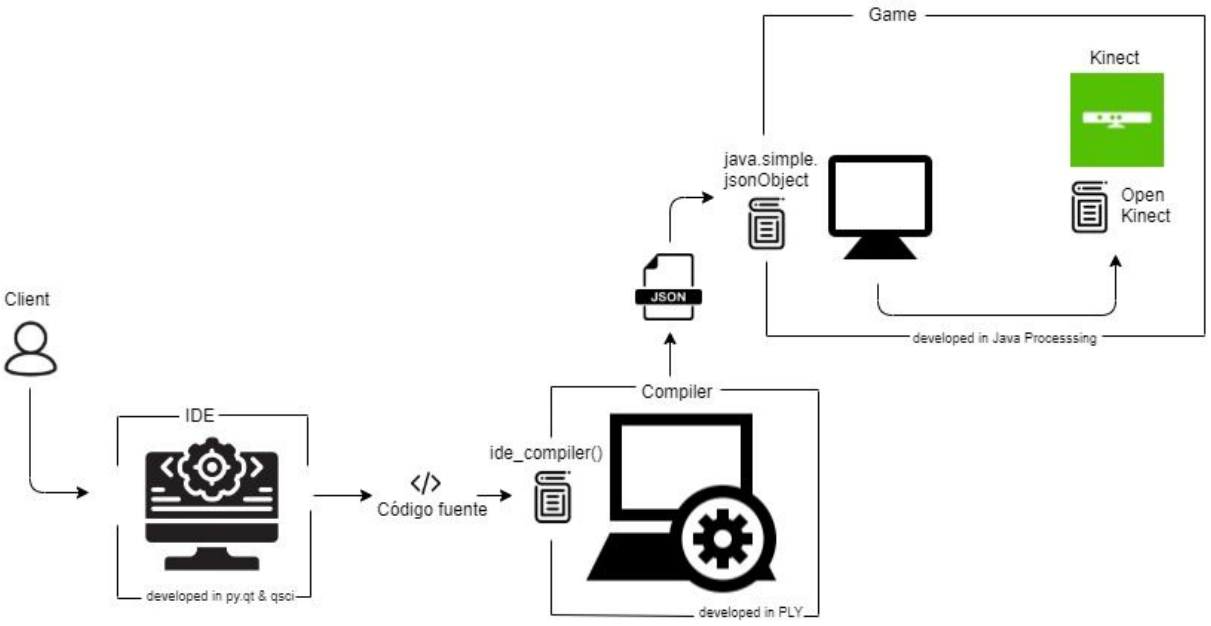
Documentación y aspectos operativos

Instituto Tecnológico de Costa Rica
Lenguajes, Compiladores e Intérpretes | CE3104
Prof. Marco Hernandez
Esteban Alvarado - Erick Barrantes - Sahid Rojas - Jessica Espinoza - María José Zamora

---

1. Diagrama de Arquitectura



2.Alternativas Consideradas

Hardware

- *Acelerómetros:* Un acelerómetro es un dispositivo que mide la vibración o la aceleración del movimiento de una estructura. Esto permite obtener la velocidad de los movimientos realizados por el niño con sus extremidades.

- *Sensores de profundidad:* Estos proveen la distancia que hay desde los mismos hasta a algún objeto posicionado cerca.

- *Cámara digital y uso de redes neuronales:* Mediante estos l es una cámara fotográfica que, en vez de captar y almacenar fotografías en película química como las cámaras de película fotográfica, recurre a la fotografía digital para generar y almacenar

imágenes y estas imagenes seran procesadas mediante el uso de redes neuronales para saber la posición de cada una de las extremidades del niño y así hacer un seguimiento en tiempo real de sus movimientos

Seleccionada: Kinect

El sensor de profundidad funciona a partir de dos elementos incluidos en el Kinect: un proyector de luz infrarroja y un sensor de este tipo de luz. El primero proyecta una matriz de rayos de luz infrarroja sobre la habitación, estos rebotan en los objetos y son capturados por el sensor infrarrojo. El Kinect nos proporciona valores de profundidad que siendo procesados de la manera correcta se puede modelar el mundo en 3D sabiendo así no solo cambio en los ejes de dos dimensiones sino también contemplando así los movimientos en 3 dimensiones por lo cual la información que manejamos es mucho más precisa.
Además de eso también aprovechando que las únicas dos conexiones que posee el kinect son usb y el cable a corriente nos ahorramos muchos posibles errores por mal cableado o conexiones como también generando una estandarización ya que las únicas necesidades a nivel de hardware son el sensor y una computadora creando que las personas que podrían utilizar este dispositivo sea mas y a su vez las personas que se verían beneficiadas sean mas tambien.

