

Grab them with ai: aprender programación jugando

Dr. German Antonio Montejano¹

Dr. Mario Marcelo Berón²

Lic. Mariano Gabriel Luzza Bonilla³

¹Email: gmonte@unsl.edu.ar

²Email: mberon@unsl.edu.ar

³Email: mluzza@unsl.edu.ar

Universidad Nacional de San Luis

RESUMEN

En la actualidad existen numerosos problemas, que si bien pueden ser solucionados con herramientas de propósito general, es más apropiado abordarlos con aplicaciones específicas para ese dominio. En este contexto se encuentran los Lenguajes Específicos del Dominio (LED).

Un LED es un conjunto reducido de construcciones y operaciones que brindan una mayor expresividad y optimización para un dominio particular.

Un área de especial aplicación para los LED es en el ámbito de la enseñanza en programación, ya que posibilitan abstraerse de los problemas particulares de los Lenguajes de Propósito General (LPG) para centrarse en el tema particular que se desea enseñar. Una primera aproximación a los LED son las APIs, siendo la base para una posible construcción de un LED. En este trabajo se presenta "Grab them with AI" una herramienta que provee una API para desarrollar inteligencias artificiales para competir en un divertido juego y así practicar conceptos de programación.

Palabras clave: LED, Lenguaje Específico del Dominio, Enseñanza de Programación, Generador de Aplicaciones, Videojuego.

INTRODUCCIÓN

Con el paso del tiempo, es más común ver computadoras a nuestro alrededor. Ya sea en la forma tradicional de computadoras personales, o en otros dispositivos más recientes como celulares inteligentes, tabletas, electrodomésticos, vehículos, etc. Con ello, también se hace más notoria la necesidad de que las personas aprendan a programar, principalmente porque todos estos dispositivos necesitan programas, pero también porque ayuda a pensar de una forma algorítmica, es decir más ordenada. No parece muy radical pensar que en el futuro cercano, tener conocimientos de programación sea tan importante como lo es terminar los estudios secundarios en la actualidad.

En pos de este objetivo, los ingenieros de software han buscado, a través de arduas investigaciones, como hacer más sencilla la tarea de enseñar a programar [1]. Como resultado de las tareas antes mencionadas, fue posible detectar dos principales ramas de estudio. Por un lado, resulta interesante conocer cuáles son las causas de que sea tan difícil para el alumno aprender los conceptos básicos de programación. En este contexto, se observa que pocos alumnos asocian el uso de computadoras como herramientas programables para resolver problemas. Esto

se debe al uso común que le dan los jóvenes a las computadoras, como navegar en internet, chatear o escribir un documento, entre otras tareas. Otra arista de este problema es que en materias de informática del colegio secundario, generalmente se limita a la enseñanza de ofimática [2, 3], reduciendo el horizonte que tienen los alumnos en las ciencias de la computación. Por otro lado, se puede evidenciar que con herramientas y lenguajes más didácticos y cuyo principal objetivo es facilitar el aprendizaje de los conceptos de programación, se logran mejores resultados [4]. Ejemplos de este tipo de lenguajes son Logo [5], Scratch [6], Proyecto Hoshimi [7], entre otros tantos. Claramente a pesar de los esfuerzos realizados por la comunidad científica, el problema está siendo abordado pero no se ha resuelto del todo. Un problema para añadir, es que muchas personas no aprenden fácilmente los métodos, técnicas y uso de herramientas de ingeniería de software y se dan por vencidos en forma temprana. Parte de la dificultad se presenta en que los lenguajes de programación son muchas veces complejos por demás haciendo que los alumnos tengan problemas para asimilar los conceptos [8]. Otra debilidad se observa a la hora de mostrar resultados al usuario. Los alumnos invierten gran cantidad de tiempo en la creación del programa, que al ser ejecutado, resulta ser un monótono programa de texto, en el caso de que al menos se presente un lenguaje que puede ser utilizado en una computadora, ya que a veces las prácticas se limitan a “lápiz y papel”. Ciertamente la programación y la lógica no se van a convertir en más sencillos, por lo menos no en este trabajo. Lo que sí es posible cambiar y es el foco de este artículo, es que los lenguajes y sus herramientas de trabajo sean más atractivos de utilizar, poniendo un incentivo extra al de aprender a programar. Es razonable pensar que una mayor cantidad de herramientas contribuye a que diferentes perfiles y contextos del mismo problema sean abordados, brindando así diversas soluciones para la enseñanza de programación. Por ello, en este trabajo se presenta “Grab them with AI”, un juego competitivo para practicar de forma amena conceptos de programación.

