:
1. A challenging activity that require skill
2. The merging of action and awareness
3. Clear goals and feedback
4. Concentration on the task at hand
5. The paradox of control
6. The loss of self-consciousness
7. The transformation of time

As it relates to games, this list provides us with a number of things to consider and concentrate on when designing a game – particularly the 1st, 3rd and 5th points. We want the player to be challenged and feel that they are getting better at what they're doing, clear goals and feedback to the player should be a goal to all game-makers, especially when dealing with relatively fresh players, and finally we want them to experience a sense of greater control than they actually have.

1.4.3 The 10 Social Apps Commandments

Casual games intended for social networking platforms is still a relatively fresh idea, and the social games division at FunCom has done a lot of research [2], working out their own strategies for how these games can and should be made in order to be successful. Here is a list created by Jarle Snertingdalen, producer on the social games division:

The 10 Social Apps Commandments:
1. Spread – design viral transmitters first
2. Omniscient – use analytics on everything
3. Conversion – make users an offer they can't refuse
4. Investment – players must invest time to be hooked
5. Accessible – give users instant access
6. Longing – get players to return every day
7. Accomplishments – give users instant gratification
8. Patch weekends – users play Monday to Friday, 09 – 17
9. Patch frequently – fresh content every week

10. Simplicity – keep the game simple
[appendix 5]

These points are based on research that has been done on other social games such those produced by Zynga, the current giants of this market, and includes the 10 essential advices that apply to basically all parts of the project group and process, including maintenance such as patching.

2.0 Method

2.1 Choice of development process framework

The choice fell quickly on SCRUM, an increasingly popular framework that divides and measures the total workload of a project into smaller iterations. SCRUM is commonly used for managing complex work, such as the development of new product software, and is thus ideal for game developers. The group's members have previous experience with this framework from the subject PJ311 which is a mandatory subject for all 2nd year students at NITH, and found that although it had certain, minor flaws when used for game development it was still the best option for the project.
Another important reason that made the choice very easy for the group was that it was already the standard process framework in use at FunCom, meaning that would likely have become a profound dreg on the overall efficiency of the project had we selected a different method.

SCRUM is a good choice because it takes into consideration that requirements for the product might change up to several times before the final delivery and also facilitates continuous testing and optimizing of gameplay little by little by demanding a potentially shippable product by the end of each sprint.
As the size of game development teams and projects continues to expand the use of Agile Methods such as SCRUM has proven to be a great way to manage both the team's members and the project's development process [5].

That being said, the group would like to point out some of the problems with SCRUM as seen from a game-developers point of view.
First and foremost, one of the main ideas of SCRUM is that the production team should consist of members with cross-functional skills, as the SCRUM process traditionally only recognizes 3 different roles: product-owner, SCRUM-master and the SCRUM-team [4]. But this situation is something one will only very rarely come across on any game-development team, where the respective members usually have little or no clue about how to do each other's jobs, and having a greater number of clearly defined roles is literally vital to any project. What this essentially means is that while working together as a team, the individual members depends very much on each other to complete the project and that the project cannot be completed unless each person does his or her share of the given tasks.

2.2 Choice of development software

The process of developing a game can, very roughly, be divided into the following:

- Designing the overall idea and system for the thought game
- Creating the different graphical elements and sound effects
- Writing the code to implement these and make the game play as intended

In order to effectively accomplish this, the group needed to make several choices regarding the choice of different software.

2.2.1 Programming tools

We used Flex for development of flex applications. Flex is a software development kit released by Adobe Systems. It is mainly used for linking ActionScript-files together into a running .swf-file, which can be run in web browsers or via any Flash player. Flex and Flash are pretty similar, but they appeal to different user groups. While Flash is mainly made for designers, Flex is made more for the programming users. But as a programmer you will get more possibilities as you can tweak and make your own versions of the files and functions that are included in the Flex-library. Another big advantage by using Flex is the possibility to easily merge code with other programmers by using third party software, which is really useful when there are more than one programmers working on a single project.
Flex is an open-source project, which means it is available for everyone, free of charge!
This makes it cheaper for the developer, because Flex-applications can be developed using any standard IDE(Integrated Development Environment). Adobe Flash on the other hand is not free, but it offers a lot of features that are useful for designers, like drawing vector graphics, making movie clips and so forth. However, most programmers couldn’t care less about these features, and in our case it’s not necessary to have them.

2.2.2 Design tools

the design documents, including the pitch put up during the pre-production phase and the system design was all put directly into the social media wiki site; one for the pitch alone and one for the game design and system design.
This is the standard at FunCom and works excellently because it can be easily accessed and has an intuitive interface which makes the site easy to navigate. It did, however, take some getting used to for someone who is used to working with Windows Office Word, which has an almost completely automatic way of structuring documents. The wiki syntax does not facilitate this in the same way, and it took some time figuring out how to make the documents quick and easy to read while at the same time managing to emphasize the points that were essential for the reader. For such a small game in particular this was an important point, as nothing that was written in the system design was really superfluous.

2.2.3 Graphical design tools

For the task of making the graphical components and animations the group decided to go for Photoshop. FunCom already had a licensed version of Photoshop CS3 and plenty of graphical artists who use this as their standard graphics tool and can be asked for help. Another important reason for choosing Photoshop over any other tool was that the group's graphical artist had some experience with it and would have an easier time working with it, rather than having to learn something new.

Another tool that was used for creating the visual graphics was the free, open-source program Inkscape. Inkscape uses vector-graphics, which basically means that after a line or shape has been drawn it can be easily scaled up and down without affecting the image quality, and it is easy to quickly fix mistakes. Inkscape also provides simple ways to create shapes directly, whereas in Photoshop it is a lot more time consuming and difficult to correct upon later.
Inkscape was used primarily by the group's extra member, Truc Trung Duong, who felt the most comfortable working with this tool.

2.2.4 Communication tools

2.2.4.1 Skype

For chatting with each other in the office during work hours, the group used the free communication application Skype. This is standard at FunCom and even though the three team members sat close together and often solved issues and answered questions verbally, the point with having a tool such as this was to avoid making unnecessary noise in an office shared by several teams. In addition to being able to have private, soundless conversations Skype offers a quick and safe way of sending files that came in particularly useful when the graphics were being made and it was possible to comment on mock-up graphics instantly, thus saving the graphic artist a lot of unnecessary work.

2.2.4.2 Mozilla Thunderbird

For sending mails within FunCom the e-mail reader, Thunderbird, was installed and the Facebook team (including the group) would use this as a means of sharing slightly longer and more important messages than what one might normally convey through the Skype-chat.

2.2.4.3 Dropbox

For sharing files strictly related to this report the group made use of the online file-synchronizing tool, Dropbox. This tool creates a separate folder that can be accessed from any computer anywhere and it is very useful when working on a project where all team members are required to contribute and stay up to date. This way the group was able to set up a folder where all documents could be read and updated consequently.

2.2.4.4 MediaWiki

MediaWiki is a free software wiki package written in PHP that was originally for use on Wikipedia. It offers all the same possibilities as any Wikipedia page and is particularly useful in terms of providing the reader easy navigation. Text formatting did turn out to be somewhat lacking, but in terms of space and general set-up it was more than sufficient.

2.3 Techniques used

2.3.1 Daily SCRUM

This daily session is an essential part of the SCRUM project planning method, and has proven to be an excellent way of keeping all team members up to date on how the project is progressing in all areas. The group held their daily SCRUM meetings at 09:15 sharp every morning at FunCom together with FunCom's social games division members. Here we spent roughly 15 minutes where each of us summed up what they had done the previous day and what they would be working on this day. All social issues regarding team work and cooperation, although none arose during this specific time period, should also be brought up during these meetings so that they may be solved in plenum.

2.3.2 Jira and biweekly iterations

Jira is a proprietary enterprise software product developed by Atlassian. It is commonly used for project management and bug- or issue-tracking. It features a convenient way of creating a product back-log and assigning tasks to various iterations, as well as ranging the individual task's difficulty and commenting on them. It is used at FunCom and the group found it to be a very useful tool, especially the designer, as it allows you to keep control over all your remaining tasks as well as how close you are to finishing it.

Every 2nd Tuesday the group would revise the past iteration and assign new tasks to the next iteration from the product back-log. According to the contract the group would spend Tuesday, Wednesday and Thursday each week working at FunCom, and with a 3 day week it felt pointless to operate with 1-week iterations and so we decided to work with 2-week iterations instead.

2.3.3 Project plan

The group was already provided by a very general schedule for the project period by NITH, with dates and details concerning the written report and the oral presentation. As for the making of the game itself, a complete project plan was not made, except for the first report delivery at the end of the first project week because it was specifically required. Rather than set up a detailed plan for the entire game, the group broke the game up into as many small parts as possible (e.g. design for element X, graphic image of element X, code to implement element X etc.) and put these in the product backlog, from which the tasks were then selected at the beginning of each new iteration. We then kept track

of these tasks with Jira.
[appendix 3]

2.3.4 Risk analysis

We wrote a risk report towards the end of our first iteration to help us to manage uncertainty. We wrote about what could go wrong with our project, both in terms of technical issues and other aspects such as what would happen if one of us got sick, and described how we planned on managing in case something happened. The risk report is very useful to have in the event that any of the risks listed there should actually happen. Whether they are big, small, major or less important, issues are bound to arise during a project of this size and having this risk report saves us both time and effort since we would essentially be prepared for any complication and be more aware of what precautions should be taken to minimize the damage and solve the problem.
Since the group was not paid for the job, all the risks mentioned were focused on completing the tasks within the given time limit, while none of them were concerned with monetary issues or license disagreements, both of which one would normally include when doing a risk analysis of a game development project.
[appendix 2]

2.3.5 Group contract

We made sure to write and agree a group contract during the first iteration. Here all our obligations and responsibilities within the group are listed, in addition to a set of general rules such as when to meet up in the mornings and what to do if a member doesn't do their job, and by signing each member agrees to follow the rules and guidelines specified there. When a team is put together for a specific project such as this, agreeing on a contract for the group should be first priority. That way everybody knows who is responsible for doing what, who reports to who and what rules they are to follow during the project period.
[appendix 1]

3.0 Development and analysis

3.1 Specifications regarding the project

FunCom was very clear from the beginning what sort of job they wanted the group to do, but the specifications of the exact tasks we were going to do were vague. As a very fresh and small branch of the company, the Social Games Division had several ideas of games that they would like to make one day and it was necessary to include in the pre-production phase a meeting with the sole focus of deciding which one of these would be our first task.

3.1.1 Platform

The game was created to be an online web-browser game, and is not planned for release on any other platform than Facebook.

3.1.2 Reusable code

In accordance with FunCom's wishes the code created for the game has been made so that it can be easily reused, either partially or fully, in other similar or semi-similar project later.

3.1.3 Scope

The game is intended for only one platform and since no further content has been planned, it was never intended to keep players entertained for very long. Playing time is estimated to 5 – 10 minutes pr session, taking into consideration the fact that players will, likely as not, be simultaneously working on something else as well, as this tends to be the norm where Facebook in general is concerned.

3.1.4 Social game design principles

Social games or games designed for social apps platforms, such as ours, need to be designed in a quite different way than other types of games, even other browser-based ones. For this reason the social games division at FunCom uses a set of design rules meant specifically for social apps games

For this project, the group actively attempted to follow these rules to the best of our ability, but some of the points listed were found not to apply very much. For instance, patching was never an issue for us, and while we did intend for people to enjoy playing the game was never designed to keep them hooked for a very long span of time. What was the most important to us and our game was keeping it simple, easy to spread, instantly accessible and to use analytics as much as possible in regard to who might play and what sort of player audience our game might attract.

3.1.5 Theme

One challenge the group was aware of from the beginning was choosing a suitable theme for the game. The social games division originally intended to make use of FunCom's rights to the Conan franchise, but there were uncertainties around whether or not this would be allowed and, even if it were, to what degree. Conan Inc would have to approve of the game and game presentation itself and, while awaiting feedback from FunCom's Conan division, various other possible themes were discussed.

One of the main goals was to have the theme be somewhat topical and, if possible, related to previous or upcoming works by FunCom.

- **Conan (or ideal choice)**
  - FunCom has already published a game with this franchise and expanding on it with smaller, themed games is something they had planned to do before the group came to work there
  - It is a relatively easy theme to work with several possible options for mini-games (e.g. Strongest Man, shoot with bow

and arrow etc.)

- Would be a great way of making more people aware of the existing MMO and perhaps make them interested in playing it
- Opens up for the possibility of giving the Conan MMO players advantages in the game by playing the Facebook game and spreading it to their friends

▪ **Zombies**

- FunCom will be releasing a new MMO called "The Secret World", which includes zombies and other forms of undead creatures (the commercial campaign has so far been centered around zombies)
- With games such as Left 4 Dead, Plants vs. Zombies and Resident Evil and a steadily growing amount of zombie movies, this has now become a popular and well-known phenomenon which tends to invoke people's enthusiasm rather than revulsion
- Again this is a theme that it would be fairly easy to come up with gameplay ideas for (e.g. escape the zombies, feed the zombies etc.)
- With a humoristic approach it would not be very difficult to create art for this

▪ **Medieval**

- The social games division is currently in production of a Facebook game with a medieval setting, and the group's game could function as either a mini-game or a pre-release of the actual game to make the audience curious and aware of FunCom's planned entry into the social network gaming platform
- There would be no problems connected to maintaining a certain style, while at the same time we would be able to use the same ideas as we concocted for a Conan-themed game

3.2 Challenges

3.2.1 Prioritizing

When using agile methods in game development it has become standard procedure even in the case of smaller games such as the ones the group was hired to create, to break the game up into as small components as possible [5]. In line with FunCom's agile methods each component was given a complexity rating decided by how difficult and time-consuming it appeared to the group that the respective components might be to complete. Based on these ratings, as well as each component's importance to the game mechanics, the group was able to work out the following priority list:

- **Priority 1**
  - Complete the game design document and design upon the final vision for the game

    - Complete core game mechanics
    - Create place-holder graphics for the core game components to enable testing at an early stage
    - Make the game compatible with Facebook's framework
    - Testing (by ourselves)

- **Priority 2**
    - Create and implement the final graphics for this test-version of the game (note that this does not actually mean final graphics, but placeholder graphics of such quality that they will clearly convey what the final version of the game will, or can be made to, look like)
    - Implement a functioning high score list
    - Implement invite-friends functionality
    - Testing (by ourselves and others outside the team)

- **Priority 3**
    - Polishing and improving the details of the game (such as GUI elements position on the screen)
    - Design and implement the content and gameplay features ranked as nice-to-haves
    - Put up a system for further developing the game (e.g. facilitate the addition of new updates in the future)
    - Testing (by ourselves and others outside the team)

3.2.2 Gameplay presentation

After the basic functionality for the gameplay was in place, the challenge became deciding on how the gameplay should be visually presented. Or, in other words, the group had to make a decision on what the game should actually be about.

FunCom's social games division had a clear idea from the beginning what sort of game they wanted us to make, but once we had gotten to the point where the buttons did what they were supposed to do, it became a matter on how to present this to the player. There were several possibilities and many suggestions. Operating on the assumption that our request to use the Conan franchise for this would go through the group tried to think of different ways of adjusting the gameplay presentation to fit this theme

3.2.3 Time and resources

While the task of designing and producing the basic functionality for the game was a relatively simple task, getting the entire game finished turned out to be a lot more time consuming than the group had

anticipated. Even after cuts were made to simplify the final result, the process of producing the game complete with all Facebook functionality and graphics proved to be quite formidable.
In order to meet deadlines, the group's designer deemed it necessary to bring in a 4th member to help out with graphic production. While the graphics were never meant to be anything but place-holders, creating backgrounds, GUI elements and animations took a lot more time than what had been estimated.

3.2.4 Technical challenges

The group encountered several challenges on this project, starting already at the very first task. Neither of the team members had any previous knowledge on Action Script 3, so even though they were eventually able to make things work, the programmers found themselves modifying classes and changing code almost constantly to try and make a better program each time and constantly learning something new about AS3. One of the ideas for the project was to create a flexible enough Engine to be able to program new, similar games really fast, and as the programmers kept learning more about AS3, they also had to change the code in order to accomplish this as well.

The next challenge was to learn how to use IntelliJIdea, which, once again, none of the members on the team had used before. This wasn't a big challenge, however, since the programmers managed to learn it very quickly. Although they were not familiar with all the available shortcuts (this is still the case), they know enough to use it for the purposes required.
Another challenge was the version control system. Only one of the team's programmers was familiar with Perforce, and since the interface of this tool is a little bit intimidating at first sight, it takes some getting used to, and it has caused the group some issues until the very end.

Finally, the last and hardest challenge has been the Facebook connectivity. Facebook, we can conclude with, is definitely not a gaming site. Not to mention that Facebook has been constantly evolving, changing various things along the road, which unfortunately means that the documentation often isn't up to date. FunCom wanted the server to be written on Java, and since Java is not officially supported by Facebook, the programmers had to use all kinds of technologies to accomplish this, mixing FBML, JavaScript and Java.

3.2.5 Game title

One of the most difficult issues was to come up with a name for the game. While not essential to any part of the actual development process, and not a decision that would be the group's to make in the end, the fact remained that the game's working title was: Conan Strongest Man Challenge Game. A game such as this would need something quick and catchy that people can remember and which tells them what kind of game they are clicking on. For a long period of the time the title was simply shortened to CSCG (Conan Strongest man Challenge Game), and the thumbnails were made with this four-letter title. In the end though we decided on the name Pict-a-pult, which was thought up on a creative impulse and which the group felt fitted the game quite well.

3.2.6 Copyrights

Throughout the project period the group had to be careful not to violate any copyrights issues. The primary one being, of course, the Conan franchise itself around which our game would be based and it was important to FunCom and to the Conan Inc (the owners of the Conan brand) that nothing in the game compromised FunCom in terms of whether or not our game violated the Conan franchise in any way. Furthermore it was important to make sure that none of our place-holder graphics or temporary music and SFX could be accessed or viewed by anyone outside of FunCom's servers as it may lead to issues with other companies.

3.3 Feedback

There were a lot of questions and suggestions brought up during the project period. This mainly included things like gameplay fixes, additional features and bugs.
These were some of the main feedbacks and issues brought up:

3.3.1 Gameplay

- The Pict should be able to hit the roof of the tower/wall when doing a great hit.
- The level difficulty increase is too slow, it should have a linear increase
- Five rounds per level is too much, it should be less.
- Do we need lives? Do some more testing.
- Do we need levels?
- The game freezes when you get a completely miss on the power bar. Fix it!
- Bad feedback to the player, make it more obvious to him what's happening.
- Make the accuracy-bar more consistent.
- Make it possible for the accuracy-bars' color boxes to move.
- Should we include a bonus system?
- Challenge on a particular level or for a game session?

