

StickMotion: Editor de posturas, posiciones y movimientos

*Carmona Varo, Fernando; García García, José Manuel; López Fernandez, David;
Navas Torres, Francisco Javier; Porras Bueno, Javier*

19 de abril de 2010

Resumen

El presente documento se corresponde con el entregable nº 3. El cliente recibirá la descripción informal de la gramática y el nuevo prototipo de la interfaz, más accesible, usable e intuitiva, promoviendo así un uso autosuficiente y mejorando la experiencia del usuario con la misma.

1. Gramática del lenguaje

A continuación se describe la gramática de Sticky que se ha desarrollado:

- Todas las instrucciones del lenguaje llevarán el carácter fin de instrucción (;).
- Los operadores aritméticos reconocidos son: +, -, *, /, ++, --, %, ^, raiz(x).
- El lenguaje es sensible a las mayúsculas y minúsculas.
- Tipado dinámico de datos:
 - Definición de variables: *var nombreVariable*.
 - Destrucción de variables: *sup nombreVariable*.
- Tipos de variables soportadas: booleanas, enteras, reales y cadenas.

- Comentarios: monolínea ('//') y multilínea ('/*' y '*/').
- Asignaciones: símbolo igual ('=').
- Operadores condicionales: '<', '>', '<=', '>=', '==' y '!='.
- Operadores lógicos: 'Y', 'O', 'OX', 'NO', '!'.
- Los bloques de código irán encerrados entre llaves ('{' y '}').
- Construcción de bucles:
 - Bucle for:
 - para* (inicio; fin; incremento) { sentencias }
 - para* (inicio; fin; incremento) sentencia
 - Bucle while:
 - mientras* (expresión o variable) { sentencias }
 - mientras* (expresión o variable) sentencia
- Estructuras condicionales:
 - if:
 - si* (condición) { sentencias }
 - si* (condición)
 - if-else:
 - si* (condición) { sentencias } sino { sentencias }
 - si* (condición) sentencia sino { sentencias }
 - si* (condición) { sentencias } sino sentencia
 - si* (condición) sentencia sino sentencia
 - switch:
 - opcion* (expresión o variable) {
 - caso* expresión: { sentencias } *fincaso*;
 - ...
 - defecto*: { sentencias } *fincaso*;
 - }
- Movimientos del Stickman:
 - *girar* CABEZA x y z.
 - *girar* [BRAZO | PIERNA] [IZQ | DER] x y z.
 - *girar* STICKMAN x y z.
 - *flexionar* [BRAZO | PIERNA] [IZQ | DER] angulo.

- Inclusión de código fuente: *incluir('rutaAlFichero.stk');*
- Definición de funciones: *def nombreFunción([Parámetro/s]) { ... }.*

2. Interfaz de la aplicación

Con respecto al prototipo que aparece en el entregable número 2 vemos que algunos elementos permanecen y otros han cambiado.