El artículo está organizado como sigue. En primera instancia se introduce el concepto de Lenguaje Específico del Dominio. Luego se presenta “Grab them by the Eyes”, un juego desarrollado por Terry Cavanagh, que se usó como base para el desarrollo de este trabajo. Después se describe “Grab them with AI”, una herramienta que presenta una API y front-end para el desarrollo de inteligencias artificiales (IA) para el juego, basado en el original. La API supone el primer paso para la futura implementación de un LED. Como paso siguiente se expone un caso de estudio en donde se manifiesta el uso de la API para resolver problemas típicos de programación. Finalmente, se presentan las conclusiones y el trabajo futuro de éste artículo.

LENGUAJE ESPECÍFICO DEL DOMINIO

Un Lenguaje Específico del Dominio (LED) es un conjunto reducido de construcciones y operaciones que brindan una mayor expresividad y optimización para un dominio particular. Según Hudak [9] un LED es la “última abstracción”, la cual captura precisamente la semántica de un dominio de aplicación.

Algunos LED bien conocidos incluyen SQL y expresiones regulares entre otros. Claramente, cada uno es mejor que un lenguaje de propósito general para representar operaciones sobre base de datos y cadenas respectivamente. Pero no sucede lo mismo cuando se intenta describir soluciones fuera de su dominio. Algunas industrias poseen también sus propios LED. Por ejemplo, en telecomunicaciones, los lenguajes de descripción de llamadas son ampliamente utilizados para especificar la secuencia de estados en una llamada telefónica, y en la industria de viajes, generalmente se necesita un lenguaje para describir reservas de vuelo. Otras áreas donde también se usan LED incluyen el plan de rutas de navegación de una web, diagramas de conexión de componentes electrónicos, un árbol genealógico, etc.

Cuando se diseña un LED, se deben especificar claramente los conceptos y relaciones relevantes para el dominio al cual dicho lenguaje va a estar destinado, como por ejemplo un árbol genealógico. Para armar un árbol

genealógico es necesario representar las relaciones familiares entre distintas personas. De aquí se puede rescatar como conceptos importantes “Persona” y “Parentesco”.

Los usuarios de un LED crean modelos los cuales son las instancias del LED [10]. Es común encontrar LEDs cuyo uso es generar código de programa en otro lenguaje o un ejecutable, pero también son utilizados para generar otros artefactos como un esquema de visualización para ciertos datos. Cuando se define un LED, se pueden especificar plantillas que lean un modelo del LED y generen ejecutables, fuentes de otros lenguajes, archivos de texto u otros artefactos. Siguiendo el ejemplo del árbol genealógico, se podría tener una plantilla que tome varias relaciones de parentesco simples como padre, hijo y hermano y genere otras relaciones deducibles como abuelo, sobrino y primo. Otro caso donde sería útil, podría ser para un equipo de desarrollo de sitios web, que cuente con un LED para generar la navegabilidad de las diferentes páginas y que cree tanto un esqueleto del sitio como una herramienta que verifique dicha navegabilidad. Generalmente, los LED son creados cuando un grupo de usuarios (no necesariamente desarrolladores) tienen que generar código similar para varios productos. Esto es debido a la ventaja que ofrecen los LED en abstracción y focalización en un dominio, sumado a que el código producido es más confiable gracias a la automatización y refactorización de las tareas.

Una primera aproximación a los LED son las API (application programming interface o interfaz de programación de aplicaciones en español) construidas sobre un LPG. Muchas veces las API son el primer paso hacia el desarrollo de un LED. También muchos de esos desarrollos de LED quedan solamente en ese primer paso, cuando se evalúa la relación costo beneficio [11]. Luego en la transición de la API al LED, un posible patrón de implementación es un lenguaje que se traduzca en invocaciones a la API ya desarrollada.

En este artículo se presenta la construcción de una API centrada en el desarrollo de IA para el juego “Grab them by the Eyes” y así practicar conceptos de programación me-

dante un videojuego. La API se describe detalladamente más adelante.