3.3.2 Graphics

- The animations should be funnier/less silly.
- The level backgrounds should be more different to each other.
- The background of the start menu should be the first level. It will look better than an own startmenu background image.

- The score should have a more visible font.
- Improve power-bar/accuracy-bar graphics.
- Add level progression-meter
- Change the icons in-game and make it more obvious to the player what they're for.

3.3.3 Sounds/music

- It should be possible to mute/un-mute sounds and music.

4.0 Solution

4.1 Game presentation

Our project was mainly about making a game for facebook, using actionscript as a programming language, and then to link the files together using Flex. The concept of the game was fairly simple, but considering the fact that this was our first game developed for facebook, it was more than good enough.

4.1.1 Gameplay

The game itself is a Conan-themed "hit as far as you can"-game where you have a power-bar and an accuracy-bar to determine the strength of your hit. Basically the players have to use timing to hit at the right spot on the power-bar and the accuracy-bar, then the power of the hit is a calculation based on the values of both the power- and the accuracy-bar. However, if the player manage to miss on the accuracy-bar, he'll get nothing. The difficulty of the game increases by each level, and in theory the game can last forever until the player fails, as there is no limit on how many levels we have got. However, a maxcap was added to prevent the difficulty on the levels from getting to a point where it is impossible for the player to win.

4.1.2 Visual presentation

As for the visual stuff, everything is sprite-graphics, which obviously is in 2D. The "ball" used was a Pict. This was done to make a more Conan-like atmosphere in the game, and thus make it more visually entertaining. In fact, one of the top level priorities in this game was to make it more of a "visually entertaining" game than anything else, as the mechanics itself was pretty simple. Therefore both Conan and the Pict had a lot of different animations based on how well the player did. To make the player experience some kind of progress while playing, we got several different backgrounds which changes as the player advance to a new level. We also originally planned to have a group of spectators in the background to give feedback to the player, but chose in the end to use an alternative solution due to time issues. To further improve the player-experience we also put in sound effects.

4.1.3 Fun factor

To make the whole game more interesting a challenge feature was put in. This makes it possible for one player to challenge another, though it is only possible to challenge someone who is on your "friends"-list on facebook. The challenge is stored in a database and the opponent may accept or deny it. If the challenge is accepted, each player plays a game session, and the player that gets the best score wins the challenge. All wins/losses are stored into the database for challenges, so each player can see afterwards how many times s/he has won against a particular player. Even though the scores of the challenges are not stored for later review, it will still get stored as a high-score if the player gets a new record during the challenge.

We needed every high-score of each player to be stored in the game. To make this possible we decided to use a database. Using the unique facebook-ID each user on facebook has, we could store the high-score in an database, and link it to that ID. In this way we could easily get and set a high-score. If the player got a new high-score it was stored in the database, and easily viewable on the bottom of the screen. As an extra feature the player could see the high-scores of all the friends who were playing the game along with his own. This was to make it more interesting for each and every player, and since we obviously wanted as many users as possible to play the game, we thought this was a good way to make a player want to invite his friends to the game and compete for the best high-score. We put in an invite-functionality in the game which makes it possible to invite your friends, and thus ask them to install the application/game on facebook.

4.2 Feature list

Our project process was divided into iterations. The game started with some basic ideas, but the further we got in the project, the more suggestions of expanding the gameplay came up. The main ideas brought up during the different iteration meetings were:

4.2.1 Core gameplay functionality

- An Age of Conan-themed game where you have to hit "something" as far as possible (vertically).
- The game mechanics should be easy, and visual entertainment should be prioritized.
- The thing you hit should be a pict (a bushman-looking creature from AoC) who is placed on a board.
- A power bar and an accuracy bar to determine how far the pict is supposed to fly.
- Three lives, where you lose a life if you miss on one of the bars.

4.2.2 Start menu

- The game level should be visible in the background.
- Three menu buttons:
    - **Start game:** Start the game.

  - **Challenge:** Challenge a friend.(see extra features)
  - **Invite:** Invite a friend and ask him to install the app/game on facebook.(see extra features)

4.2.3 Visual components

- Conan shall use a hammer to hit the Pict.
- Conan should do different posing-animations after how well he does.(bad/decent/awesome)
- The Pict should stick to a wall when it has reached its top point, then fall down a few seconds later.
- The pict should have funny animations when in the air.
- Level background changes after a certain amount of levels are completed to show the player that he is progressing.
- Highscores available on the bottom of the screen at all times.

4.2.4 Music and SFX components

- Sound effects for Conan:
  - Hitting the board.
  - Got a bad/decent/awesome hit.
  - Losing a life
- Sound effects for pict:
  - Getting hit
  - Screaming
  - Hitting wall/ground.
- General sfx:
  - Hitting board.
  - Menu-click.
  - Getting points.
  - Player performance response.
- Background music that fits the theme.

4.2.5 Level system

- A linear difficulty increase throughout the game, increasing the difficulty each level.
- Endless levels: Play until you lose all your lives!
- An algorithm for level-increase.
- A maxcap to make sure that a level never gets impossible to complete.

4.2.6 Viral/social gameplay elements

- A highscore that is saved to a database. Thus the highscore will never be deleted.
- A challenge-system where you can challenge your friends and compete to get the best score.
- Invite your friends to the game. (ask them if they want to install the application/game on their facebook apps)

5.0 Thoughts on the solution and project

5.1 Assessment of product result

Here we will take the result of the project in to consideration. This will be done in accordance with the requirements that are set on the criteria for project reports. There will be various evaluations on the project result, such as how the solution turned out, how useful it was to the company we were working for and if the files applied in the solution can be re-used for future projects that they might have.

5.1.1 Evaluation of solution

FunCom had just got the idea of following the wave of the modern "browser gaming" age and therefore, for the first time, they wanted to start producing Facebook-games. So that is what they put us to work with. As it was supposed to be their first game ever produced for Facebook, their main priority were to create a simple game with re-useable code/content. And that's what we ended up with; an easy "hit as far as you can"-game with easy and basic Facebook-functionality and fully re-useable code.

Out of the criteria given to the project we believe the project has been solved with a good and satisfactory result. The working process during the project was divided into iterations. Each iteration had a duration of two weeks. At the end of each of them we got feedback on the work we had done that far. The feedback was given by the project managers, which did not work on the same game as we did. In other words, their evaluation of the game was very trustworthy.

The basic parts that were used in the game was 2D graphics with animations, sound effects, music, simple voices for the main character and a database to store values that we wanted to save for later, such as high scores, challenges and so forth. As the game was developed for FunCom, they wanted a theme taken from a game that they already were known for. The theme they wanted was from their online game Age of Conan (AoC), so who else but the one and only Conan from their series would fit in as the main character?

As said before the concept and gameplay was designed to be simple. The game itself was a timing-game. The basics contained a power- and a accuracy bar, where the player had to do the right timing to get the best score. The difficulty was set by a simple algorithm that told the game to increase the difficulty by a small value each time a level was cleared. So the game could basically go on forever, but the difficulty would not rise if the player reached the maximum amount of difficulty required to complete a level. How many points the player would get was determined by a formula that mixed the score of the power and the accuracy bar. As the game characters we had the main character (Conan) and a Pict (bushman-looking creature) which had the role as a ball that was placed on a board.

When the power of the timing-results was determined, Conan hit the board and the Pict went flying vertically.

Due to the simple gameplay, we had to do something to make it more interesting to the players. Therefore the focus was put on making the game visually interesting with funny animations, events and feedback to the player. Based on how well the player did, he got feedback from the main character. For getting the fun factor up we made the animations of the Pict funny and put in some features like making it hit a wall on the level when he reached the top. To give the player a feeling of progressing we made the background change when the player had finished several levels in the game.

We put in functionality for players to challenge friends and invite them to the game to make it more available for everyone and make it more competitive, in a casual way. The idea behind this was that the whole thing would be more interesting if the players could compete against other friends. For saving information for later use we linked the game to a database which stored all such values. This was things like high scores, challenges, challenge statistics (wins/losses) and so forth.

5.1.2 Usefulness to the company

As this game was the first Facebook-game ever made by FunCom, we believe this was a really good way to show the world that they joined the wave of Facebook games, and already got basic Facebook-functionality in it.

As all programmers know; code that is created for one project can easily be modified and re-used in another. In that sense this project has been really useful to the company regardless of if it’s getting successful or not. The code for Facebook-calls, database-connection, multimedia-loading (images/sounds/music etc.) is already made, and may be used for any projects made by FunCom in the future, which will save them a lot of time.

If this project however becomes a success, they might earn money on it in some way, thus giving them a larger profit and good public relations. In addition they won’t have to put a lot of effort in to make people want to try any new Facebook-games in the future!

5.2 Assessment of method and tools

5.2.1 SCRUM

It was never really an option to work with anything other than SCRUM, since that’s what the rest of the social division does. It worked out very well for the group, however, as the Daily SCRUM sessions helped us keep track of the daily progress of our team members, and we found that having a product owner who we could ask about specifics and who could evaluate the product as it came along was very helpful. Gathering all our tasks in a product backlog and selecting tasks for each sprint was also great way of maintaining focus throughout the project period, because we could always check to see how much was left of the entire project at any time.

One of the weaknesses of SCRUM in regard to game development is the lack of clearly defined roles within the team, but since these roles tend to assign themselves automatically it was never really an issue to

us. It can be debated, however, if adding a second designer would have spurred the need for a more official delegation, but as it were this was not necessary.
In addition the knowledge that we needed to have an actual executable delivery for each iteration worked as a motivational factor when we worked.

5.2.2 Choice of development software

5.2.2.1 Programming tools

Having a strong foundation on programming languages and theory of computation, learning a new language is usually very easy, at least the core mechanics of the language. The programmers did not spend a lot of time learning to code with as3, but they did spent a lot of time researching about Game Engines, specifically done with AS, and based on what was required for this and future games at Funcom. Although, since Flex and AS3 is relatively new, there are not a lot of tutorials about it, and most of the ones we found, used Adobe Flash, which we didn't have a license for(until sometime later but it wasn't worth it learning since we already had information on Flex).

In the end, we have a very small engine, with just a few classes, but we believe they provide the basic functionality, you can have your game menu and challenge screen menu in a couple of hours, so you only need to dedicate to the gameplay itself, and we have classes for Sprites, Texts, Pop-ups and other elements. So we are very confident that a very simple game can be done in a couple of days, assuming we have the design and art ready for it.

5.2.2.2 Design tools

All the design was written either directly into the wiki or in open office documents, neither of which were particularly effective to work with. Structuring the documents was troublesome because of syntax and automatic formatting, and for someone who is used to working with Microsoft Office, they both seemed very outdated and slow

5.2.2.3 Graphical design tools

While Photoshop was great for creating backgrounds and GUI elements, making animations was not optimal in this program due to the resolution decreasing a little for each copy-paste. Fortunately the opposite was the case with Inkscape, and so with a combination of these two both static and moving images could be made in a satisfactory way. Photoshop is commonly regarded as an industry standard, but was never intended for graphic production as much as picture and photo editing. And while the vector-based Inkscape fulfills its purpose, further reading has lead us to think Adobe Illustrator should be considered for graphic production, as it provides both the level of detail and options as Photoshop and the possibility of making vector based graphical components like Inkscape

5.2.3 Communication tools

5.2.3.1 Skype

Enabling both easy two-way communication between two parties as well as the exchange of files, Skype worked very well. Also, while having less of the flashy social settings than MSN, it also has significantly less spam and viruses. Skype was used to send graphical images and audio files, sharing links and quick messages

5.2.3.2 Mozilla Thunderbird

This e-mail reader was used for sending longer messages, normally mails that went out to everyone and each new task that was selected for each iteration were sent out as separate mails

5.2.3.3 Dropbox

This tool was very useful to the group because of the instant synchronizing functionality, and was used for sharing documents related to the NITH report delivery

5.2.3.4 MediaWiki

This tool was used only by the designer, who feels that while a wiki page does provide easy site navigation, it has a somewhat lacking syntax and for a game as small as ours it would perhaps have been sufficient with a document in a shared folder, which would have been quicker and easier to write too. The advantages are obvious, but at the same time it felt troublesome to organize the text in the wiki

5.2.4 Techniques used

5.2.4.1 Daily SCRUM

This turned out, interestingly enough, to have a very different effect than anticipated. While the concept of daily SCRUM is simple enough, it forces you to keep track of everything you do in the project as well as helping you keep your focus on the next delivery, and we experienced several times that we didn’t remember what exactly it was that we had been working on or what we were supposed to be finished with. This in turn made us realize that we needed to start focusing again, like a daily wake-up call, and work more like humans and less like zombies

5.2.4.2 Jira and biweekly iterations

Dividing the project into biweekly iterations proved to be a good call, most of all because it saved time. Planning and reviewing an iteration is time-consuming, and since one of the ground principles of agile methods is to have a new deliverable project at the end of each iteration, it is doubtful how much there would be to show for after only 3 days worth of work, had we gone for the weekly iterations. Jira was an excellent tool for the designer, who found it very satisfying to be able to keep track of tasks this way, but the programmers found it somewhat troublesome to have to update their own tasks, especially because coding often depends on other pieces of code to work thus making it difficult to determine when a coding task can actually be considered done

5.2.4.3 Project plan

This particular method seemed somewhat forced in our case and aside from the Product Backlog and the individual set of tasks for each of the iterations, we did not use or miss it. Clearly a more defined set of deadlines with specific dates would have been necessary for a larger project, but when the scope is as small as this, the product backlog and simple verbal communication proved to be sufficient

5.2.4.4 Risk analysis

As with the project plan, the risk analysis was something the group did because it was required and not because we felt we needed it, and we never used it after it was written up. This may have to do with the nature of the project, where no money and no particularly new and special software are involved and there is almost nothing at stake. That being said, however, making such a list together as a team at the beginning of a project can be a useful exercise because it makes everyone aware of the risks involved and can work as a motivational factor even if the list is never looked at again

5.2.4.5 Group contract

The group has worked very well together both with each other and in relation to FunCom and the social games division, so the group contract fortunately was not brought up again after it had been written, but in contrast to the Project plan and Risk analysis, the necessity of having a group contract is indisputable. In the case of our group, Hanne and Robin had worked together before, but never alone together on a separate project like this, and since Edgar did not even attend NITH, there were several uncertainties around how well we would work together and how well each of us worked on his/her tasks. Therefore it was important to us to establish a set of ground rules, including how to distribute the workload, the areas of responsibilities, as well as the hierarchy within the group and extended to the product owners and project supervisors.

5.3 Assessment of product in comparison to scientific research

While the game has maintained the simplicity and instant-fun factor that define social games, the nature of the core-gameplay as well as the specific requirements stated by the employer prevented the group from following up on certain aspects. These aspects include planning and facilitating release of further content and patching, which were never a part of the equation from the beginning, and adding game elements that make the game addictive over a longer period of time or to make them return every day.
Basing our product-analysis on the 10 social apps commandments [appendix 5], the game has succeeded in being Accessible, Simple and give players a feeling of Accomplishment. Being a challenge-type game there are no viral elements similar to those one can find in Zynga games, where wall-posts often offer free gifts to people who click on them or give advantages to players with many playing friends. However, our game does have a high-score list as well as a Challenge-mode and an invite function, and it is possible to post your score to the wall at the end of every game session, and the group considers these to be valid viral elements in a game such as this. The idea is that they will encourage players to compete with their friends and try to beat each other and get as high as possible up on the high score list.
The reason for this is that the group created only one game with a competitive challenging type of gameplay.

6.0 Conclusion

6.1 The group's conclusion of the project

In conclusion the group has asserted that even when making a game that is essentially meant to be a prototype and as small and simple as possible, things rarely go as planned and our final product turned out to be almost nothing like what we originally intended it to be.

Technically the group feels satisfied with the fact that the game is as stable as a Facebook application can possibly be. As previously stated, Facebook is difficult to work with and thus we consider this to be quite a feat. The engine that was created is easy to use and provides communication with Facebook and keeps the player's high score, as well the usual graphics and text.
The Facebook server has undergone changes throughout the process and we now have a very basic module that should, in theory, work well for further developments and that is robust enough to resist most of the new Facebook changes. A possible solution for further development would be to develop a scripting system that would make the module as independent from Facebook as possible, but there was no time for that within the scope of the project.

Gameplay-wise the group was pleased that the game turned out as well as it did in the end. It went through several phases of testing, re-assessing and re-designing and the group spent a lot of time working on how to make it fun. We believe now that it is, and have even discovered that the gameplay had a type of short-term addictiveness, which we feared we would not be able to accomplish, and during the testing phases there were even players competing actively to beat each other. It is unlikely that anyone will experience any long time addictiveness to it, but that was, in any case, never part of the intended scope.

6.2 The group's conclusion of the project period

Working at FunCom was an invaluable experience, and it was truly a pleasure to be able to work with so many talented people. The group's individual members have each benefited from the good advice and professionalism of our temporary colleagues, and the project period has been blissfully free of disputes and major misunderstandings.

While the designer and programmers may have expanded their respective fields of knowledge separately, the planning and execution of each new iteration as well as using SCRUM on a daily basis has taught us a lot about how to work together and coordinate our work as a team during a game development process.

6.3 What the employers thought

FunCom expressed that they were satisfied with the group both as employees and with the product itself, but that some things could have been better. FunCom is not a company that will release anything that does not *look* professional, so regardless of whether it eventually is scheduled for release or not, it would have to undergo a serious face-lift before it can be properly evaluated.

The focus of the project has been to create a game that runs and plays as smoothly as possible, for the Facebook tie-ins to work and for the gameplay experience to be fun. The team has achieved this and the current status of the game is that graphical resources have been allocated to take the game to the level of a finished product.

No decision has been made yet as to what will happen to the game after this, but basically there are two outcomes; one being that nothing ever happens to it, or that it is put on hold indefinitely until it's decided whether time and resources should be spent on it or not. The other is that a professional art team is put to work on improving it and that it gets released on Facebook in the end.