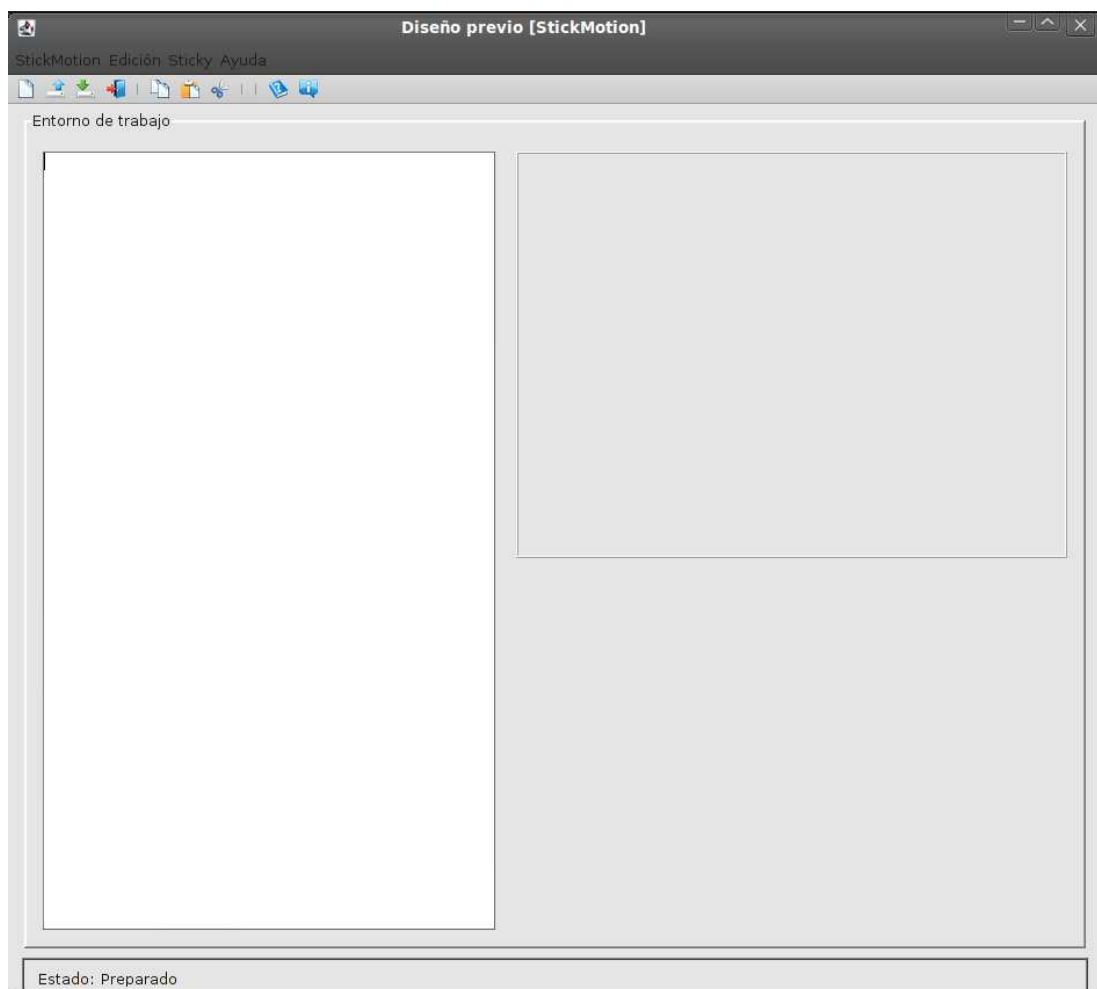


Figura 1: Nuevo prototipo de la interfaz de Stickman.

A simple vista se observa que el área del editor de texto se ha aumenta-

do para permitir la visualización de una mayor cantidad de código Sticky, lo que evitará al usuario el tener que hacer un uso continuado y/o hasta excesivo de las barras de desplazamiento. A su parte derecha se ha desplazado ahora la ventana de animación, quedando aún reservada la parte inferior para los propósitos que aún especifique el cliente, como por ejemplo, las acciones más comunes.

Tal y como se puede apreciar en la figura 2 se muestra un detalle de una nueva característica: la barra de herramientas. Esta nueva función permite que ciertas acciones consideradas como más comunes se puedan llevar a cabo reduciendo el número de clicks necesarios por el usuario, facilitando de esta manera su trabajo con la aplicación. Asimismo, cada icono incorpora una breve ayuda sobre el objetivo del mismo, muy útil para los usuarios que comienzan a manejar *StickMotion*.



Figura 2: Áreas de la interfaz de la aplicación.

Los iconos que componen dicha barra de herramientas son:

- Ficheros fuente (.stk): nuevo documento, abrir/cargar fichero existente y guardar/salvar el fichero actualmente abierto.
- Salir de la aplicación.
- Edición de texto: copiar, pegar y cortar.
- Ayuda.

Las barras de menú y estado permanecen iguales, sin ninguna modificación con respecto al anterior prototipo (entregale nº 2).