GRAB THEM BY THE EYES

Grab them by the Eyes es un juego basado en tecnología Flash desarrollado por Terry Cavanagh y publicado en febrero de 2015. El escenario del juego presenta un Jay el propietario de un carrito de hamburguesas, a quien de repente se le presentan dos jóvenes con otro carrito pero que tiene una novedad, un cartel luminoso para atraer a los clientes. Los dueños de “Hamburguesas Sucias” (o Filthy Burgers en inglés) desafían a Jay retándolo a que no tema un poco de competencia y le hacen la siguiente apuesta: el vendedor de hamburguesas que más clientes tenga durante la semana, se queda con la zona de venta. Jay cree que para competir, también debe armar letreros luminosos y es allí donde entra el papel del jugador.

Una partida del juego se divide en seis días. Cada día se repite una rutina que consiste en tres etapas:

1. Comprar partes de carteles.
2. Armar carteles.
3. Ver el resultado de los carteles.

Durante la primera etapa ambos vendedores se turnan uno a otro para comprar componentes de carteles. Además, cada día se alterna el vendedor que comienza la compra para dar equilibrio a esta etapa. La tienda de carteles presenta cinco partes de carteles en vidriera de diferentes tipos, puntajes y precios. Los tipos son: (i) mensaje; (ii) color; (iii) borde; (iv) efecto; y (v) marco. Los puntajes son un número natural que representa la cantidad de clientes que atraerá. El precio está determinado por la posición en la tienda, siendo el costo del primer cartel \$10, el segundo \$20 y así siguiendo. Cuando un vendedor compra una parte de cartel, se quita de la tienda y las demás se corren para ocupar la posición desierta, bajando el costo de esos componentes. Por ejemplo, sean A, B, C, D y E partes de carteles en las posiciones 1 a la 5, si el jugador compra la parte C a \$30, luego D y E se desplazarán una posición, costando ahora \$30 y \$40 respectivamente. Además, la tienda suele tener más de cinco partes de

carteles, siendo las sobrantes una reserva que no se muestra ni se puede comprar, pero que van apareciendo a medida que se compran las que están en vidriera. Cada turno, un jugador puede comprar un único componente. Los turnos se suceden hasta que ambos jugadores se queden sin dinero para comprar o se acaben los componentes.

En la segunda etapa los vendedores arman sus carteles. Los carteles están divididos en marcos, que al principio de la partida son dos, pero se pueden comprar más en la tienda. Cada marco puede tener a lo sumo uno de cada tipo de componente restante (mensaje, color, borde y efecto). La suma de los puntos de los componentes usados es el puntaje total del cartel.

Al final, en la tercera etapa se computan los puntos (que representan clientes) en cada vendedor. Además los componentes utilizados restan un punto hasta un mínimo de uno, que representa la pérdida de actualidad. Cuando transcurren los seis días, se suman todos los días y se define el ganador.

GRAB THEM WITH AI

El desarrollo original de Terry Cavanagh es un juego donde un vendedor (Jay) lo dirige en tiempo real el jugador y el otro es una inteligencia artificial (IA) fija. En este trabajo de investigación, se replicó el juego, pero donde ambos jugadores son IA, que se deben implementar a partir de una API que incluye el nuevo juego desarrollado. Así surge "Grab them with AI" que toma prestado parte del nombre original y lo mezcla con el nuevo agregado, que representa el desarrollo de IA.

COMPOSICIÓN DE LA API

La única forma de jugar a "Grab them with AI" es a través de una inteligencia artificial que debe ser desarrollada usando la API provista por el juego. La API consta de las siguientes clases:

- Carta: representa cada parte de cartel. Tiene propiedades que posibilitan conocer el tipo y el puntaje.
- Cartel: se utiliza para realizar la composición de las partes.

- Jugador: es una clase abstracta que debe implementar quien desarrolla una IA.
- Partida: es el motor de la partida y presenta métodos públicos para que el jugador interactúe con el juego.

En la figura 1 se puede ver un diagrama de clases que resume la API. A continuación se describen las clases con más detalle.