7.0 Reference list

1. “Casual Games Discussion” (2007, by: Jussi Kuittinen, Annakaisa Kultima, Johannes Niemelä and Janne Paavilainen)

2. Facebook wiki (internal mediaWiki article at FunCom, 2010)

3. “Facebook” (Wikipedia article, 2010, http://en.wikipedia.org/wiki/Facebook)

4. “SCRUM” (Wikipedia article, 2010, http://no.wikipedia.org/wiki/Scrum)

5. “Paper Burns: Game Design with Agile Methodologies” (2006, by Rory McGuire, http://www.gamasutra.com/view/feature/2742/paper_burns_game_design_with_.php)

6. “Effect of Success versus Other Players on the Perception of Fun” (2009, by Charles Butler, M.I.T., the Guildhall at SMU)

8.0 Appendixes

8.1 Group contract

Here is the full Group Contract that was written up and signed by the group's members at the beginning of the project period.

Group Contract PJ600

The object of the group contract is to maintain the motivation for each of the members and to ensure that a certain set of ground rules concerning the project and the work process are being followed. This contract also describes further the group's expectations regarding the amount of work we will be doing as well as the respective roles within the team.

Responsibility

- **Roles**

  Product owner: Jarle Snertingdalen of FunCom's Social Games division
  Scrum Master: Yngvild Lothe of FunCom's Social Games division
  Lead designer and graphical artist: Hanne Lien
  Programmer: Edgar Omar Cebolledo
  Programmer: Robin Berg Pettersen
  NITH project supervisor: Faltin Karlsen

- **Tasks**

  Each member's tasks and areas of responsibility limit themselves to our respective fields of expertise. The group will together work out a complete project backlog divided into smaller tasks for each sprint, in accordance with SCRUM and agile methods. Each task will be clearly marked and assigned to a team member, who, once all the tasks for that sprint have been completed, can then choose a new task from the project backlog and ask the Scrum Master to assign this task to them. Hanne Lien is in charge of the project-related documentation and of making sure that the group makes all the deadlines set by NITH and that all parts of the respective deliveries are included at the end of each project week.

Attendance

- **Project period**

  The project PJ600 lasts from January 12th until May 19th, and all through that period the group's members are obliged to follow the rules of attendance stated below.

- **Work days**

  In accordance with the weekly schedule of NITH the days Tuesday, Wednesday and Thursday are to be spent working at FunCom's main office in Skøyen. If a group member wishes to spend more time working there or the team's external schedule allows it, they are free to come in as often as they wish.

- **Working hours**
  The work day begins at 9 AM in the morning and lasts until 5 PM in the afternoon. If a group member wishes to come in earlier or leave later they are free to do so. Daily Scrum will be held at 09:15 and all members are obliged to attend this daily session.

Absence

If a group member is or knows that they are going to be absent they must report this to the Scrum master. Valid reasons for absence are the same as general standards within any line of work (e.g. illness, death etc.), but includes also in our case urgent, school-related matters (e.g. classes, tests etc.).

Communication

- **Contact information**
  Each member of the group must inform the others of how they can be contacted and phone numbers and e-mail addresses must be easily accessible by the other members of the group.

- **Chatting**
  In accordance with the standard at the Social Games division at FunCom, each of the group's members is required to create a Skype account and be available there during work hours as far as this is possible. Considering that the group's members are within close vicinity of each other in the offices provided by FunCom immediate matters or questions may also be solved simply by talking to each other so long as co-workers are not disturbed.

- **Problems and discussions**
  Group members agree to respect and listen to the opinions of the others and to voice their own in a respectful way. When a problem arises it should be discussed and resolved internally. If the problem is of such a nature that it cannot easily be solved within the group, a meeting will be arranged with the Scrum Master and the matter be put up for discussion there.
  Should any trouble connected to the group's working conditions or general treatment from the employers arise these matters will be reported to the group's internal project supervisor at NITH, Faltin Karlsen, and discussed at a meeting.

Confidentiality

- **In regard to FunCom**
  All the group's members are required to sign FunCom's obligatory confidentiality contract and act according to the guidelines stated there.

- **In regard to other groups**
  In accordance with the contract mentioned above none of the group's members are permitted to reveal details about the group's project to others, including the

concept and specific work tasks. The only exception to this rule is Faltin Karlsen, who will need to know enough about the project that he can give the group actual, constructive feedback and advice.

Work flow

- **SCRUM**

  The group will use SCRUM and agile methods for our work, with a SCRUM Master and daily SCRUM sessions. This is the standard working method at FunCom. Due to the nature of the project as a game development project, certain alterations have been made to the standard SCRUM method, such as how all members will have tasks that only they and none of the others will be able to do and how our Scrum Master is also in this case partly the product owner.

- **Jira**

  The sprint backlog will be divided into as small tasks as possible, then their complexity will be given a rating and be assigned to a group member and all of this will then be available on Jira, which is a tool used by FunCom, and all group members are required to update their tasks on this tool, when their respective tasks are *in-progress* and *completed*.

- **Media Wiki**

  The design document, including the pitch and the specifics for each game element, is to be available for all the group's members on the social games media wiki provided by FunCom. The designer is obliged to fill in all details here and update as soon as changes to the design occur and the programmers are required to keep themselves up to date on the wiki before they send questions to the designer or make independent decisions regarding the game structure.

- **Distractions**

  Each group member is responsible for setting their own daily work schedule and completing the tasks given to them within the set time frame. Non-work related distractions are discouraged, but is not problematic the group unless it disrupts the work flow of the others, affects the quality of the work and/or delays the completion of the sprint.

Quality assurance

- **Goal**

  The project is to be completed with focus on completing a product that is satisfactory to the employers and that meets with their expectations. It is the employers, FunCom, who decide whether the quality of the result is acceptable. The documentation that is to be delivered to NITH should be thorough and complete in all regards and meet with all requirements in a satisfactory way. It is the group's project supervisor at NITH who will determine the quality of this work and all the group's members should be in consensus before the final delivery is made on May 19th.

- **Quality**

The group aims for the highest grade, 'A', and thus agree to spend all the time and effort it requires to attain this and the work should be satisfactory in accordance with the guidelines given by NITH on what qualifies as a top-grade project deliverance. While the work done will and should be based on research and guidelines set by the employers, but it will all be original work done by the group during the project period and none of it should be copy-pasted or stolen directly from someone else.

The group contract

In the event that the group contract needs to be revised the following procedure goes:

The potential changes should be discussed in plenum and voted over.

Signatures
Place ________________________ Date ___________________
________________________ ___________________________ ___________________________
________________________ ___________________________ ___________________________

8.2 Risk List

Here is the Risk List that was written by the group at the beginning of the project period.

Risk plan for group 3, PJ600

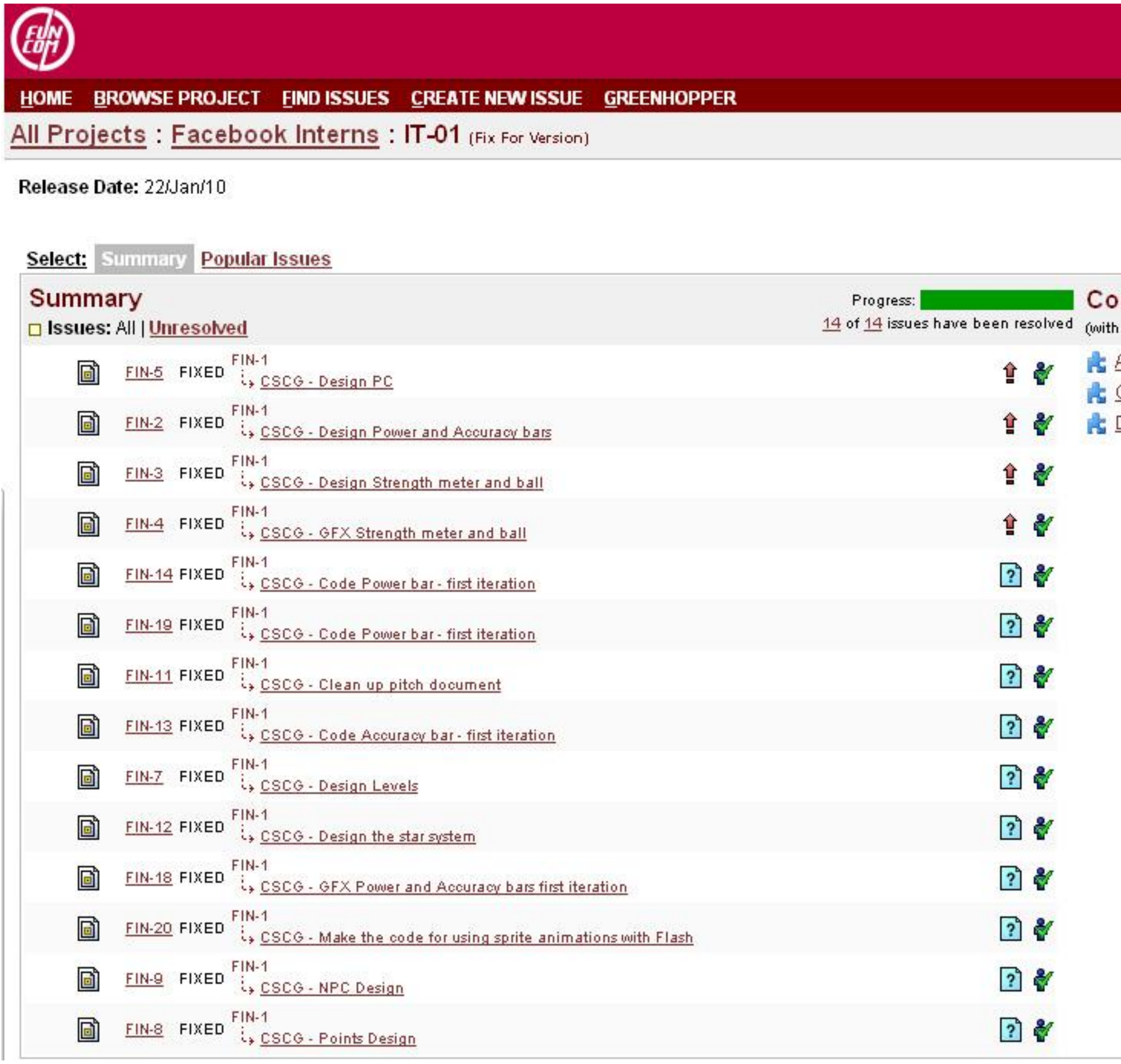
Risk	Consequence	Likeliness	Risk points	To prevent	Effort if problem occurs
Missing knowledge	4,00	0,90	3,60	Research before starting to work	Research, ask fellow programmer
Wrong estimate of time	5,00	0,80	4,00	Know your limits, good planning	Re-estimate time
Sickness	5,00	0,20	1,00	Proper clothing, think about what you eat.	Inform team, relax, medicine
Lack of motivation	9,00	0,10	0,90	Motivation seminars, equal distribution of work	Take a break
Files disappear	10,00	0,03	0,30	Backup	Start over, tell the boss
Unfamiliar software	2,00	1,00	2,00	Use the software you're familiar with or LEARN new software	Ask someone who knows the software to teach you.
Network Problems	5,00	0,05	0,25	Do not modify network settings	Call internet company and ask for help
Lack of computers	10,00	0,01	0,10	Do not spill liquid on the computers/break them	Get new computers
Accident/Injury	8,00	0,30	2,40	Be careful	Call hospital, get medication
Losing job	10,00	0,01	0,05	Be kind, do your work	Beg for mercy
Train delay	1,00	0,90	0,90	Go earlier to work	Tell the boss, end work later than usual.

Group members:

Hanne Lien
Robin Berg Pettersen
Edgar Omar Cebolledo Gutiérrez

8.3 Project Plan

The following screens are taken from Jira and show our work progress and different tasks for the different iterations.



All Projects : Facebook Interns : IT-01 (Fix For Version)

Release Date: 22/Jan/10

Select: Summary Popular Issues

Summary

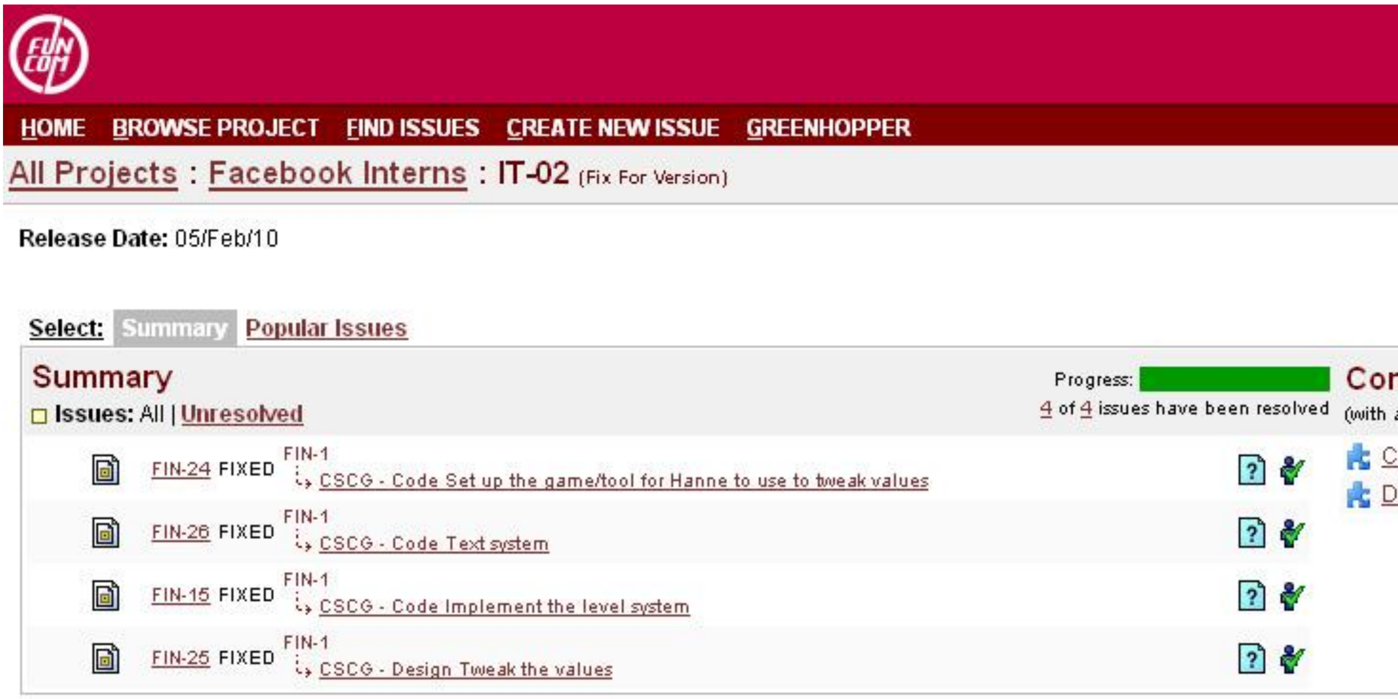
□ **Issues:** All | Unresolved

Progress: 🟩
14 of 14 issues have been resolved

Co
(with

FIN-5	FIXED	FIN-1 ↳ CSCG - Design PC	↑
FIN-2	FIXED	FIN-1 ↳ CSCG - Design Power and Accuracy bars	↑
FIN-3	FIXED	FIN-1 ↳ CSCG - Design Strength meter and ball	↑
FIN-4	FIXED	FIN-1 ↳ CSCG - GFX Strength meter and ball	↑
FIN-14	FIXED	FIN-1 ↳ CSCG - Code Power bar - first iteration	?
FIN-19	FIXED	FIN-1 ↳ CSCG - Code Power bar - first iteration	?
FIN-11	FIXED	FIN-1 ↳ CSCG - Clean up pitch document	?
FIN-13	FIXED	FIN-1 ↳ CSCG - Code Accuracy bar - first iteration	?
FIN-7	FIXED	FIN-1 ↳ CSCG - Design Levels	?
FIN-12	FIXED	FIN-1 ↳ CSCG - Design the star system	?
FIN-18	FIXED	FIN-1 ↳ CSCG - GFX Power and Accuracy bars first iteration	?
FIN-20	FIXED	FIN-1 ↳ CSCG - Make the code for using sprite animations with Flash	?
FIN-9	FIXED	FIN-1 ↳ CSCG - NPC Design	?
FIN-8	FIXED	FIN-1 ↳ CSCG - Points Design	?