Software:

- *lexer-and-parser:* Esta es una librería desarrollada en python encontrada en un repositorio de GitHub, que facilita al programador la implementación de un compilador con análisis léxico y sintáctico. Sin embargo, la documentación era muy pobre y al no ser muy utilizada, no contaba con muchos ejemplos, por lo que fue descartada para este proyecto.

- *rply:* Esta librería para python es mucho más elaborada y completa que la anterior, además de que simplifica el trabajo del programador de gran manera. Sin embargo, nuevamente la documentación de esta librería es muy escasa. A pesar de ser más usada y si existen muchos ejemplos con rply, se tomó la decisión de seguir investigando por mejores opciones

Seleccionada: ply (Python Lex-Yacc)

Ply fue la librería escogida y utilizada en el proyecto, ya que cuenta con muchas comodidades, donde el programador solo se encarga de declarar cuáles serán los tokens en el análisis léxico y cuál debe de ser la gramática por cumplir para el análisis sintáctico. Además de aligerar el trabajo del programador, cuenta con una documentación oficial muy completa con ejemplos codificados.

Una vez que escogimos esta librería para el desarrollo del compilador, se estableció los tokens léxicos que tendrá el lenguaje, cumpliendo la especificación del proyecto. Cabe recalcar que el lenguaje desarrollado es *case sensitive*, por lo que los tokens a continuación se deben cumplir como están presentados para no caer en errores léxicos.

Tokens para operadores

Nombre de token	Símbolo
PLUS	+
MINUS	-
MULT	*
DIVIDE	/
EQUAL	=
LPAREN	(
RPAREN	)
SEMICOLON	;
LBRACE	{
RBRACE	}
LBRACKET	[
RBRACKET	]
COMMA	,

Tokens para palabras reservadas, como funciones y ciclos

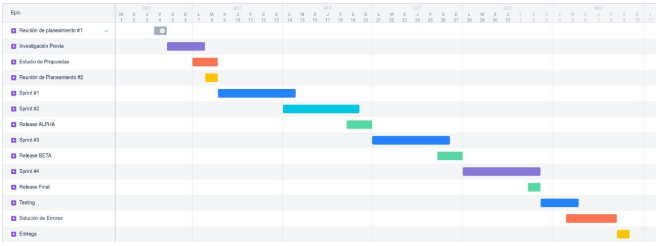
Nombre de token	**Símbolo**
MAIN	main
FORASSIGNWORD	forAssignWord
FOR	for
DOW	dow
ENDDO	enddo
FOREND	forend
DO	do

TIMES	times
USING	using
BEGIN	begin
END	end
INT	int
STR	str
BALLOON	balloon
RANDOM	random
TELAARANA	telaArana
ASSIGNWORD	assignWord
OBJECT	object
INC	inc
DEC	dec

Por último, algunos tokens fueron especificados con expresiones regulares:

Nombre de token	Expresión regular
GAME	game[0-4]
ID	[a-z]([a-zA-Z0-9&@_-])* Si y solo si, el ID ingresado que cumple con esta regla, no es una palabra reservada.
NUMBER	[0-9]+
STRING	“[a-zA-Z | 0-9]*”
NEWLINE	\n+

3. Actividades realizadas por estudiante



Como en todos los proyectos es necesario organizarse para cumplir los objetivos y obtener un resultado de calidad. Es por esto que desde el comienzo nos hemos dedicado a planear el desarrollo del proyecto, tratando de implementar metodologías de desarrollo ágil como scrum (Desarrolladas en herramientas digitales como *Trello o Jira*). Al finalizar el plazo de tiempo, cada uno de los integrantes del equipo a realizado un resumen de su trabajo en una bitácora de trabajo. ***Esta bitácora se adjunta en un archivo pdf aparte para que se pueda apreciar mejor la información del documento.***

4. Problemas

Durante el proceso de desarrollo del proyecto **Fun Skills™** nos encontramos con inconvenientes para llevar a cabo tareas y alcanzar objetivos. Por esto hemos recopilado todos aquellos problemas que se tuvieron junto con las soluciones que nos ayudaron a seguir adelante. Es necesario decir que no todos los problemas que se tuvo son iguales ya que algunos sí lograron ser solucionados, a estos los llamamos *problemas encontrados*. A aquellos problemas que no les encontramos solución los llamamos *problemas conocidos*.

A continuación se presentan los problemas encontrados y conocidos del producto en su *versión de prueba* 3.0.