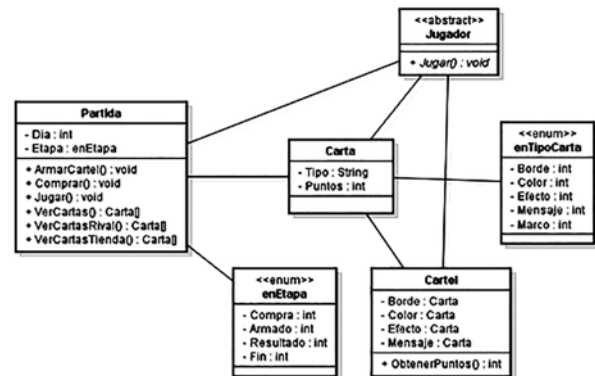


Figura 1 – Diagrama de clases de la API

La clase Carta es la parte más atómica de la API. Cada instancia de Carta representa una parte de cartel. La denominación de carta surge del aspecto visual que tiene en el juego, ya que al comprar las partes del cartel en la tienda, tienen la abstracción similar a la de una carta. La clase Carta tiene dos propiedades:

- Puntos: una propiedad de tipo entero que devuelve los puntos que otorga la parte.
- Tipo: una propiedad de tipo enumerado que devuelve el tipo de parte. Los tipos posibles son: Mensaje, Borde, Color, Efecto y Marco.

Cabe destacar que el tipo Marco es especial y sirve para poder armar carteles adicionales. Como se menciona en la sección anterior, el jugador inicialmente tiene la capacidad de armar dos carteles, limitación que se puede ampliar adquiriendo nuevos marcos.

Siguiendo en importancia, la siguiente clase es Cartel. Cada instancia de Cartel está compuesta por varios objetos de tipo Carta. La única limitación semántica es que no puede haber más de una carta del mismo tipo. La clase Cartel, además de exponer miembros que devuelven sus partes en forma individual y de colección, presenta una propiedad que totaliza los puntos de sus partes.

La clase Jugador representa a un jugador en la partida. Es abstracta porque justamente la implementación la especifica el usuario de la API. Presenta un único método abstracto denominado "Jugar". Es en este método donde el jugador implementa la estrategia de su inteligencia artificial. Dentro del método, recibe como parámetro un objeto del tipo "Partida" al que podrá enviar mensajes para conocer el estado del juego o interactuar con el mismo. Normalmente, este método estará dividido en porciones de código o invocaciones a métodos propios, que se encargarán de cada etapa de la partida.

La clase Partida es el núcleo de la API. Como su nombre lo indica, representa una partida del juego. Expone métodos informativos para que los jugadores conozcan el estado de la partida, como las cartas que están a la venta o la mano del otro jugador. Esta clase es la encargada de hacer transcurrir la partida a medida que los jugadores realizan sus acciones.

INTERFAZ GRÁFICA

"Grab them with AI" presenta un interfaz gráfica sencilla, que permite especificar los ensamblados (dll) que contienen las IA que van a competir y ver el progreso de la partida paso a paso. Al iniciar el juego, se instancia una partida con los jugadores especificados y a través de la interfaz provista, el usuario puede avanzar paso a paso la partida y ver las jugadas de las IAs. En la figura 2, se puede apreciar la ventana completa de la herramienta. En la parte superior de la ventana se encuentra una interfaz que permite especificar los jugadores, a través de la búsqueda y selección del ensamblado. El centro de la ventana contiene listado para ver las cartas de la tienda y su reserva, las cartas de ambos jugadores, los puntos logrados en cada día por ambos y su totalidad, y los carteles armados por ambos jugadores. Finalmente, con el botón de la parte inferior de la ventana, se avanza a través de las etapas y días del juego. A medida que transcurren estas etapas, se van mostrando las acciones de los jugadores en la interfaz descrita anteriormente. En la

figura 3, se muestran las cartas compradas (3.a) y jugadas (3.b) por las IAs en un día de una partida de ejemplo.