Iteration 01



HOME BROWSE PROJECT FIND ISSUES CREATE NEW ISSUE GREENHOPPER

All Projects : Facebook Interns : IT-02 (Fix For Version)

Release Date: 05/Feb/10

Select: Summary Popular Issues

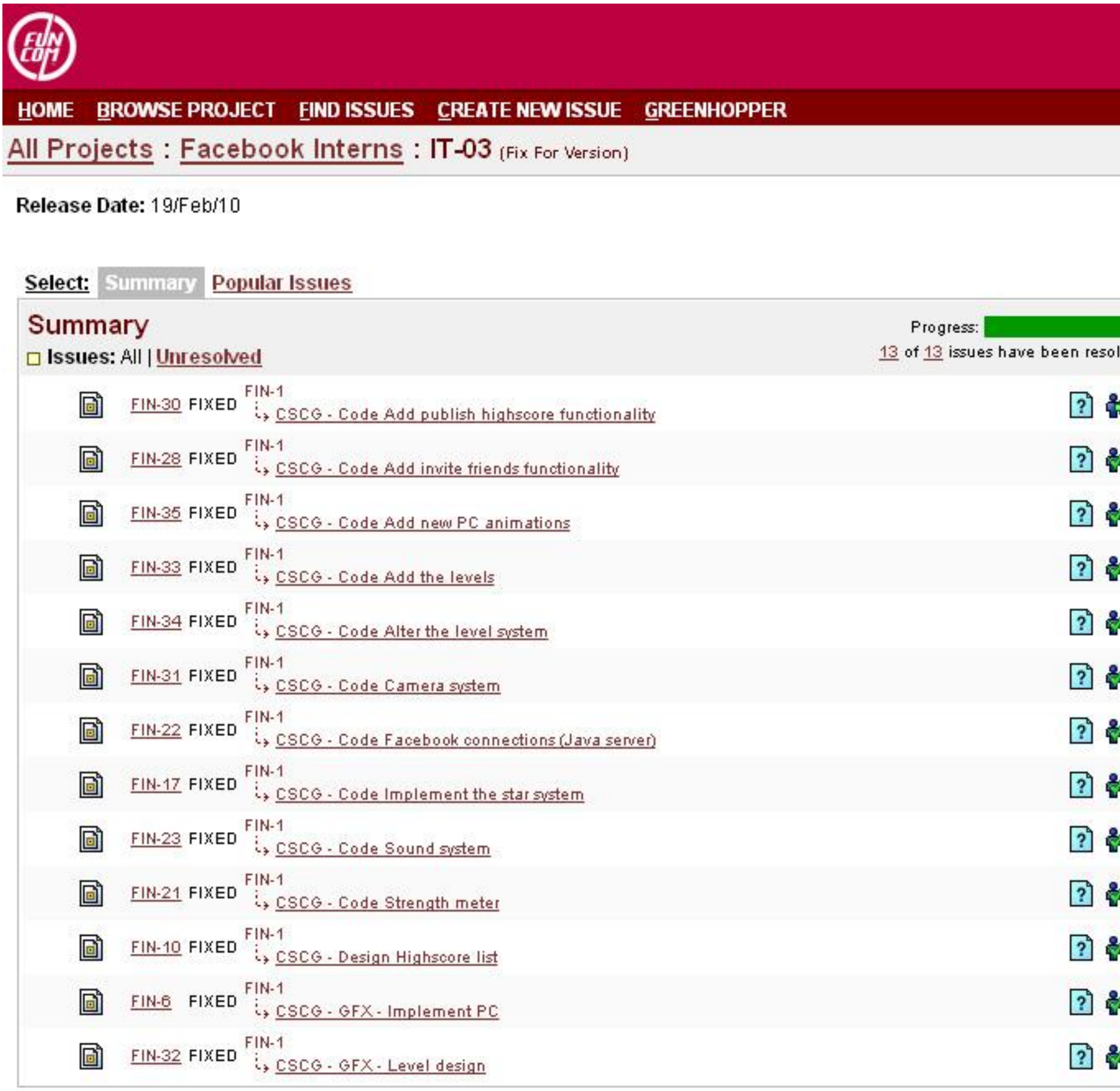
Summary

□ Issues: All | Unresolved

Progress: 4 of 4 issues have been resolved

FIN-24	FIXED	FIN-1 ↳ CSCG - Code Set up the game/tool for Hanne to use to tweak values
FIN-26	FIXED	FIN-1 ↳ CSCG - Code Text system
FIN-15	FIXED	FIN-1 ↳ CSCG - Code Implement the level system
FIN-25	FIXED	FIN-1 ↳ CSCG - Design Tweak the values

Iteration 02



All Projects : Facebook Interns : IT-03 (Fix For Version)

Release Date: 19/Feb/10

Select: Summary Popular Issues

Summary

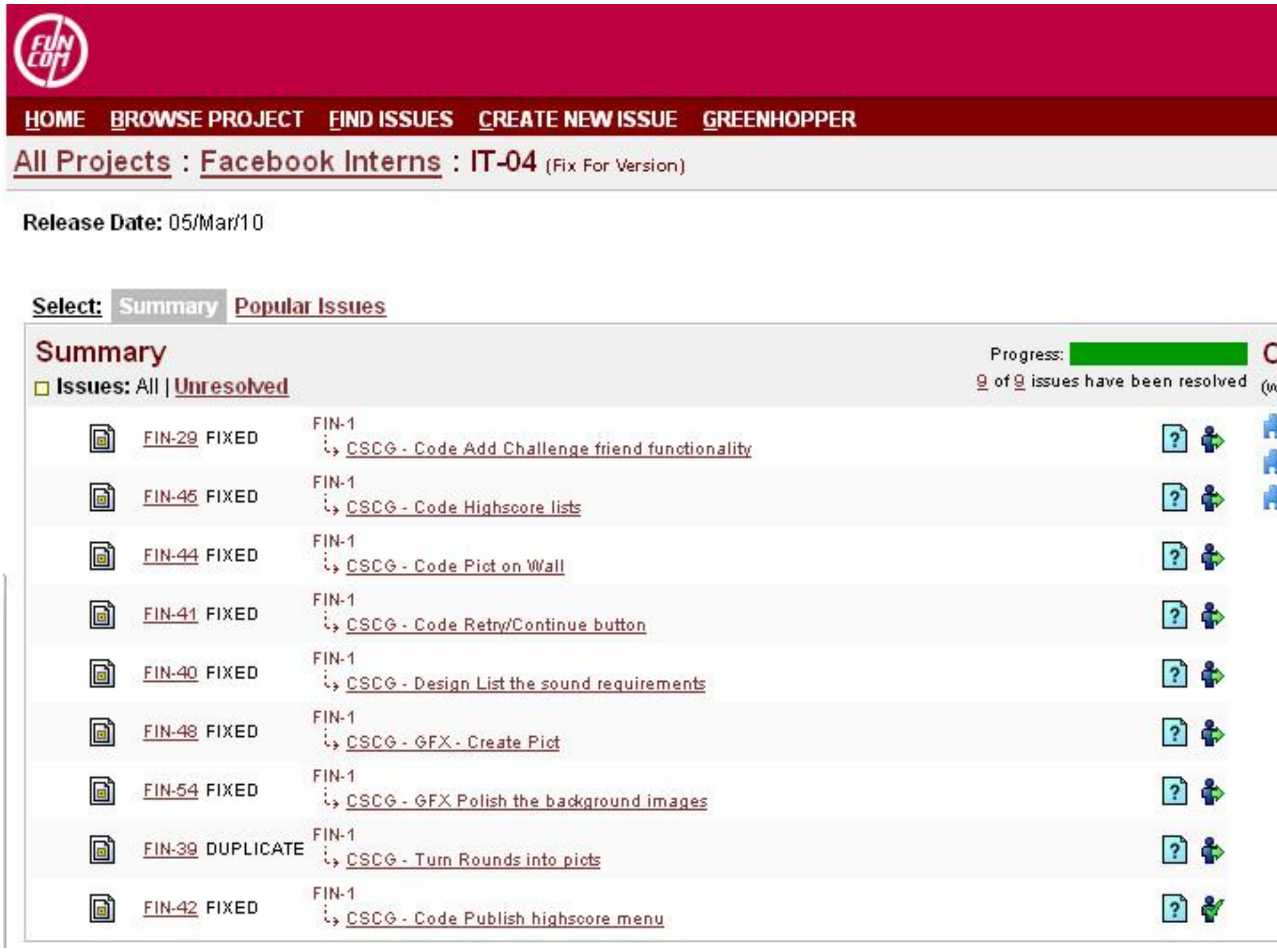
□ Issues: All | Unresolved

Progress:

13 of 13 issues have been resol

FIN-30	FIXED	FIN-1 ↳ CSCG - Code Add publish highscore functionality
FIN-28	FIXED	FIN-1 ↳ CSCG - Code Add invite friends functionality
FIN-35	FIXED	FIN-1 ↳ CSCG - Code Add new PC animations
FIN-33	FIXED	FIN-1 ↳ CSCG - Code Add the levels
FIN-34	FIXED	FIN-1 ↳ CSCG - Code Alter the level system
FIN-31	FIXED	FIN-1 ↳ CSCG - Code Camera system
FIN-22	FIXED	FIN-1 ↳ CSCG - Code Facebook connections (Java server)
FIN-17	FIXED	FIN-1 ↳ CSCG - Code Implement the star system
FIN-23	FIXED	FIN-1 ↳ CSCG - Code Sound system
FIN-21	FIXED	FIN-1 ↳ CSCG - Code Strength meter
FIN-10	FIXED	FIN-1 ↳ CSCG - Design Highscore list
FIN-6	FIXED	FIN-1 ↳ CSCG - GFX - Implement PC
FIN-32	FIXED	FIN-1 ↳ CSCG - GFX - Level design

Iteration 03



All Projects : Facebook Interns : IT-04 (Fix For Version)

Release Date: 05/Mar/10

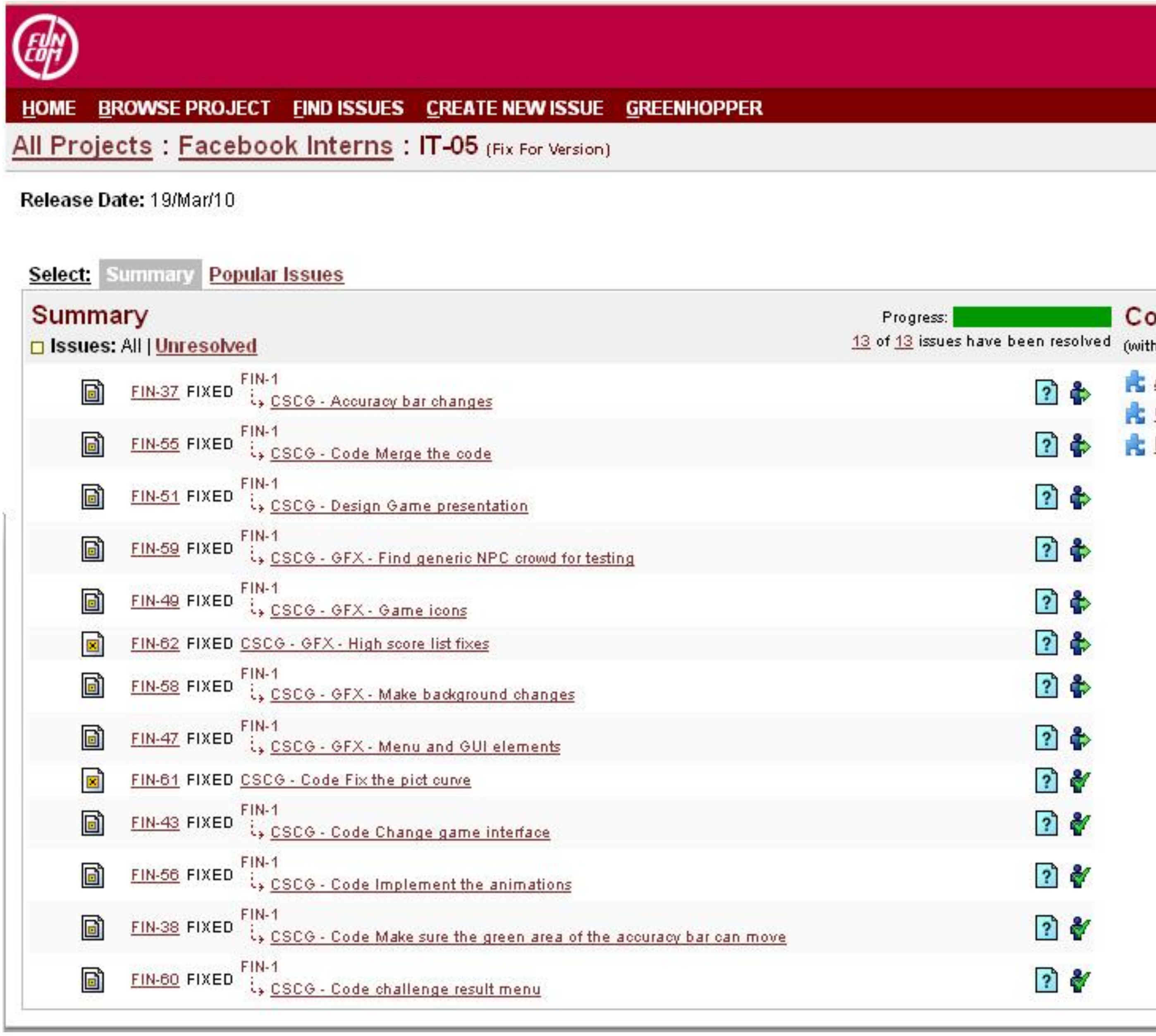
Summary

□ **Issues:** All | Unresolved

Progress: 🟩
9 of 9 issues have been resolved

FIN-29	FIXED	FIN-1 ↳ CSCG - Code Add Challenge friend functionality	?
FIN-45	FIXED	FIN-1 ↳ CSCG - Code Highscore lists	?
FIN-44	FIXED	FIN-1 ↳ CSCG - Code Pict on Wall	?
FIN-41	FIXED	FIN-1 ↳ CSCG - Code Retry/Continue button	?
FIN-40	FIXED	FIN-1 ↳ CSCG - Design List the sound requirements	?
FIN-48	FIXED	FIN-1 ↳ CSCG - GFX - Create Pict	?
FIN-54	FIXED	FIN-1 ↳ CSCG - GFX Polish the background images	?
FIN-39	DUPLICATE	FIN-1 ↳ CSCG - Turn Rounds into picts	?
FIN-42	FIXED	FIN-1 ↳ CSCG - Code Publish highscore menu	?

Iteration 04



All Projects : Facebook Interns : IT-05 (Fix For Version)

Release Date: 19/Mar/10

Select: Summary Popular Issues

Summary

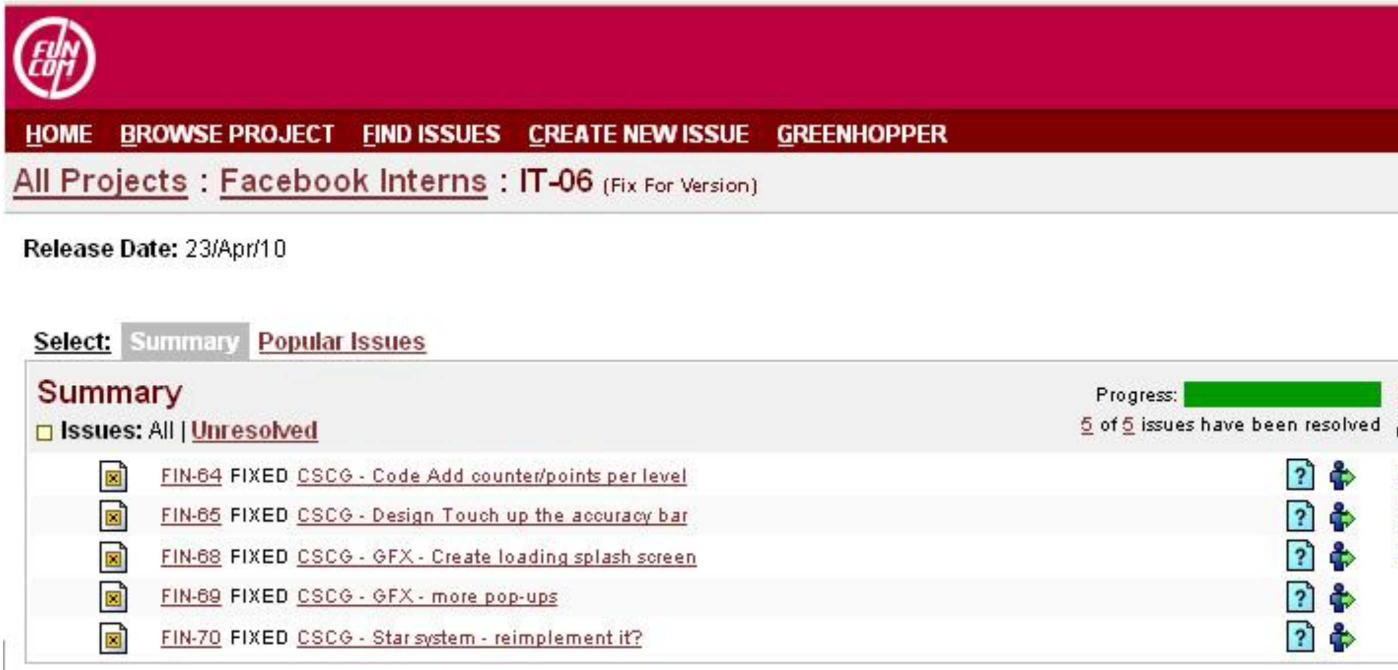
□ Issues: All | Unresolved

Progress: 🟩

13 of 13 issues have been resolved

FIN-37	FIXED	FIN-1	↳ CSCG - Accuracy bar changes
FIN-55	FIXED	FIN-1	↳ CSCG - Code Merge the code
FIN-51	FIXED	FIN-1	↳ CSCG - Design Game presentation
FIN-59	FIXED	FIN-1	↳ CSCG - GFX - Find generic NPC crowd for testing
FIN-49	FIXED	FIN-1	↳ CSCG - GFX - Game icons
FIN-62	FIXED		CSCG - GFX - High score list fixes
FIN-58	FIXED	FIN-1	↳ CSCG - GFX - Make background changes
FIN-47	FIXED	FIN-1	↳ CSCG - GFX - Menu and GUI elements
FIN-61	FIXED		CSCG - Code Fix the pict curve
FIN-43	FIXED	FIN-1	↳ CSCG - Code Change game interface
FIN-56	FIXED	FIN-1	↳ CSCG - Code Implement the animations
FIN-38	FIXED	FIN-1	↳ CSCG - Code Make sure the green area of the accuracy bar can move
FIN-60	FIXED	FIN-1	↳ CSCG - Code challenge result menu

Iteration 05



All Projects : Facebook Interns : IT-06 (Fix For Version)

Release Date: 23/Apr/10

Select: Summary Popular Issues

Summary

□ Issues: All | Unresolved

Progress: 🟩

5 of 5 issues have been resolved

FIN-64 FIXED CSCG - Code Add counter/points per level

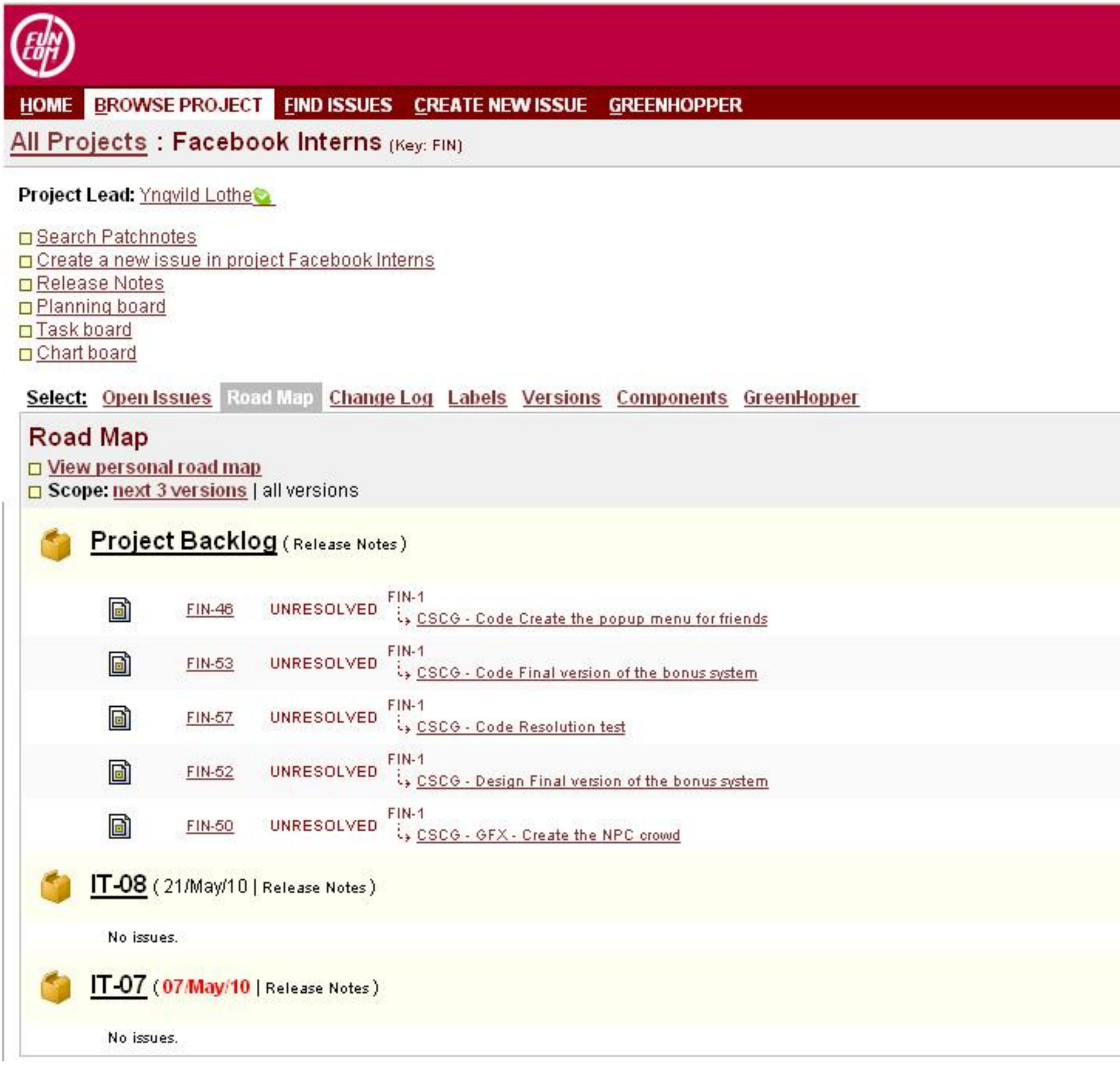
FIN-65 FIXED CSCG - Design Touch up the accuracy bar

FIN-68 FIXED CSCG - GFX - Create loading splash screen

FIN-69 FIXED CSCG - GFX - more pop-ups

FIN-70 FIXED CSCG - Star system - reimplement it?

