Post mortem - UCM PROYECTO II

Autores - Mad Diggers

- Cañellas, Lluis LluiCS lluiscan@ucm.es
- de los Santos, Mario MarioDeLosSantos mdelos02@ucm.es
- Docampo, Marcos marcosdo marcosdo@ucm.es
- García, Jorge jorgar17 jorgar17@ucm.es
- Gavilán, Sergio srgxv1 sgavil01@ucm.es
- Mateos, Diego dimateos dimateos@ucm.es
- Serna, Álvaro alsern01 alsern01@ucm.es
- Suárez, Antonio antonsua antonsua@ucm.es

Planificación

La planificación escrita concreta se encuentra en el documento **file**, pero en nuestro caso no funcionó, y la dejamos de escribir alrededor de las 4-5 semanas.

En un principio lo hablamos en equipo y probamos pivotal traker, pero no todo el equipo lo veia necesario y nos pasamos a solo un documento escrito. Diseñado y redactado por Diego Mateos.

La intención era reunirse cada lunes y plantear las tareas de la semana, y al lunes posterior hablar de lo conseguido/fallado y reajustar constantemente la planificacion para cada semana.

El problema apareció cuando tras varias semanas parte del equipo no cumplia los objetivos y reajustar tanto la planificación cada semana paso a ser una perdida de tiempo para los que sí cumplian los objetivos. Como los que escribian activamente la planificación eran los integrantes que cumplian la mayoria de objetivos, se paso a hablar verbalmente de que se encargaria cada persona dispuesta a trabajar.

Después de Semana Santa, se volvio a intentar planificar activamente usando waffle.io, pero de nuevo se quedo en intento. Se rellenaron las tareas y algunos integrantes las ponian como completadas y otros se olviraron completamente. Por ello paso a ser perdida de tiempo de los que si lo usaban activamente y decidieron dejarlo y volver a hablar verbalmente de las tareas que cada uno estaba dispuesto a hacer.

Llegados a este punto, hubo varias reuniones a ver si el equipo se motivaba y trabajaba más, pero a pesar de todos coincidir en querer hacer un buen juego para algunos se quedo en palabras.

Diego Mateos habló con el equipo de la falta de participación equilibrada, y de que se lo comentaría a Carlos para buscar una solución. Tras hablarlo con Carlos realizamos una encuesta de participación en la realización al juego. Aquella no era la definitiva pero servía para hacerse una idea e intentar que el equipo quisiera balancearla.

Dos semanas después llega la presentación final. El equipo ha trabajado bastante en general, pero sin equilibrarse la participación ya que los integrantes que habían trabajado más, también se han esforzado más para poder entregar un juego que se acercase todo lo posible a lo que les hubiera gustado.

Reflexión

De los éxitos y de los no tan éxitos hay que reflexionar y aprender. Distinguir lo que ha ido bien y lo que se podría haber mejorado. En nuestro caso, parece que no nos lo tomamos suficiente en serio y dejamos pasar señales fatales como permitir la ausencia de planificación.

Simplemente pensamos que era mas importante trabajar que escribir la planificación y que el grupo empezaría a trabajar más cuando las bases estuviesen hechas (arquitectura, diseño, etc). Pero no fue tal caso, al menos no lo suficiente como para coger el ritmo. Como los que no trabajaban tampoco escribian la planificación, los integrantes ocupados se siguieron programando etc sin hacer la planificación en pos de no perder tiempo.

Otro fallo importante que dejamos pasar es que siempre hay cosas que hacer aunque no todo el mundo pueda programar a la vez: arte, musica, diseño, balanceo, leer codigo para conocer la arquitectura y progamar mejor/evitar preguntas, etc. Y en nuestro caso hubo periodos al principio en los que había gente sin hacer nada ya que ellos mismos no se buscaban tarea y que se la buscase otro era un impedimento.

Aquí es donde se aprende: si además de trabajar tienes que buscarle tareas a gente del equipo (ya que hay cosas obvias que podrían estar desarrollándose simultáneamente), hay un problema. Si el problema ocurre una semana se intenta solucionar hablando. Si no se arregla/vuelve a repetirse (basándose en acciones/resultados), la gente afectada debe actuar, no esperar que el mismo intento de solucionar el problema hablando cambie el resultado.

Y por ello, la semana que dejamos la planificación tras ver su inefectividad de manera repetida debida a su incumplimiento por algunos miembros, deberíamos haber intentado buscar una solución, en nuestro caso, hablar con Carlos.