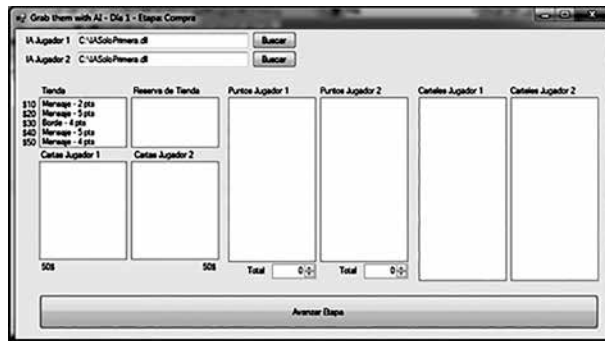


Figura 2 – Pantalla principal del juego

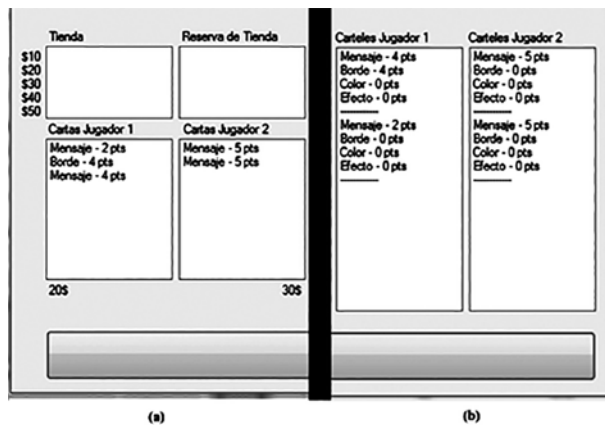


Figura 3 – (a) Cartas compradas por IA. (b) Carteles armados por IA.

NOTAS SOBRE LA API

Con lo mencionado en los párrafos anteriores, el alumno y el docente pueden utilizar este juego para practicar conceptos de programación en un entorno más entretenido que el que presentan los habituales prácticos de clase. Lo más evidente, resulta utilizarlo para escenarios avanzados como inteligencia artificial, puesto que es el principal tópico del juego. Sin embargo, por la forma de trabajo que presenta la API, el juego puede ser útil también para tópicos más simples como herencia y colecciones. Otra característica que resulta especialmente atrayente, es el factor que presenta la competencia entre los jugadores. A diferencia del original, "Grab them with AI" es un juego para dos jugadores, donde ambos jugadores son IA desarrolladas con la API del juego.

CASO DE ESTUDIO

Un desarrollo básico de una IA en el juego es comprar siempre la parte más barata, siempre y cuando alcance el dinero. Luego en la fase de armado, crear un cartel básico eligiendo una parte de cada tipo. Esto implica una estrategia muy sencilla que implica comprar siempre la primera carta, pero corroborando que haya al menos una y que alcance el dinero. Luego la forma más simple de armar un cartel es revisar la mano y colocar

la primer parte de cada tipo. El procedimiento consiste en los siguientes pasos:

1. Verificar que alcanza para comprar la primera carta y ver que exista al menos una carta en la tienda
 - a. Comprar la carta en caso afirmativo.
2. Revisar la mano de cartas disponibles
 - a. Si el tipo de la carta revisada aún no está en el cartel
 - i. Ocupar ese tipo de carta con la carta dada.
3. Jugar el cartel.

```
public class IABasica : Jugador{
  public override void Jugar(Partida p){
    switch (p.Etapa){
      case enEtapa.Compra:
        if(p.GetDinero(this) >= 10 && p.VerCartasTienda().Lenght > 0)
          p.ComprarCarta(this, 1);
        break;
      case enEtapa.Armado: //código para armar el cartel
        Cartel cartel = new Cartel();
        foreach(Carta c in p.VerCartas(this)) {
          if(c.Tipo == enTipoCarta.Borde && cartel.Borde == null)
            cartel.Borde = c;
          if(c.Tipo == enTipoCarta.Color && cartel.Color == null)
            cartel.Color = c;
          if(c.Tipo == enTipoCarta.Efecto && cartel.Efecto == null)
            cartel.Efecto = c;
          if(c.Tipo == enTipoCarta.Mensaje && cartel.Mensaje == null)
            cartel.Mensaje = c;
        }
        p.ArmarCartel(this, cartel);
        break;
      case enEtapa.Resultado: //código para resultado
        break;
      case enEtapa.Fin: //código para fin del juego
        break;
    } //fin switch
  } //fin Jugar
} //fin class
```

Figura 4 – Parte de una IA que resuelve el problema dado

Los pasos descriptos anteriormente, implementados en una programa utilizando la API provista, se muestra en la figura 4. Si bien el código mostrado es una estrategia válida, carece de buen criterio tanto para la compra de las cartas como para el armado del cartel. Cabe recalcar igualmente que es un ejercicio introductorio al tema, sencillo y de rápida resolución. Una vez completado el programa,

se compila y la dll generada se utiliza en el juego junto a otra IA (que puede ser la misma) para que compitan y ver el desempeño de la estrategia.