Iteration 06



HOME BROWSE PROJECT FIND ISSUES CREATE NEW ISSUE GREENHOPPER

All Projects : Facebook Interns (Key: FIN)

Project Lead: Yngvild Lothe

- □ Search Patchnotes
- □ Create a new issue in project Facebook Interns
- □ Release Notes
- □ Planning board
- □ Task board
- □ Chart board

Select: Open Issues Road Map Change Log Labels Versions Components GreenHopper

Road Map

- □ View personal road map
- □ Scope: next 3 versions | all versions

Project Backlog (Release Notes)

FIN-46	UNRESOLVED	FIN-1 ↳ CSCG - Code Create the popup menu for friends
FIN-53	UNRESOLVED	FIN-1 ↳ CSCG - Code Final version of the bonus system
FIN-57	UNRESOLVED	FIN-1 ↳ CSCG - Code Resolution test
FIN-52	UNRESOLVED	FIN-1 ↳ CSCG - Design Final version of the bonus system
FIN-50	UNRESOLVED	FIN-1 ↳ CSCG - GFX - Create the NPC crowd

IT-08 (21/May/10 | Release Notes)

No issues.

IT-07 (07/May/10 | Release Notes)

No issues.

Iteration 7, 8 and the Product Backlog with the tasks that were eventually cut

8.4 Facebook

Hereunder Facebook is presented in more detail; both as a social networking platform and as a gaming platform.

Facebook

- **Facebook**
  Facebook, now the greatest social networking site in the world (overtaking its main

competitor, MySpace, in April 2008), was initially created by Mark Zuckerburg in 2003 as something called 'Facemash', a Harvard equivalent of a Hot or Not website. From then on the concept developed into a digital version of a printed book that was printed each year at Zuckerberg's prep school, Phillips Exeter Academy, which had pictures of all the staff and students and was distributed to them as a way of making it easier to get to know one another.

Facebook as we know it today was founded by Zuckerberg and his fellow computer science students at Harvard, Eduardo Saverin, Dustin Moskovitz and Chris Hughes. Since September 2006 it is owned and operated by Facebook Inc.

While originally intended only for Harvard students, and limited to this by the founders, the network gradually expanded to include other Ivy League colleges, then all colleges in North America and Canada. Since then it has continued to expand and is currently available worldwide to everyone and anyone who is over the age of 13 and can provide a valid e-mail address.

It maintains, however, its philosophy of being a social networking site where people can connect with their friends and relatives, post updates about themselves, join and participate in groups and now, as of recently, they can also play a number of social and/or casual games.

- **Facebook as a gaming platform**
  Games for social networking sites such as Facebook and MySpace is a fresh venue, consisting of simple, browser-based games intended for a marked of casual players. The Facebook Platform was launched in spring 2007, and provided software developers with a framework for creating applications that could interact with core Facebook features. These include social games, where the player may invite their friends to play and share their own results through wall posts, gifts, free classified ad posting, events and video sharing.

There are obvious advantages to developing for platforms such as this due to the potentially enormous mass of available players. Currently, it is the San Francisco based game development company Zynga that appears to be the company who has been able to make the most of making games for social networking sites. With big titles such as FarmVille, FishVille and YoVille on their record, the latter of which was estimated to have over 60 million active users in December 2009, Zynga has, in spite of having been criticized on several points regarding their business model, rapidly grown to become a name recognized by everyone. They practice a development process with consequent testing and updating, often releasing 10 or more versions of one product to see which ones get the most desirable results.

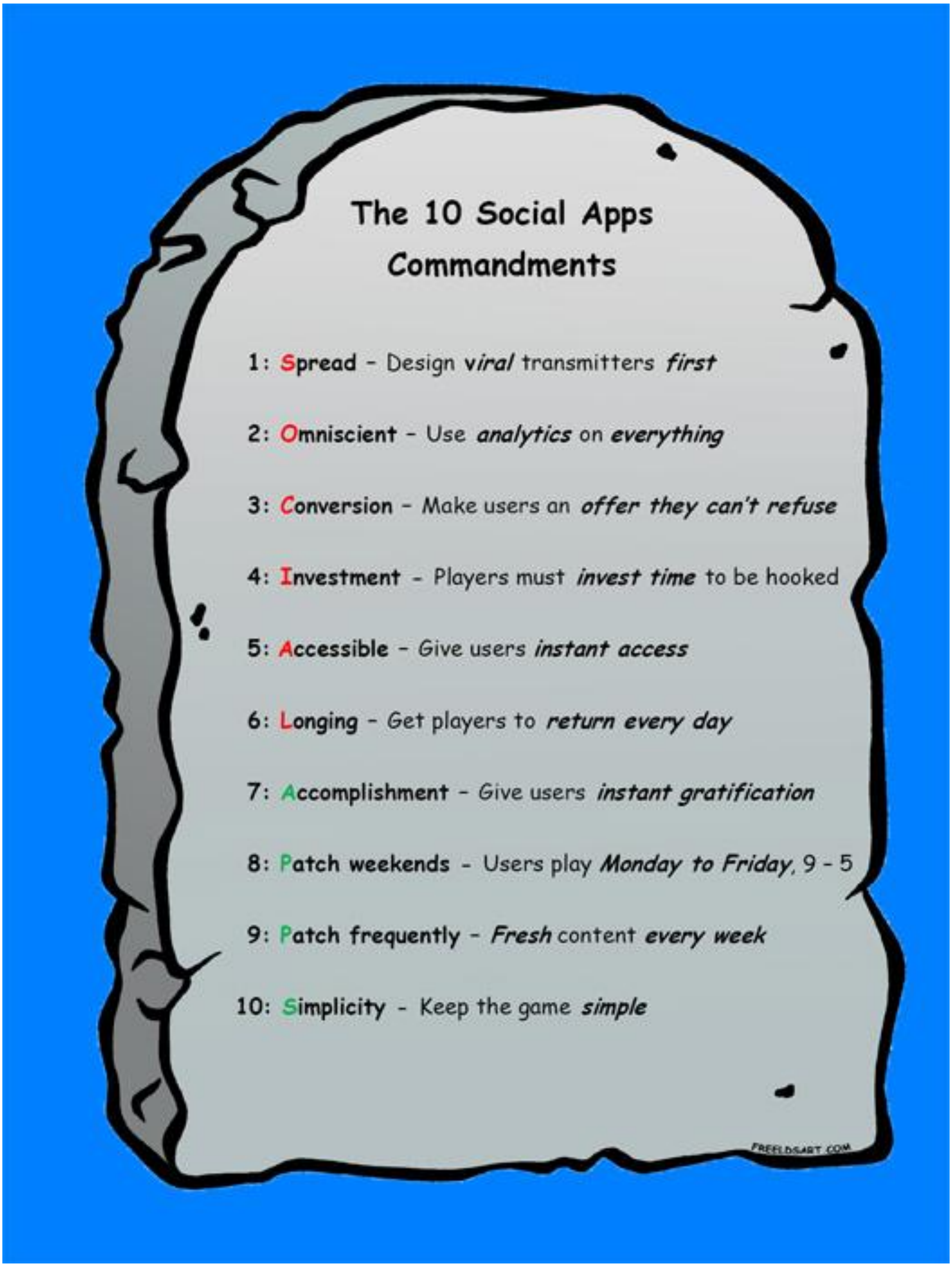
Another name worth mentioning in this context is the company Playfish, bought up by Electronic Arts in 2009, who has made several strikingly similar games to those of Zynga (Zynga has been openly criticized for plagiarism of Playfish's games on several occasions) and had their first major hit with the Facebook game 'Who has the biggest Brain?'. Incidentally this game was the first to attract millions of daily players on Facebook.

The cons, however, are many. Creating games for Facebook is very challenging, as it was never really made with gaming purposes in mind.

Generally speaking, it has a bad API prone to produce seemingly random errors, and because they update relatively often but infrequently it has proven difficult to find accurate code examples and to know when to update which part of the application you’re working on.

8.5 The 10 Social Apps Commandments

Here is the original image of The 10 Social Apps Commandments that the group has been trying to adhere to during our production of the game



The 10 Social Apps Commandments

1: **Spread** - Design *viral* transmitters *first*

2: **Omniscient** - Use *analytics* on *everything*

3: **Conversion** - Make users an *offer they can't refuse*

4: **Investment** - Players must *invest time* to be hooked

5: **Accessible** - Give users *instant access*

6: **Longing** - Get players to *return every day*

7: **Accomplishment** - Give users *instant gratification*

8: **Patch weekends** - Users play *Monday to Friday*, 9 - 5

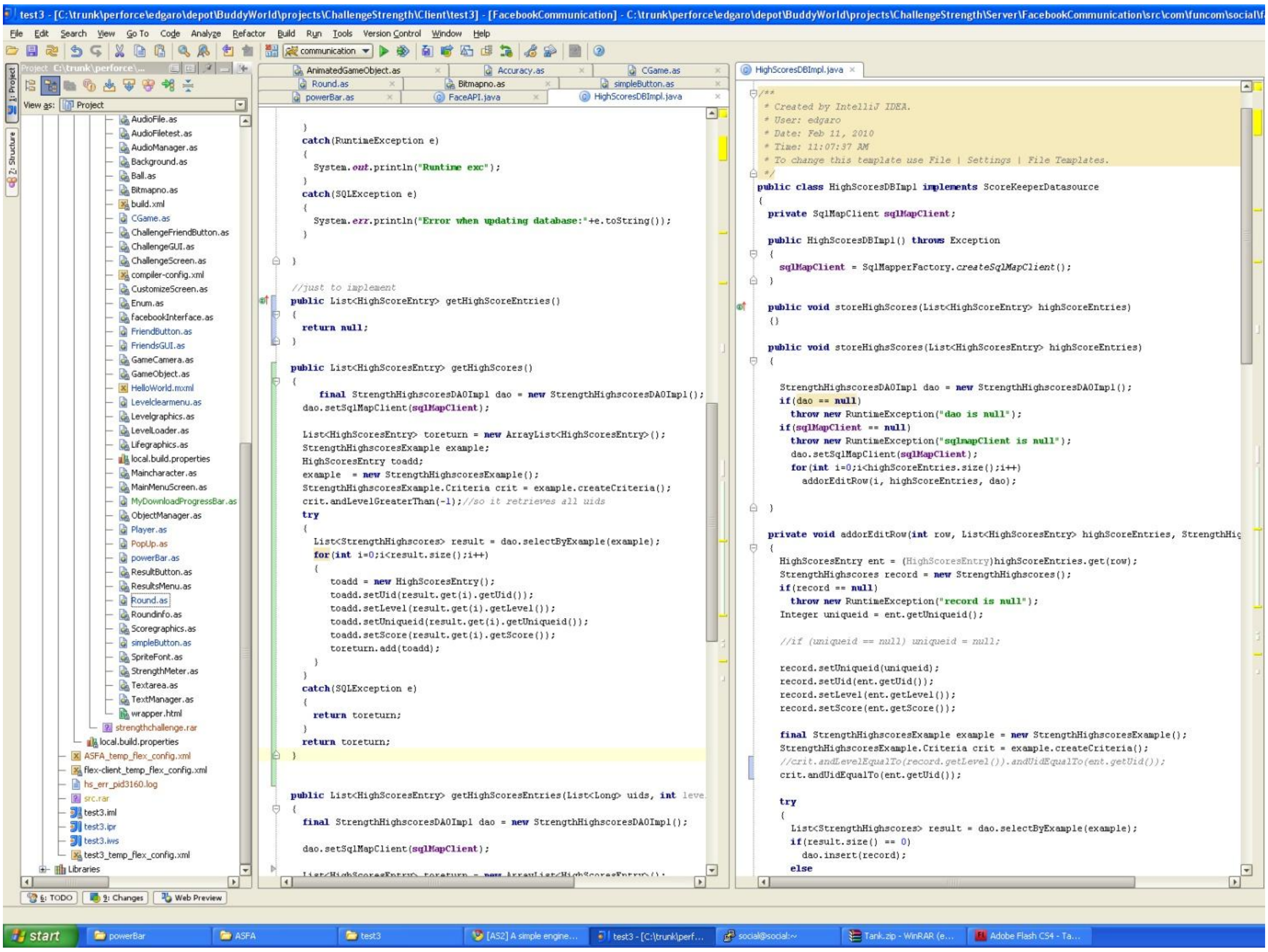
9: **Patch frequently** - *Fresh* content *every week*

10: **Simplicity** - Keep the game *simple*

These 10 advices served as guidelines for the group and for the Social Games Division, and offer plain, clearly defined rules for where a game designer working with social games should focus his or her efforts in order to make a successful game. They include directions on how to market a social game, how to get players hooked and even tips about patching and updating the game.
These commandments were carefully selected by Jarle Snertingdalen, producer at FunCom's Social Games division, and apply to every member of a small, social-apps team as they include every part of the development process.