Nuestro equipo activo tardó alrededor de un mes en reaccionar (con Semana Santa por el medio), pensando que el mismo intento de hablar las cosas y las buenas intenciones les haría trabajar menos. Al final lo hablamos con Carlos, pero tarde.

Trabajar demasiado no es bueno para nadie, el estrés es muy perjudicial para la salud. No lo haciamos por la nota, sino por las ganas de realizar un buen juego y aunque creemos que es original le falta por pulir.

Participación en la realización del juego

A continuación se van a repartir 80 puntos que representan la totalidad del juego entre los 8 participantes del grupo. La repartición es un intento de reflejar su aportación al mismo.

Tras hablarlo, el equipo se encontró con dificultades para decidir una repartición exacta asi que se decidió dejarlo en forma de intervalos. Ya no suma 80 pero llegados a ese punto no pensamos que fuera importante.

- Mateos, Diego **35-30**
- Gavilán, Sergio **20-18**
- Cañellas, Lluis **15-14**
- García, Jorge **7-6**
- de los Santos, Mario **5**

- Serna, Álvaro **4**
- Docampo, Marcos **2**
- Suárez, Antonio **2**

Participación extendida (tareas principales)

35-5 - Diego Mateos

Documentacion

- Documento extenso de diseño, paso de 3 a 20 paginas, a exclusión de "Menús e interfaz" y "Minijuegos"
- Documento de planificación (diseño y redacción)
- Éste documento post mortem (cada integrante es libre de completar/concretar lo que quiera en su campo)

Arquitectura

Toda la aquitectura del juego, a grandes rasgos (al principio de la planificación se puede ver específicamente):

- Inclusión, adaptación y mejora significativa del paquete base de `SDLGame`+`GameObject`+`Vector2D`+`Resources`
    - Mejora de `SDLGame`: comentarios, metodos rnd, clase hija game, logControls...
    - Mejora de `GameObject`: comentarios, separación en Activable, Physical, Transformable...
    - Mejora de `Vector2D`: comentarios, definicion de bastantes operadores, bastantes metodos mas...
    - Mejora de `Resources:` comentarios, añadidos colores, spritesheets, shapes, renderers variados, fixed bugs...
- **Maquina de estados**, controlada por mensajes desdes cualquier estado (también es la que dibuja el fondo animado comun a todo).
- **Entity**, objeto que almacena componentes de cualquier tipo y ejecuta su logica
    - addComonent(), delComponent(), delAllComponents(), constructoras que permiten pasar vector de componentes...
    - switchComponent(), mas limpio que del->add y ademas suele ser necesario para conservar el orden entre renderers
    - toggleDebugSingle(), muestra la hitbox, direccion y velocidad utilizando un componente estatico
    - toggleDebugGlobal(), mostrar todas las hitboxes
- **Transformable**, clase itermedia entre Physical y Entity, permite aplicar transformaciones "smooth" a los objetos de manera muy simplificada
    - ResizeToScale(scale, miliseconds), el objeto se reescala poco a poco. Envio de mensaje opcional al acabar

    - MoveToPoint(point, miliseconds), se mueve poco a poco en linea recta y tambien envio opcional al final
    - SmoothMoveToPoint(point, max time), se mueve simulando fuerzas y termina prematuramente en max time. Tambien envio de mensajes
    - static get/setRealPosition(), sirve para mover el mundo fisico en vez de los objetos (simular movimiento de camara)

- **ImageRenderer**, clase padre de los renderers, pide un puntero a un spriteData que lo configura (tambien tiene alpha):

    - fadeToAlpha(alpha, miliseconds), poco a poco cambiar hasta el alpha definido
    - waveAlpha (minAlpha, maxAlpha, miliseconds), el objecto varia su alpha de min a max poco a poco (miliseconds)

```
struct spriteData
{
        Vector2D scale_, offset_;       //on top of object
        bool rotation_;                 //check direction and rotate render
accordingly
        bool drawRealPosition_;
};
```

    - hija `ShapeRenderer`, modula una textura segun un color (cambiable etc)
    - hija `Message_RC`, renderizar facilmente un texto (get/set, text, color, font, etc)
    - hija `Value_RC`, renderiza un valor de un objeto y se updatea constantemente
    - hija `SpriteSheetRenderer`, permite renderizar un frame de un spriteSheet (y cambiarlo, etc)