Tabla 1. Problemas Encontrados en el desarrollo de Fun-Skills

Desarrollador	Problema	Solución
Esteban	No se lograba obtener el código escrito en el IDE mediante una variable.	Se creó un *archivo temporal* en el cual se escriben todos los cambios y a partir de este se pueden efectuar todas las funcionalidades del IDE. Como RUN y SAVE.
Erick y Jessica	No se lograba detener el análisis cuando se encontraba un error léxico o sintáctico.	Mediante el *raise* de Python, se levantan excepciones cuando se encuentra un error en el análisis, para manejarlo en un try-except
Esteban	El proyecto no permitía una independencia del equipo en cuanto a paths del sistema.	Utilizando la biblioteca sys y os de Python se logra generar un script que obtiene el directorio del proyecto en cualquier equipo permitiendo el uso de *paths relativos*.
Erick y Jessica	No se lograba enviar al IDE mediante una variable un mensaje de error del compilador.	Se creó un archivo temporal *error_log.txt* donde se escribe el error encontrado a la hora de hacer el análisis. A la hora de tratar este error, se lee el archivo y se envía este texto al IDE

Esteban	No se podían unir sketches de Processing en carpetas distintas sin un JAR.	Al final migramos los códigos a clases para poder instanciarlos dentro de un archivo principal llamado gameSuite.

Tabla 2. Problemas Conocidos en el desarrollo de Fun-Skills

Desarrollador	Problema	Solución
Erick y Jessica	Error semántico: No se valida para los valores máximos de fila y columna de la telaraña,	No se encontró una solución en el tiempo de desarrollo.
Erick y Jessica	Error semántico: No se valida que la asignación a un valor de un array cumpla con el tipo especificado en su declaración.	No se encontró una solución en el tiempo de desarrollo.
Erick y Jessica	Error al generar código: En el juego de colores, no se es asociado un score con cada color.	No se encontró una solución en el tiempo de desarrollo.

5. Conclusiones y Recomendaciones del proyecto

En base al proyecto realizado durante este tiempo se puede concluir que el uso el kinect Facilitó el desarrollo de modelar el mundo ya que diferencia de las cámaras digitales que lo que captan es la luz el Kinect capta los niveles de profundidad y en base a eso se puede construir un modelo mucho más completo y real generando así una gama amplia de posibilidades.

En cuanto al editor de código implementado *"Fun Skills - Playground"* podemos resaltar la gran ayuda que brindó la biblioteca pyQt y su extensión Qscintilla que permitieron un desarrollo del resaltador de sintaxis sencillo y bien modulado, dejando muy en claro que estas bibliotecas son superiores a tkinter. Se recomienda para el desarrollo de cualquier tipo de Editor de código utilizar Qsci, pues ofrece muchas más cosas de las utilizadas en este proyecto.

Para el desarrollo del compilador, se necesitó de una investigación previa para la elección final de ply como librería a utilizar. Después de múltiples pruebas codificadas, se inició el desarrollo del analizador léxico y sintáctico. Para esto, fue necesario aplicar el conocimiento adquirido en el curso, como la creación de expresiones regulares y de gramática libre de contexto para que los análisis sean correctos.

Como ply no cuenta con la creación de un AST, fue necesario desarrollar una clase para un árbol n-ario y armar este en el análisis sintáctico sin el uso de ninguna librería. Siendo esto un gran reto, se diseñó el AST esperado, donde la generación de código y análisis semántico depende de esta.

Como recomendación, luego de la investigación realizada concluimos que la librería ply, la cual es una adaptación de Lex y Yacc para python, es una de las opciones más simples y mejores documentadas para el desarrollo de compiladores e intérpretes. Sin embargo, se debe de tomar en cuenta que este no genera un AST ni se encarga del análisis semántico.

Como el proyecto está diseñado para niños, durante el desarrollo de la interfaz de usuario en los juegos concluímos que los botones grandes con figuras geométricas básicas como cuadrados y círculos de colores brillantes y diversos atraen más la atención y aumentan el entusiasmo de un niño que realizará su terapia motivado por los juegos y el impacto visual. Como recomendación final, processing resultó ser una gran herramienta pues el desarrollo de juegos en este entorno se vuelve increíblemente cómodo, versátil y eficiente, y también cabe mencionar que posee render 3D que no utilizamos en este proyecto pero que aportaría mucho a la experiencia de juego.

6. Bibliografía

Beazley, D. M. PLY (Python Lex-Yacc). Retrieved 08 October 2019, from https://www.dabeaz.com/ply/ply.html#ply_nn24.

Bodnar, J. (2019). Menus and toolbars in PyQt5 - QMainWindow, QAction, QApplication. Retrieved 17 November 2019, from http://zetcode.com/gui/pyqt5/menustoolbars/

How to use JSON with Python. (2019). Retrieved 17 November 2019, from https://developer.rhino3d.com/guides/rhinopython/python-xml-json/

loadJSONObject() \ Language (API) \ Processing 3+. (2019). Retrieved 17 November 2019, from https://processing.org/reference/loadJSONObject_.html

PyQt/Python syntax highlighting - Python Wiki. (2019). Retrieved 17 November 2019, from https://wiki.python.org/moin/PyQt/Python%20syntax%20highlighting