8.6 WIP screenshots

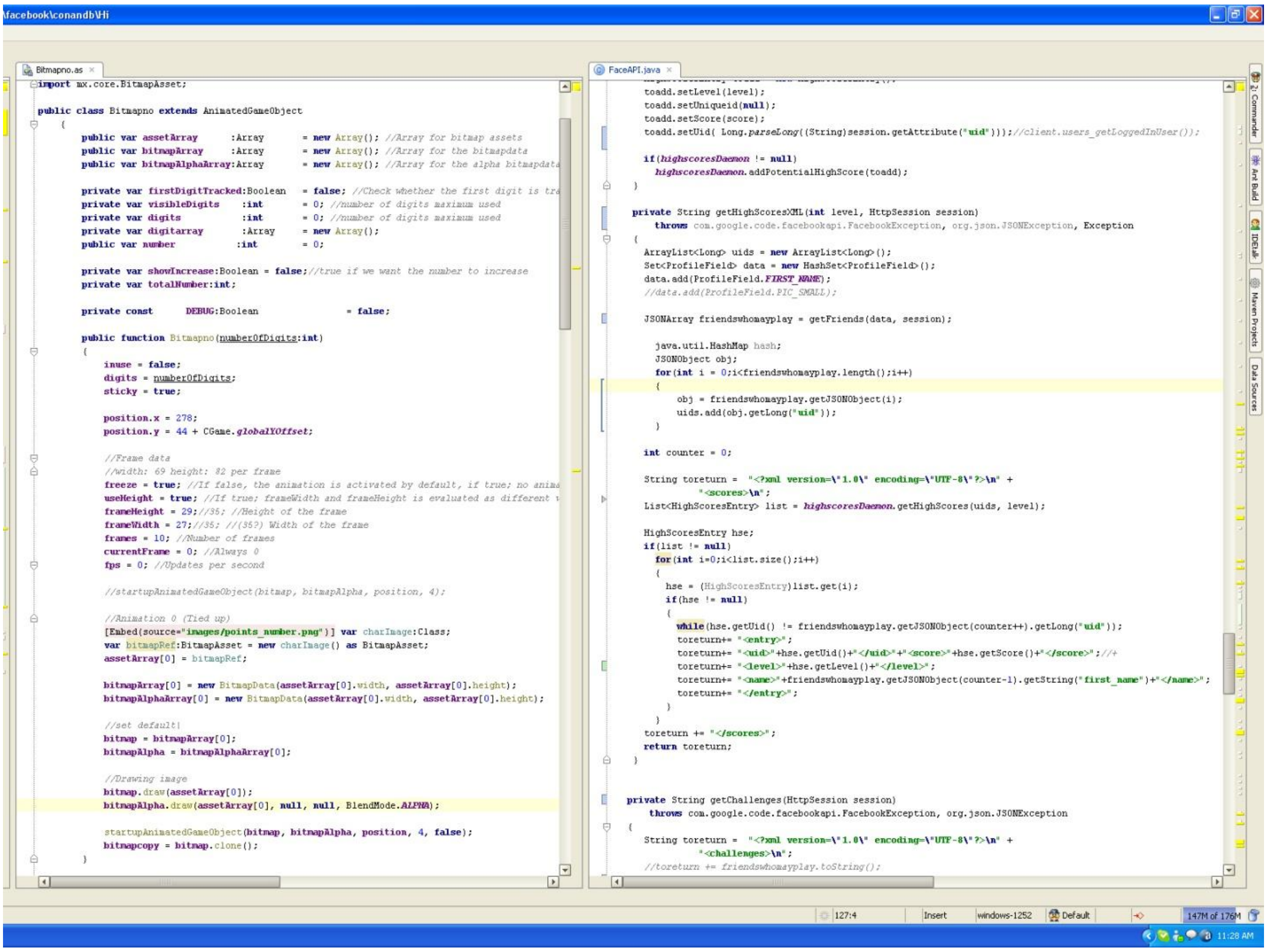
Here we have included a selection of screenshots that show our work, including examples of coding, graphics and animation production as well as the final product.



```
    }
    catch(RuntimeException e)
    {
      System.out.println("Runtime exc");
    }
    catch(SQLException e)
    {
      System.err.println("Error when updating database:"+e.toString());
    }

  }

  //just to implement
  public List<HighScoreEntry> getHighScoreEntries()
  {
    return null;
  }

  public List<HighScoresEntry> getHighScores()
  {
       final StrengthHighscoresDAOImpl dao = new StrengthHighscoresDAOImpl();
    dao.setSqlMapClient(sqlMapClient);

    List<HighScoresEntry> toreturn = new ArrayList<HighScoresEntry>();
    StrengthHighscoresExample example;
    HighScoresEntry toadd;
    example  = new StrengthHighscoresExample();
    StrengthHighscoresExample.Criteria crit = example.createCriteria();
    crit.andLevelGreaterThan(-1);//so it retrieves all uids
    try
    {
      List<StrengthHighscores> result = dao.selectByExample(example);
      for(int i=0;i<result.size();i++)
      {
         toadd = new HighScoresEntry();
         toadd.setUid(result.get(i).getUid());
         toadd.setLevel(result.get(i).getLevel());
         toadd.setUniqueid(result.get(i).getUniqueid());
         toadd.setScore(result.get(i).getScore());
         toreturn.add(toadd);
      }
    }
    catch(SQLException e)
    {
      return toreturn;
    }
    return toreturn;
  }


  public List<HighScoresEntry> getHighScoresEntries(List<Long> uids, int leve
  {
    final StrengthHighscoresDAOImpl dao = new StrengthHighscoresDAOImpl();

    dao.setSqlMapClient(sqlMapClient);
```

```
/**
 * Created by IntelliJ IDEA.
 * User: edgaro
 * Date: Feb 11, 2010
 * Time: 11:07:37 AM
 * To change this template use File | Settings | File Templates.
 */
public class HighScoresDBImpl implements ScoreKeeperDatasource
{
  private SqlMapClient sqlMapClient;

  public HighScoresDBImpl() throws Exception
  {
    sqlMapClient = SqlMapperFactory.createSqlMapClient();
  }

  public void storeHighScores(List<HighScoreEntry> highScoreEntries)
  {}

  public void storeHighsScores(List<HighScoresEntry> highScoreEntries)
  {

    StrengthHighscoresDAOImpl dao = new StrengthHighscoresDAOImpl();
    if(dao == null)
      throw new RuntimeException("dao is null");
    if(sqlMapClient == null)
      throw new RuntimeException("sqlmapClient is null");
      dao.setSqlMapClient(sqlMapClient);
      for(int i=0;i<highScoreEntries.size();i++)
        addorEditRow(i, highScoreEntries, dao);

  }

  private void addorEditRow(int row, List<HighScoresEntry> highScoreEntries, StrengthHig
  {
    HighScoresEntry ent = (HighScoresEntry)highScoreEntries.get(row);
    StrengthHighscores record = new StrengthHighscores();
    if(record == null)
      throw new RuntimeException("record is null");
    Integer uniqueid = ent.getUniqueid();

    //if (uniqueid == null) uniqueid = null;

    record.setUniqueid(uniqueid);
    record.setUid(ent.getUid());
    record.setLevel(ent.getLevel());
    record.setScore(ent.getScore());

    final StrengthHighscoresExample example = new StrengthHighscoresExample();
    StrengthHighscoresExample.Criteria crit = example.createCriteria();
    //crit.andLevelEqualTo(record.getLevel()).andUidEqualTo(ent.getUid());
    crit.andUidEqualTo(ent.getUid());

    try
    {
      List<StrengthHighscores> result = dao.selectByExample(example);
      if(result.size() == 0)
        dao.insert(record);
      else
```

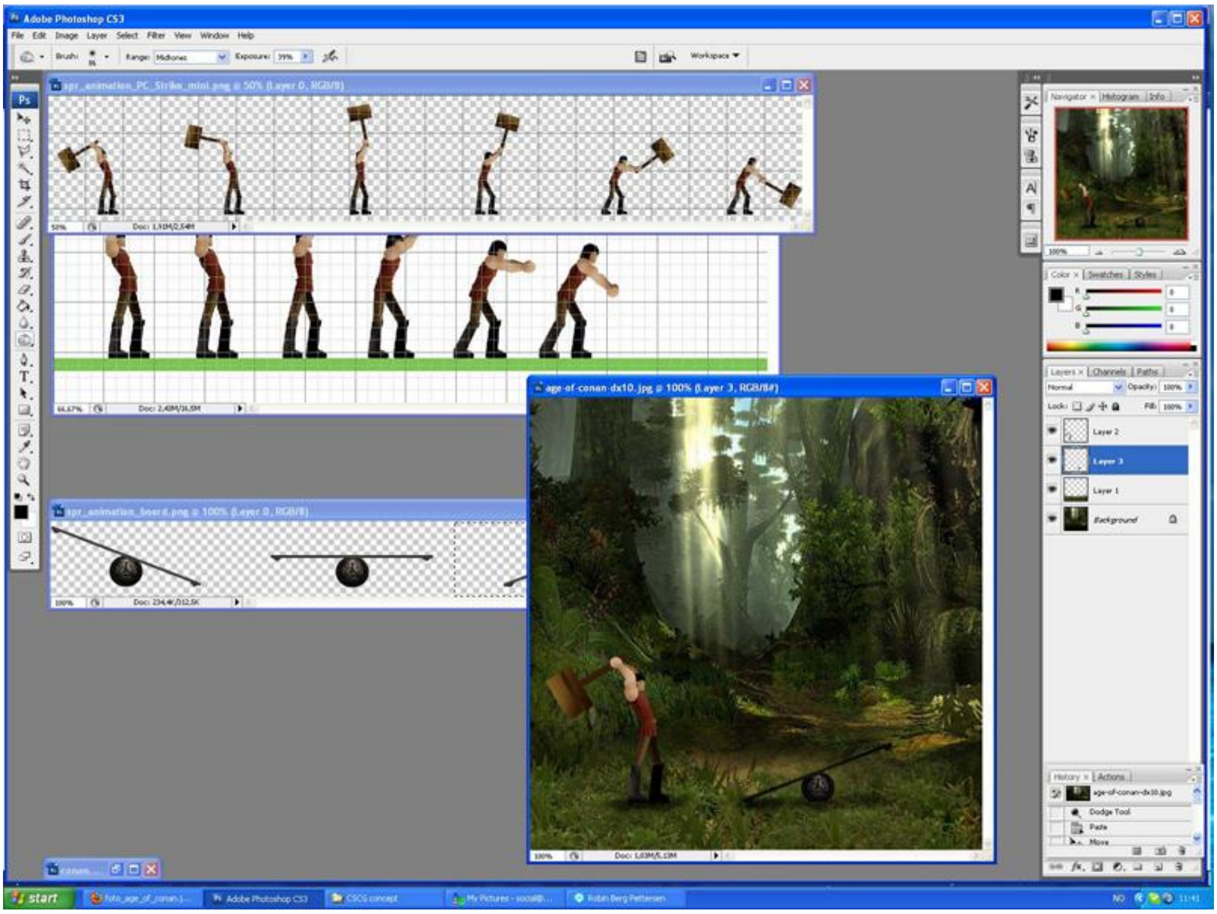
Coding in Flex 1/2



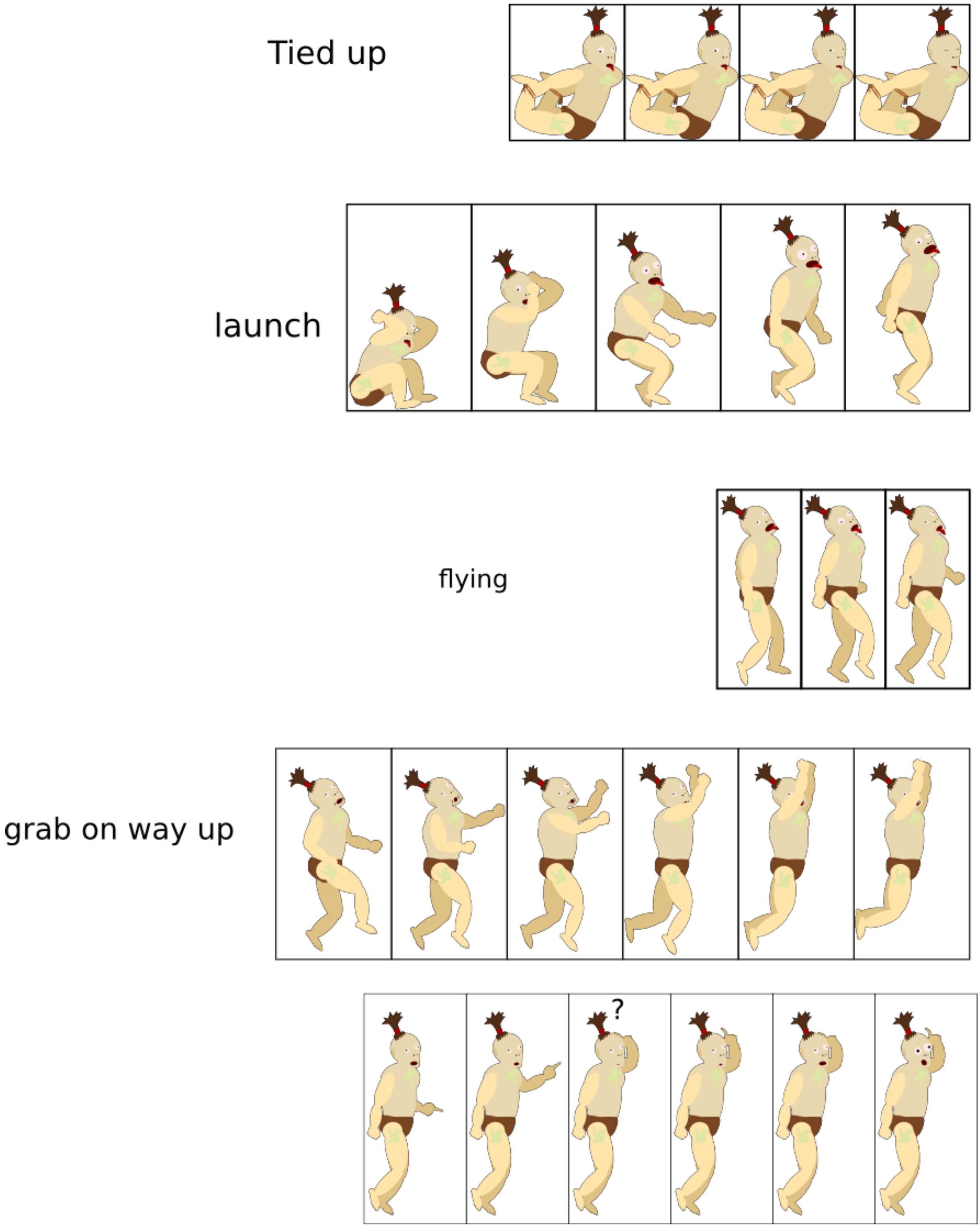
```
import mx.core.BitmapAsset;

public class Bitmapno extends AnimatedGameObject
    {
        public var assetArray      :Array       = new Array(); //Array for bitmap assets
        public var bitmapArray     :Array       = new Array(); //Array for the bitmapdata
        public var bitmapAlphaArray:Array       = new Array(); //Array for the alpha bitmapdata

        private var firstDigitTracked:Boolean   = false; //Check whether the first digit is tra
        private var visibleDigits    :int       = 0; //number of digits maximum used
        private var digits           :int       = 0; //number of digits maximum used
        private var digitarray       :Array     = new Array();
        public var number           :int        = 0;

        private var showIncrease:Boolean = false;//true if we want the number to increase
        private var totalNumber:int;

        private const      DEBUG:Boolean                = false;

        public function Bitmapno(numberOfDigits:int)
        {
            inuse = false;
            digits = numberOfDigits;
            sticky = true;

            position.x = 278;
            position.y = 44 + CGame.globalYOffset;

            //Frame data
            //width: 69 height: 82 per frame
            freeze = true; //If false, the animation is activated by default, if true; no anima
            useHeight = true; //If true; frameWidth and frameHeight is evaluated as different v
            frameHeight = 29;//35; //Height of the frame
            frameWidth = 27;//35; //(35?) Width of the frame
            frames = 10; //Number of frames
            currentFrame = 0; //Always 0
            fps = 0; //Updates per second

            //startupAnimatedGameObject(bitmap, bitmapAlpha, position, 4);

            //Animation 0 (Tied up)
            [Embed(source="images/points_number.png")] var charImage:Class;
            var bitmapRef:BitmapAsset = new charImage() as BitmapAsset;
            assetArray[0] = bitmapRef;

            bitmapArray[0] = new BitmapData(assetArray[0].width, assetArray[0].height);
            bitmapAlphaArray[0] = new BitmapData(assetArray[0].width, assetArray[0].height);

            //set default|
            bitmap = bitmapArray[0];
            bitmapAlpha = bitmapAlphaArray[0];

            //Drawing image
            bitmap.draw(assetArray[0]);
            bitmapAlpha.draw(assetArray[0], null, null, BlendMode.ALPHA);

            startupAnimatedGameObject(bitmap, bitmapAlpha, position, 4, false);
            bitmapcopy = bitmap.clone();
        }
```

```
    toadd.setLevel(level);
    toadd.setUniqueid(null);
    toadd.setScore(score);
    toadd.setUid( Long.parseLong((String)session.getAttribute("uid")));//client.users_getLoggedInUser());

    if(highscoresDaemon != null)
      highscoresDaemon.addPotentialHighScore(toadd);
  }

  private String getHighScoresXML(int level, HttpSession session)
      throws com.google.code.facebookapi.FacebookException, org.json.JSONException, Exception
  {
    ArrayList<Long> uids = new ArrayList<Long>();
    Set<ProfileField> data = new HashSet<ProfileField>();
    data.add(ProfileField.FIRST_NAME);
    //data.add(ProfileField.PIC_SMALL);

    JSONArray friendswhomayplay = getFriends(data, session);

      java.util.HashMap hash;
      JSONObject obj;
      for(int i = 0;i<friendswhomayplay.length();i++)
      {
          obj = friendswhomayplay.getJSONObject(i);
          uids.add(obj.getLong("uid"));
      }

    int counter = 0;

    String toreturn =  "<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n" +
              "<scores>\n";
    List<HighScoresEntry> list = highscoresDaemon.getHighScores(uids, level);

    HighScoresEntry hse;
    if(list != null)
      for(int i=0;i<list.size();i++)
      {
        hse = (HighScoresEntry)list.get(i);
        if(hse != null)
        {
          while(hse.getUid() != friendswhomayplay.getJSONObject(counter++).getLong("uid"));
          toreturn+= "<entry>";
          toreturn+= "<uid>"+hse.getUid()+"</uid>"+"<score>"+hse.getScore()+"</score>";//+
          toreturn+= "<level>"+hse.getLevel()+"</level>";
          toreturn+= "<name>"+friendswhomayplay.getJSONObject(counter-1).getString("first_name")+"</name>";
          toreturn+= "</entry>";
        }
      }
    toreturn += "</scores>";
    return toreturn;
  }


 private String getChallenges(HttpSession session)
     throws com.google.code.facebookapi.FacebookException, org.json.JSONException
 {
    String toreturn =  "<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n" +
              "<challenges>\n";
    //toreturn += friendswhomayplay.toString();
```