- **AnimationRenderer**, utilizado para todas las animaciones del juego

    - Configurado con un spriteSheetData (tambien permite modulacion de color y auto start):

```
struct spriteSheetData
{
    Vector2D scale_, offset_; //on top of object
    bool rotation_; //check direction and rotate render accordingly

    size_t
        //textureCols_, textureFils_,   //loaded in resources
        frameStartCol_, frameStartFil_, //if 0,0 -> auto set to min/max
        frameEndCol_, frameEndFil_;     //the first frame is 1,1 not 0,0

    Uint32 fps_; //frames / MS_IN_SECOND

    bool loop_, stay_, //render even when static (start == end, loop ended
or not started, etc)
```

```
        drawRealPosition_ = false;
};
```

- **BoardState**, con todos sus manager y objetos (Muy complejo debido a todo el flujo asincrono)
    - `BeeManager`, todo el comportamiento de las abejas
    - `PlayerManager`, comportamiento de los players (reordenacion smooth asincrona, resalto del actual, puntos que se suman poco a poco, cola de players...)
    - `DicesManager` y squaresManager, no tan complejos como tal, pero si el integrarlos con abejas/players
    - Dinamismo del juego: el board viene por la izquierda y se va por la derecha al principo/fin de la ronda
    - Mensajes en pantalla parpadeando suavemente informativos
    - Conexion con el principio (mensaje starting) y con el final (mensaje ending + ir al podium + despues creditos)
- **Spinner**, ademas de integrarlo con board y el resto del juego
    - Muy extensible, tiene dos vectores y es escribir un id extra a cada uno para añadir un minujuego+arte al random
    - Personajes animados, y titulo, todo configurable
- **Simulador** de partidas con calculo de estadísticas para balanceo -> escritura/lectura de archivos
    - Controles debug para board (junto con logs explicativos)
    - Metodo debug endMinigame que vuelve a board con puestos aleatorio

Comentarios

- El 99% de las variables constantes son constantes y editables desde json
- Practicamente todo el board esta implementad para ser configurable desde json -> rondas, casillas, players, posiciones... Y todo funcionaria acorde
- Nada deja basura, y muchos objetos ni siquiera son punteros ya que son atributos de la clase iniciados en la constructora en lista
- Amplio uso de polimorfismo y mensajes
- Contador de casteos: 0 -> solo para mensajes
- Lo unico que no he usado mucho han sido parametros por referencia constante
- `+` lo que me hay olvidado
- `+` el push de la semana final retocando/fixeando por muchos sitios

2018 - Sergio Gavilán

- Realización del Sumo (con Mario).
- Realización del minijuego de las manzanas.
- Realización del menú final (incluyendo creditos e instrucciones de juego).
- Realización de las pantallas de controles.

- Inclusión de un json parser al proyecto.
- Arte de la pantalla de controles (descripciones,diferentes iconos para cada minijuego).
- Arte de la pantalla de créditos e instrucciones.
- Arte de los iconos de la ruleta.
- Mejora del timer que había y ampliación para poder usarlo con cuenta atrás.
- Arte de la cuenta atrás de lo minijuegos.
- Arte del podium.
- Descripciones de Minijuego de las manzanas y sumo en el GDD.

15-4 - Lluis Cañellas

- Realización del Pong (con Jorge)
- Realización de ambos minijuegos de plataformas + descripcion del gdd
- Realización del minijuegos de esquivar balas + descripcion del gdd
- Añadidos inputs de mando a menu y board
- Realización del podium (versión final)

7-6 - Jorge García

- Realización del Pong (con Lluis) + descripcion del gdd
- Todo el sonido y música (recopilación e inclusión) + añadida a los créditos
- Sistema de soltado de inputs del mando
- Cambios en los controles de los minijuegos para adecuarlo a esto último
- Cambios en los sprites temporales por los definitivos y retoques de los mismos
- Installer y actualizaciones del mismo

5 - Mario de los Santos

- Realización del Sumo (con Sergio)
- Implementación del podium (básico)
- Corrección de los colores de los jugadores y por tanto de algunos sprites

4 - Álvaro Serna

- Componente de controller (no final)
- Realización de la mayoria del arte
  *Fondo animado
  *Personajes (dibujo y animación)
  *Arte de todo el tablero + ruleta + la mayoría de minijuegos
- Seccion de menús e interfaz del gdd

2 - Marcos Docampo

- Idea inicial y documento de diseño inicial
- Presentación 1 + video explicativo
- Web / Presentación 2
- Video final / trailer

2 - Antonio Suárez

- Presentación final
- Prototipo de Spinner, menu y creditos.
- Última revisión del GDD