CONCLUSIONES

En este artículo se presentó “Grab them with AI”, un juego que provee una API y un entorno de ejecución para el desarrollo de

IA. El juego pretende ser otra alternativa a la hora de enseñar programación a estudiantes, brindando un tema simple, entretenido y desafiante. Para alcanzar este objetivo, "Grab them with AI" implementa un juego entretenido, basado en el desarrollo original de Terry Cavanagh "Grab them by the Eyes". La diferencia entre el original y el propuesto, es que en este último, para jugar, se deben desarrollar IA con la API provista. De esta forma, los estudiantes se relacionan con conceptos de programación como manejo de colecciones y desarrollo de IA, pero con un atractivo especial como lo que representa un juego. Además de entretenido, el juego presenta reglas suficientemente simples como para entenderlas rápidamente, pero también complejas a la hora de mejorar las estrategias.

Los trabajos futuros están orientado en tres direcciones principales. La primera consiste en la elaboración de diferentes tests que permitan medir las mejoras que se producen en la enseñanza de la programación cuando se utiliza "Grab them with AI" para dicha tarea. La segunda, se centra en tanto mejorar el entorno gráfico y la API creada, como así también traducirla a otros lenguajes de programación como Java o Python. También como parte de esta dirección, se busca crear un LED y un ambiente de desarrollo propio para el juego, así independizar el aprendizaje de los conceptos, del lenguaje de programación. Por último, también se pretende aplicar la estrategia usada en la creación de "Grab them with AI", en nuevas herramientas para la enseñanza de conceptos de programación. De esta forma presentar más alternativas para los mismos conceptos o para nuevos conceptos.

REFERENCIAS BIBLIOGRÁFICAS

- [1] GEORGANTAKI S., RETALIS S. (2007). *Using educational tools for teaching object oriented design and programming*. Journal of Information Technology Impact. p 111-130.
- [2] CANALETA X., SANCHEZ F., JACOB I., VELÁZQUEZ A., MARQUES M. (2014) *Declaración AENUI-CODDII por la inclusión de*

asignaturas específicas de ciencia y tecnología informática en los estudios básicos de la enseñanza secundaria y bachillerato, Actas de las XX JENUI, Oviedo. ISBN: 978-84-697-0774-6 p 229-236.

[3] RIESCO M., DÍAZ FONDÓN M., ÁLVAREZ GUTIÉRREZ D., LÓPEZ PÉREZ B., CERNUDA DEL RÍO A., FUENTE J. (2014). *Informática: materia esencial en la educación obligatoria del siglo XXI*. ReVisión, 7(3).

[4] SOVIC A., JAGUST T., SERSIC D. (2014) *How to teach basic university-level programming concepts to first graders?* Integrated STEM Education Conference (ISEC). p.1-6.

[5] SEYMOUR P. (1980) *Mindstorms: children, computers, and powerful ideas*. Basic Books, Inc, New York, NY, USA.

[6] MALONEY J., RESNICK M., RUSK N., SILVERMAN B., EASTMOND E. (2010) *The scratch programming language and environment*. ACM Transactions on Computing Education, 10(4):1-15.

[7] LUZZA M., BERON M., HENRIQUES P. (2012) *PH-Helper - a Syntax-Directed Editor for Hoshimi Programming Language, HL*. 1st Symposium on Languages, Applications and Technologies. Braga, Portugal.

[8] KÖLLING M. (1999) *The problem of teaching object-oriented programming*. Journal of Object Oriented Programming, 11(8), p8-15.

[9] HUDAK P. (1996) *Building domain-specific embedded languages*. ACM Computing Surveys, 28(4es).

[10] GOLDREI S. (2007) *The Design, Implementation and Use of Domain Specific Languages*.

[11] MERNIK M., HEERING T., SLOANE T. (2005) *When and How to Develop Domain-Specific Languages*. ACM Computing Surveys, 37(4es).