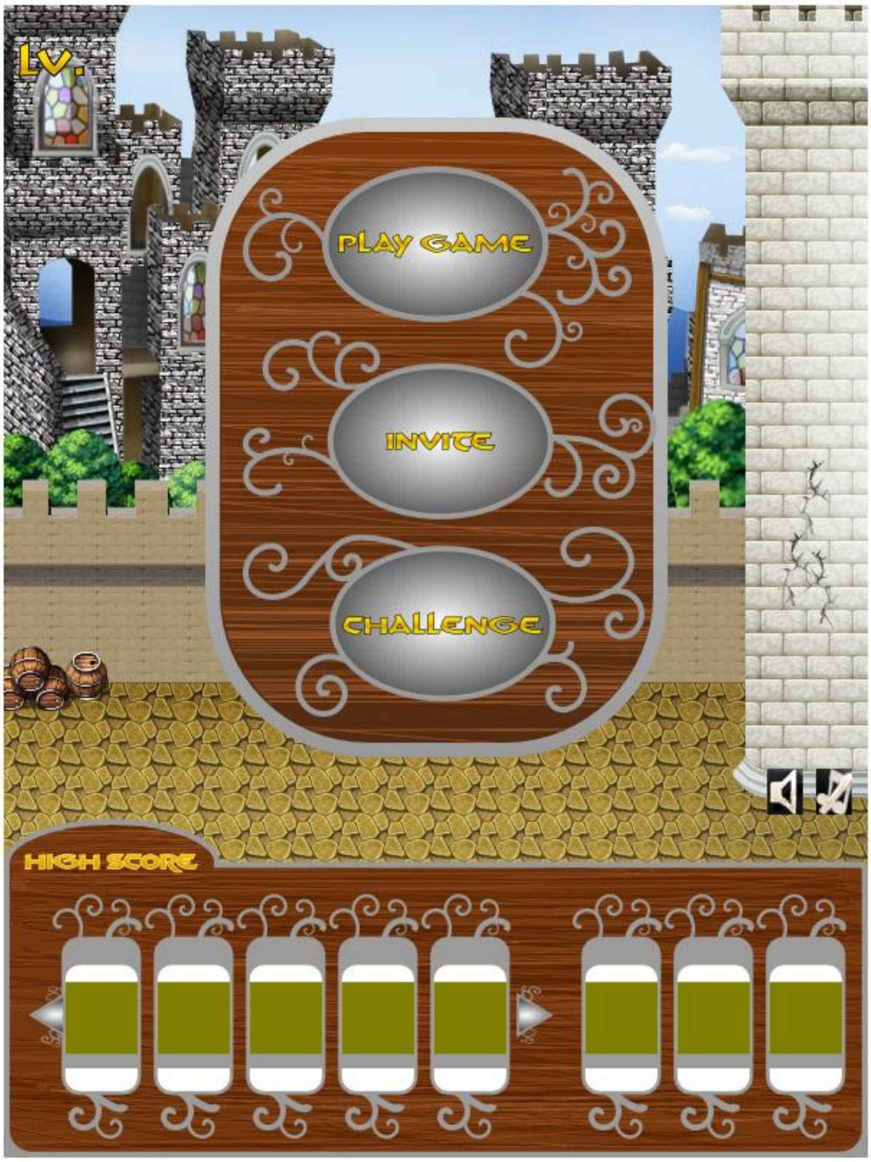
Coding in Flex 2/2



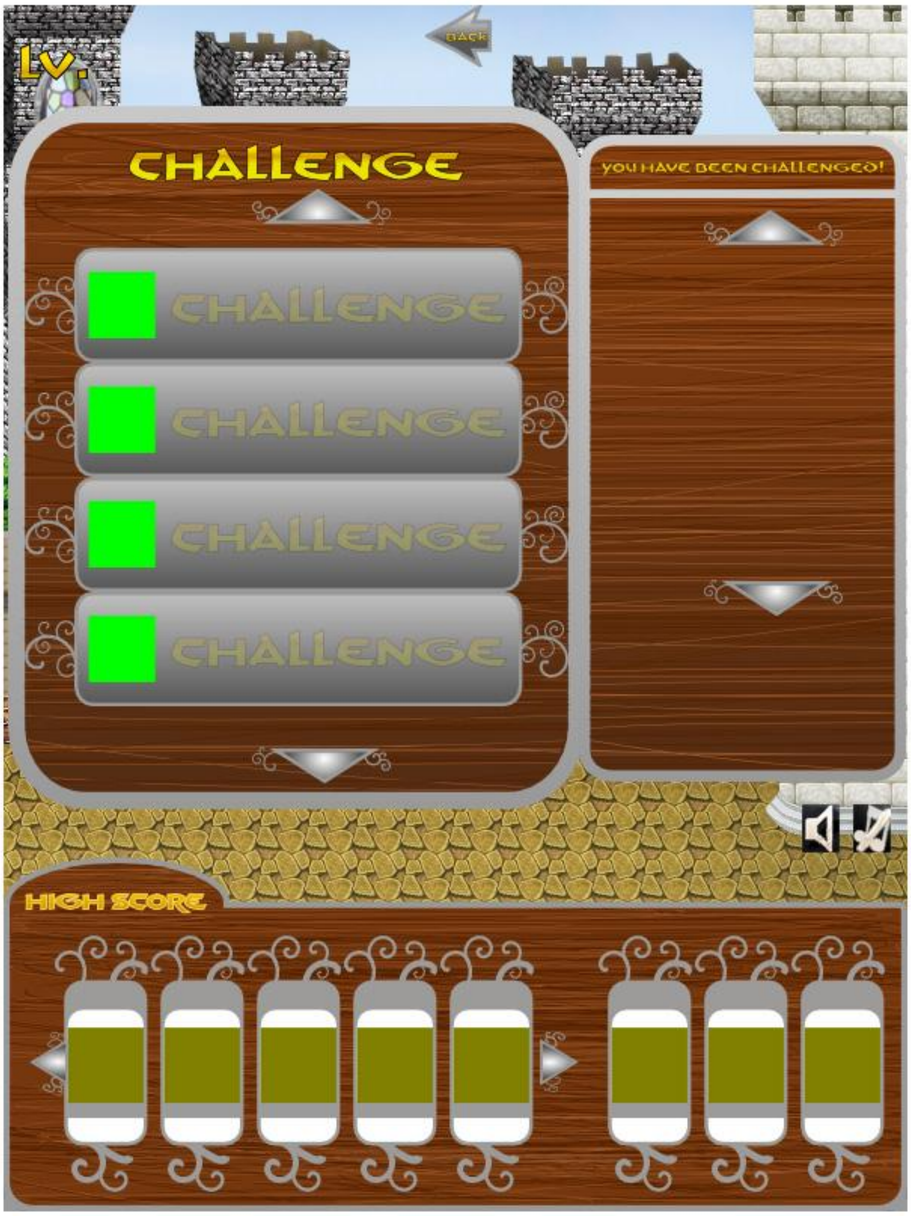
Graphic design in Photoshop



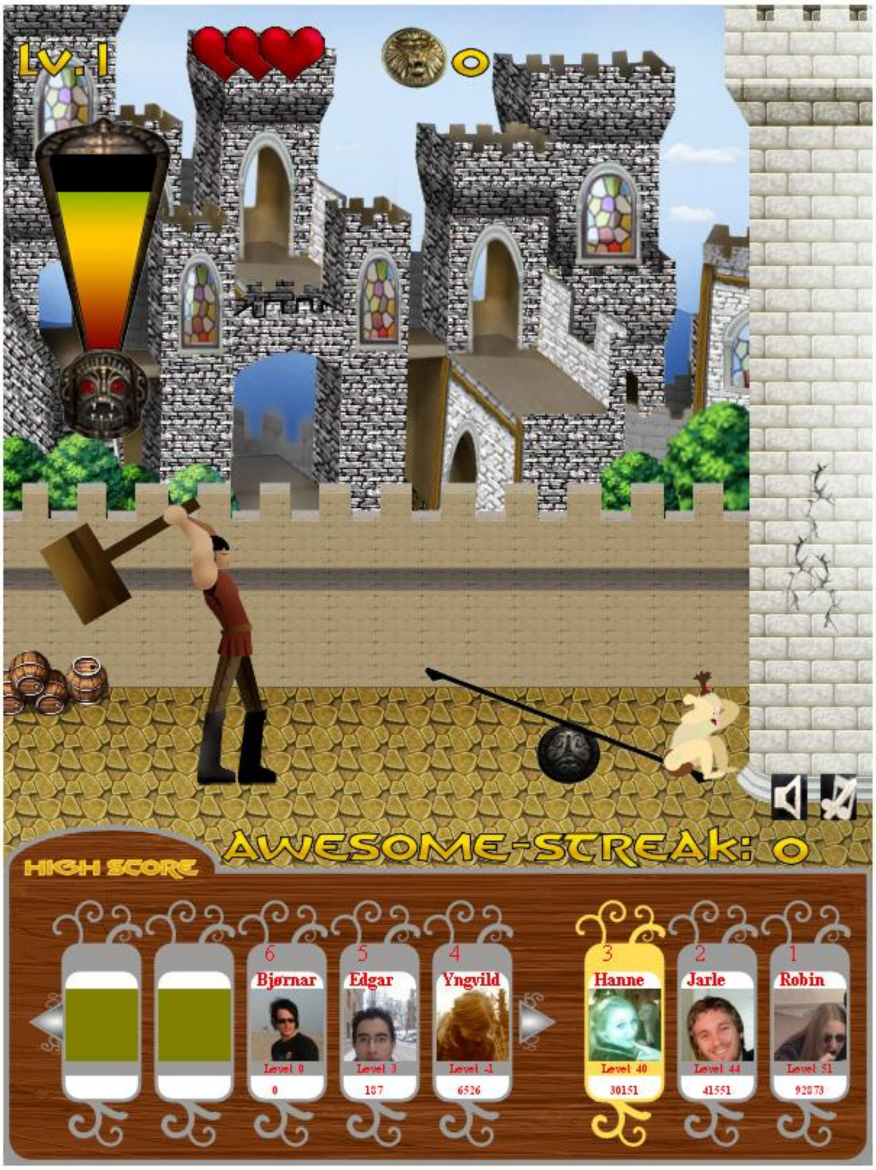
Pict animation spritesheet done in Inkscape



Final game main menu



Final game challenge menu



Final game in-game

8.7 Art Request List

Here is the full list of Art Requests for the game. Since the current graphics are not en par with FunCom's standards, this is what would need to be (re)produced before an eventual release.

<u>Loading Screen</u>

Background size: fills the entire game window

Loading animation size: 383 x 116 pxl

Format: PNG image

Appears at once when the player chooses to open the application from Facebook

Background contains the game's title at the top in big letters

The loading animation consists of a static frame around a colored bar that fills itself up as the game loads

Once loading is complete the screen disappears to be replaced by the main menu screen

Background

Size 650 x 1800 pxl

Note that this is slightly larger than what is strictly necessary

Format: PNG image

Amount: min. 1 (there are currently 3)

Ideally there should be 10 different images (so that they could change for each 10th new level the player reaches, estimating that no one would likely progress to level 100+)

These can show either completely different locations, or be variations of one scenery (as has been done in the prototype version)

Specific Requirements: The bottom 648 x 260 pxl area of the background image should preferably have only the ground and a plain wall, since this is where the PC will be standing and he should be in absolute focus

The right side of the screen is dominated by a tower (that the Pict crashes into)

Tower

Tower size 128 x 1024 pxl

Format PNG image

Tower position on screen: 515 x 50 pxl

Player Character

Size (of one PC): Height head to toe: 60 x 173 pxl

Height standing with hammer raised: 62 x 297 pxl (hammer included)

Format: PNG image

Animations: Idle/waiting, Striking, Bad result pose/failed, Mediocre/OK result pose, Awesome result pose

Specific Requirements: All animations should be as quick an expressive as possible

GUI

Size: approx. 50 x 50 pxl (all icons)

Format: PNG image

Types (top-left to right): Level (“Lv.”), Lives, Points, Streak, mute music button, mute

SFX button

The numbers representing the current score is positioned directly after the points icon. When the Pict goes up the points are already being calculated and this is displayed by numbers counting upwards in roughly the middle of the screen, and then when it's stopped counting the final result floats up to merge with the current score, updating immediately

Font: The Hyberian font (color: yellow/golden) should be used for titles and short sentences (e.g. You Won!), while longer sentences (explanations etc.) should use the font Times New Roman in the same color

Pop-ups

Size: 195 x 175 pxl

Format: PNG image

Variations: end of round, new high score, you are challenging X with score X, you won/lost the challenge with X points

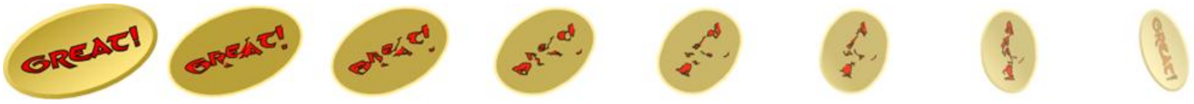
Features share/skip buttons with hover and click effects

Feedback animations

Size: approx. 270 x 150

Format GIF image

Types: failed strike, bad strike, not bad/mediocre strike, awesome strike, you got an



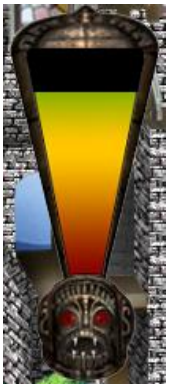
extra life

[The feedback animation for a Great Strike]

Animations appear at the center of the screen

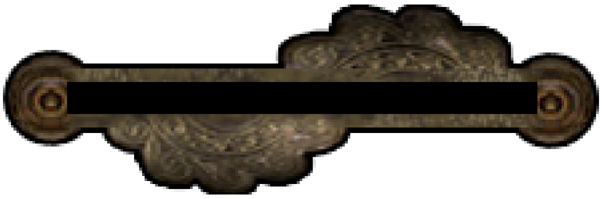
Game elements

Power bar



- Size: approx. 150 x 300 pxl
- Format: PNG image
- Consists of a static frame around a colored area that is animated to fill up
- Positioned to the left of the Player Character on the screen

Accuracy bar



- Size: approx. 300 x 70 pxl
- Format PNG image
- Consists of a static frame around a red bar. Within the red bar is a smaller orange bar with an even smaller green bar placed on top of it. The orange and green area must have an adjustable size

The accuracy bar also has a small arrow that moves back and forth across the colored bar

Main menu Screen elements

Window size (the square upon which the buttons are placed): 325 x 445 pxl

Button size: 158 x 110 pxl

Challenge menu Screen elements

Main window (where it says “Challenge”)

- Size: 384 x 467 pxl
- Format: PNG image
- Challenge-Friends bar size: 299 x 78 pxl
- This bar has a hover effect

Side window (where it says “You have been challenged”)

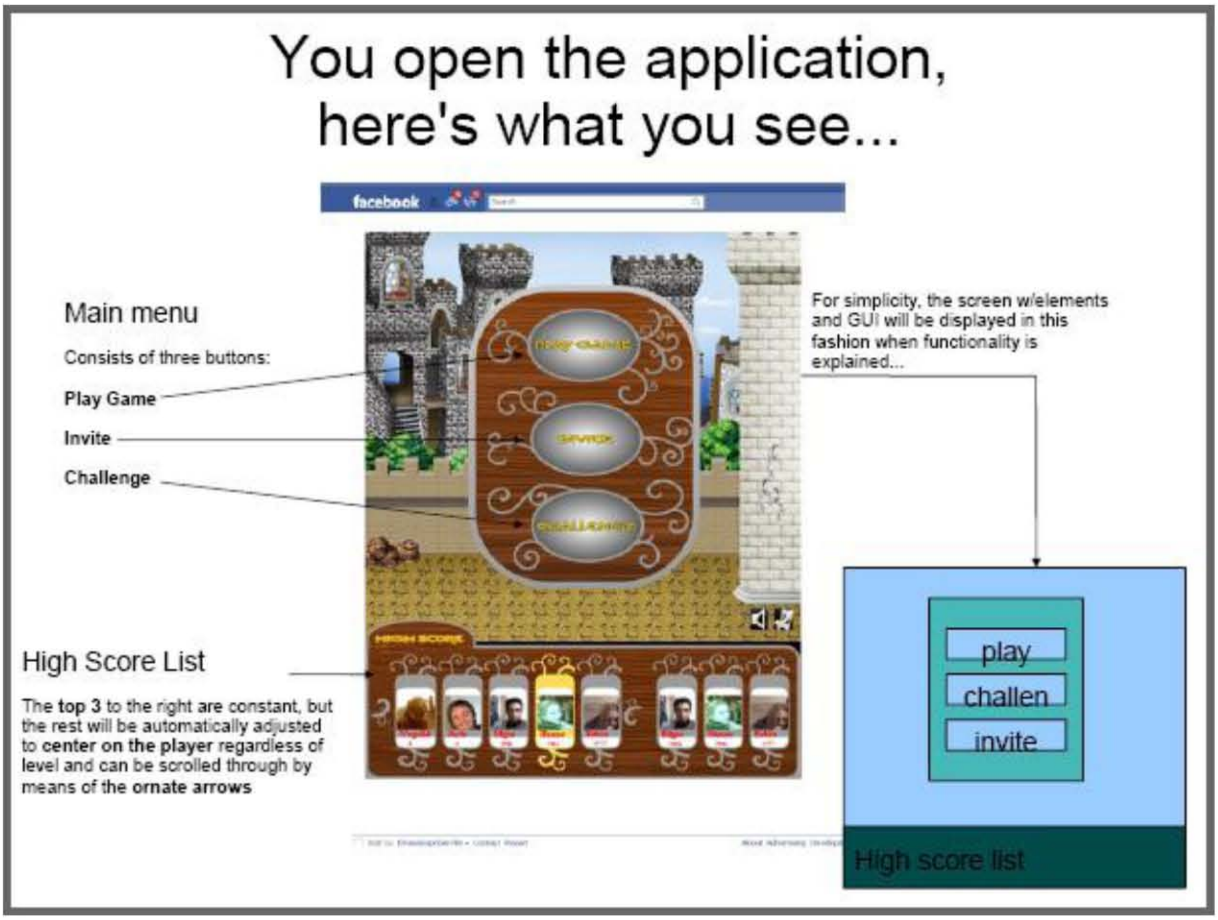
- Size: 218 x 433 pxl
- Format: PNG image

Back button and Up/Down Arrows

- Button size:55 x 40 pxl
- Arrow size: 76 x 28 pxl
- Format: PNG image
- Effect: hover and click effect

8.8 Game Flow

These screens are taken directly from a PSD file that presents the game flow.



You open the application, here's what you see...

Main menu

Consists of three buttons:

Play Game

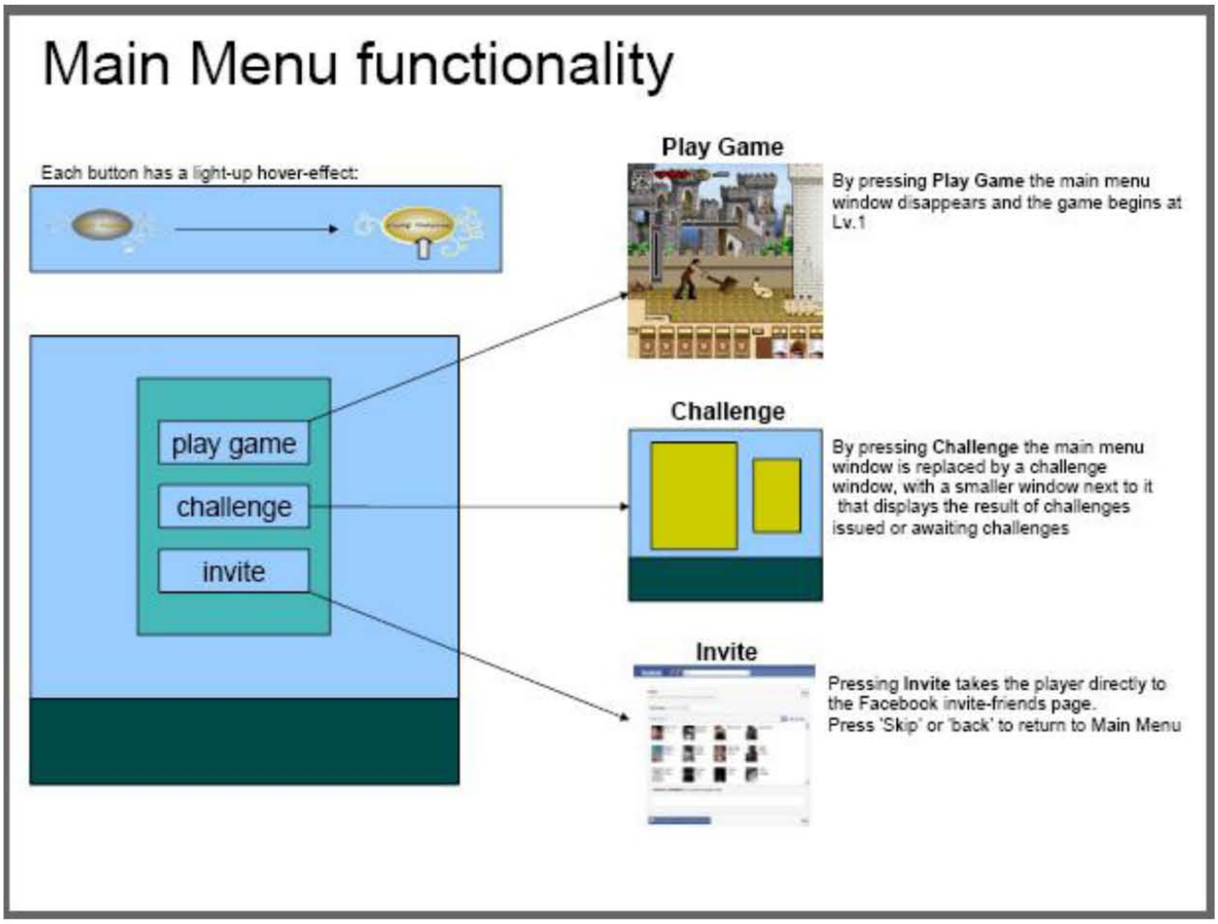
Invite

Challenge

High Score List

The **top 3** to the right are constant, but the rest will be automatically adjusted to **center on the player** regardless of level and can be scrolled through by means of the **ornate arrows**

For simplicity, the screen w/elements and GUI will be displayed in this fashion when functionality is explained...



Main Menu functionality

Each button has a light-up hover-effect:

Play Game

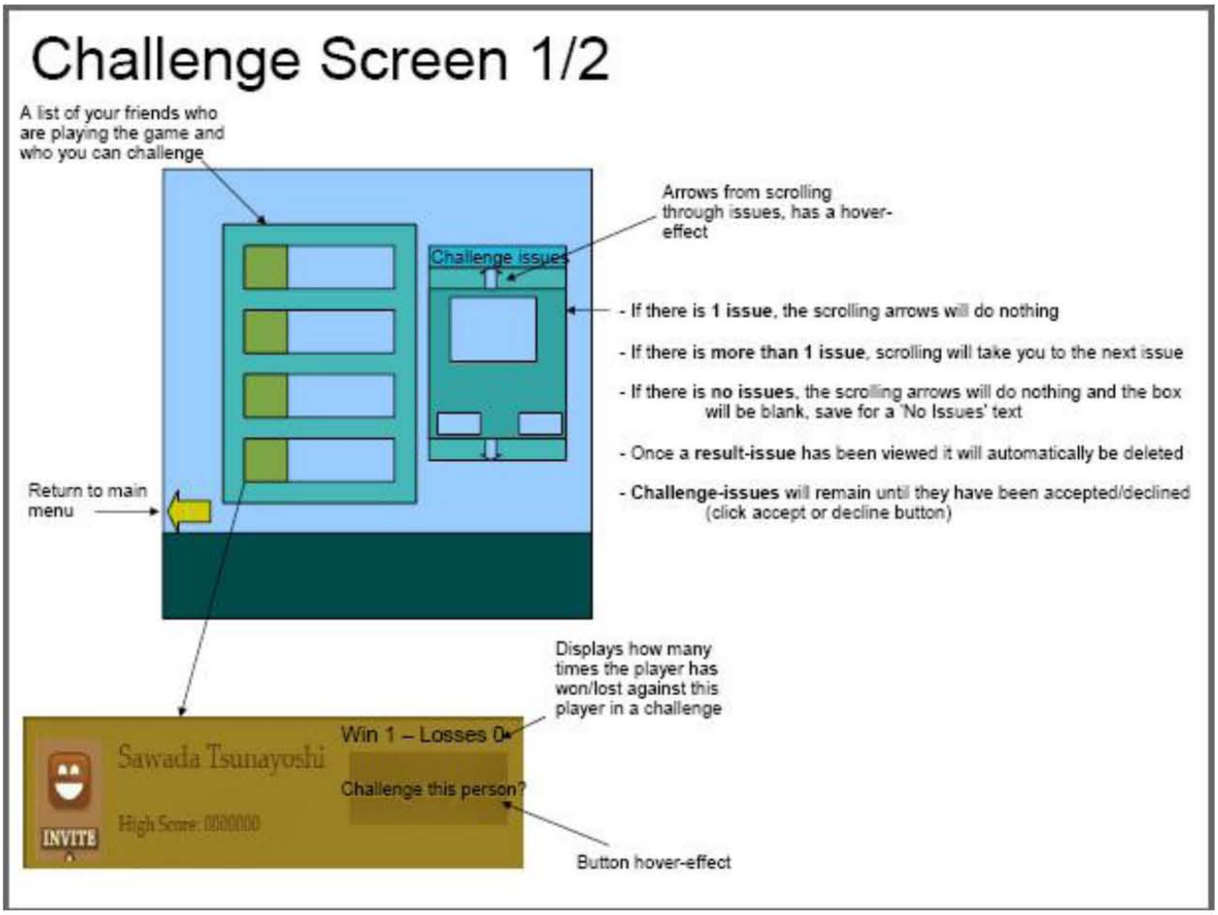
By pressing **Play Game** the main menu window disappears and the game begins at Lv.1

Challenge

By pressing **Challenge** the main menu window is replaced by a challenge window, with a smaller window next to it that displays the result of challenges issued or awaiting challenges

Invite

Pressing **Invite** takes the player directly to the Facebook invite-friends page.
Press 'Skip' or 'back' to return to Main Menu



Challenge Screen 1/2

A list of your friends who are playing the game and who you can challenge

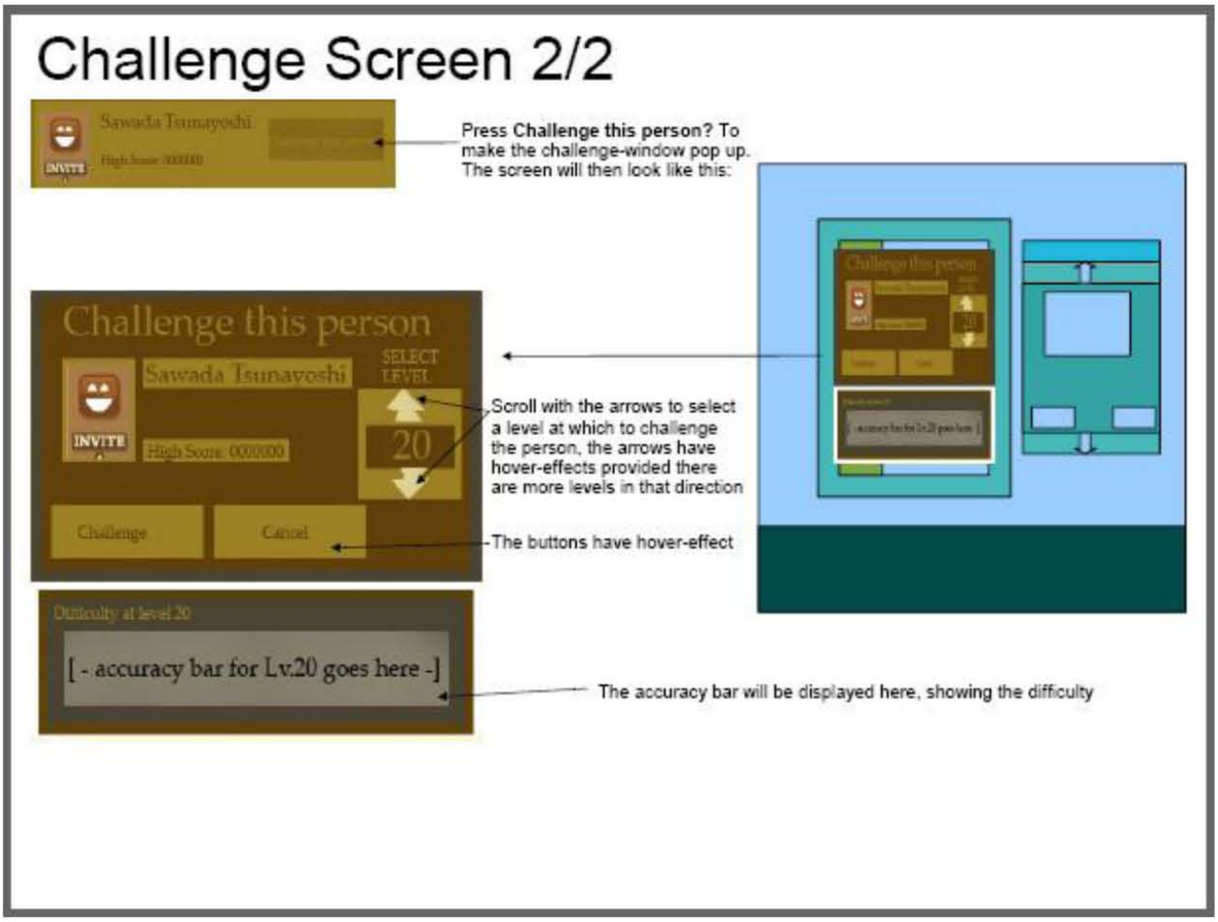
Arrows from scrolling through issues, has a hover-effect

- If there is **1 issue**, the scrolling arrows will do nothing
- If there is **more than 1 issue**, scrolling will take you to the next issue
- If there is **no issues**, the scrolling arrows will do nothing and the box will be blank, save for a 'No Issues' text
- Once a **result-issue** has been viewed it will automatically be deleted
- **Challenge-issues** will remain until they have been accepted/declined (click accept or decline button)

Return to main menu

Displays how many times the player has won/lost against this player in a challenge

Button hover-effect



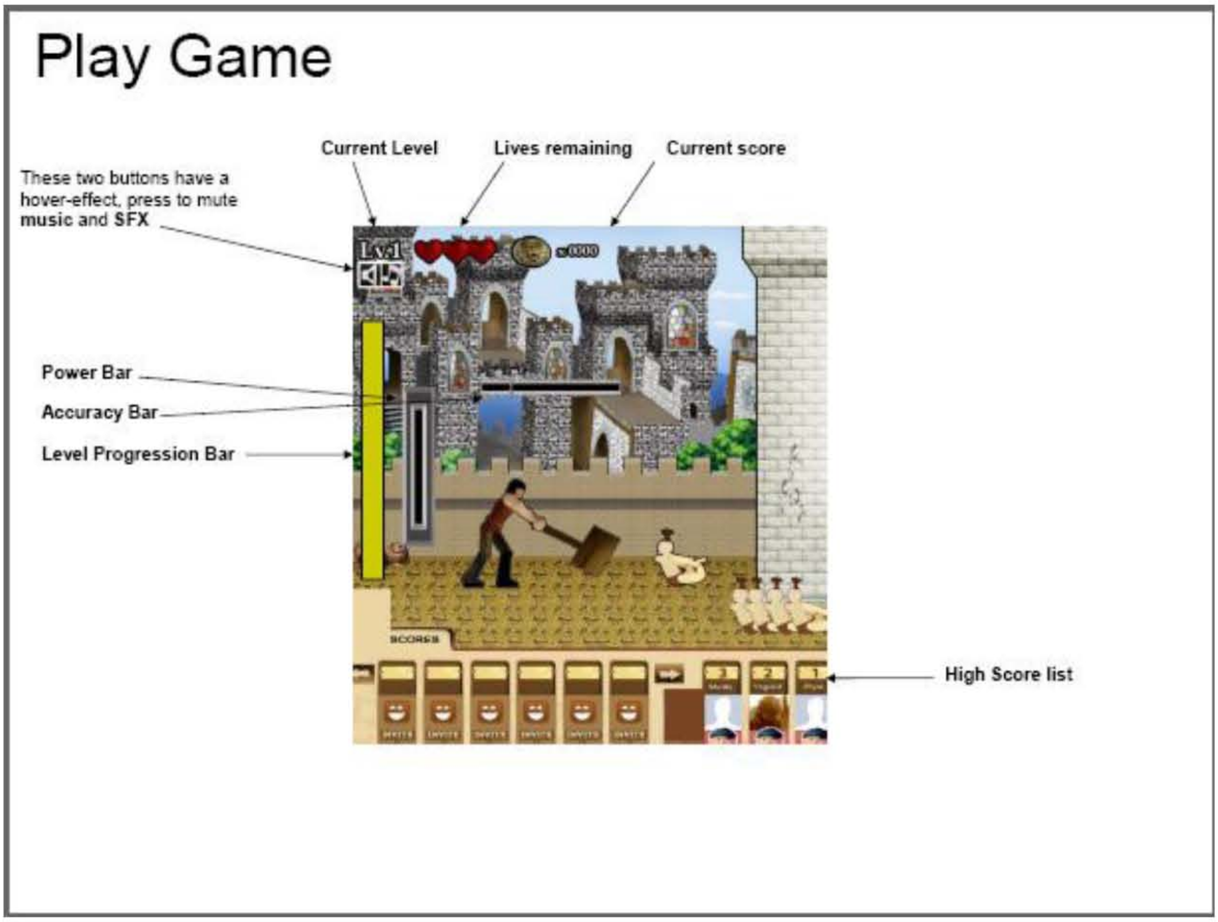
Challenge Screen 2/2

Press **Challenge this person?** To make the challenge-window pop up. The screen will then look like this:

Scroll with the arrows to select a level at which to challenge the person, the arrows have hover-effects provided there are more levels in that direction

The buttons have hover-effect

The accuracy bar will be displayed here, showing the difficulty



Play Game

Current Level

Lives remaining

Current score

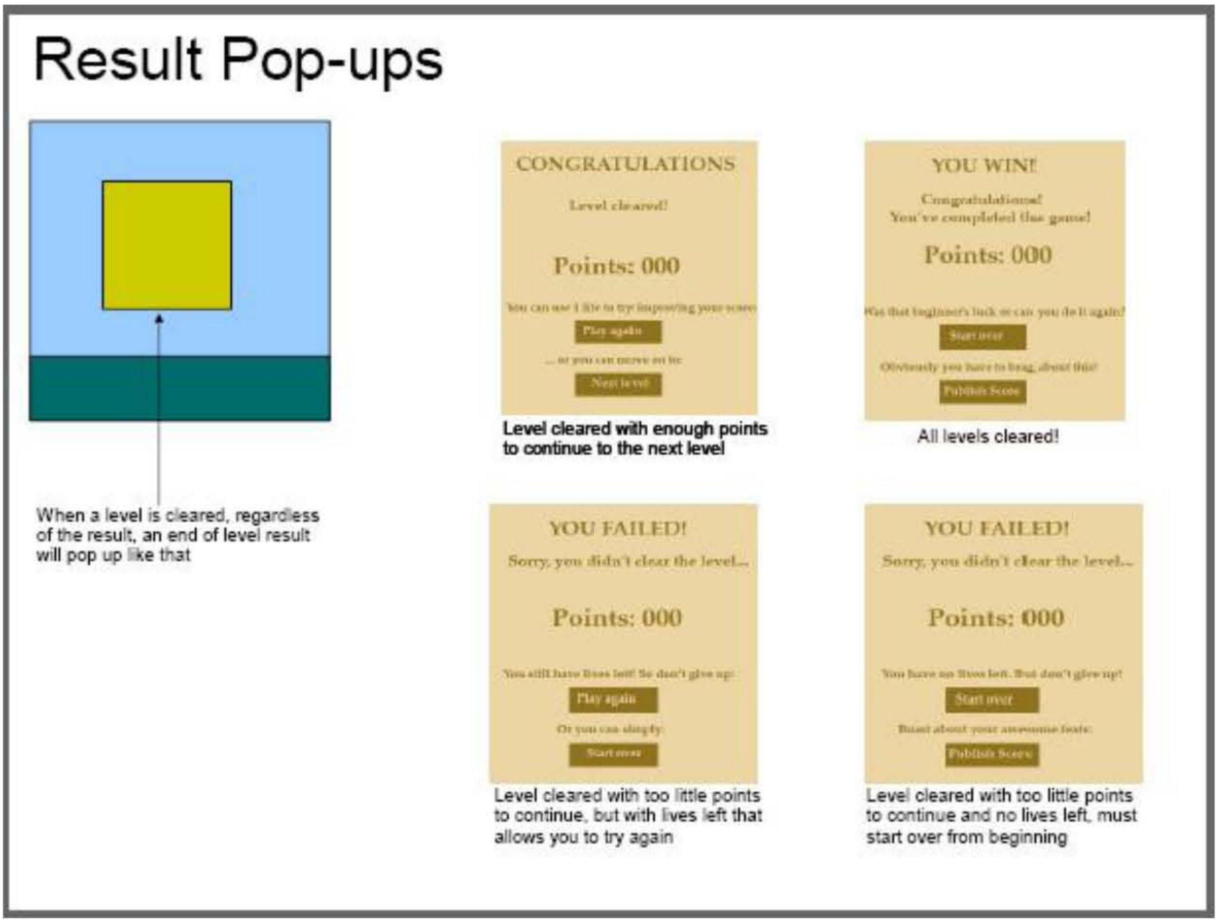
These two buttons have a hover-effect, press to mute music and SFX

Power Bar

Accuracy Bar

Level Progression Bar

High Score list



Result Pop-ups

When a level is cleared, regardless of the result, an end of level result will pop up like that

CONGRATULATIONS

Level cleared!

Points: 000

You can use 1 life to try improving your score:

Play again

... or you can move on to:

Next level

Level cleared with enough points to continue to the next level

YOU WIN!

Congratulations!
You've completed the game!

Points: 000

Was that beginner's luck or can you do it again?

Start over

Obviously you have to brag about this!

Publish Score

All levels cleared!

YOU FAILED!

Sorry, you didn't clear the level...

Points: 000

You still have lives left! So don't give up:

Play again

Or you can simply:

Start over

Level cleared with too little points to continue, but with lives left that allows you to try again

YOU FAILED!

Sorry, you didn't clear the level...

Points: 000

You have no lives left. But don't give up!

Start over

Boast about your awesome feats:

Publish Score

Level cleared with too little points to continue and no lives left, must start over from beginning