



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

**Diseño e implementación del
sistema para la rehabilitación
del paciente afásico**

TESINA

Que para obtener el título de
Ingeniero en Computación

P R E S E N T A:

Aldo Jakousi Becerril Nájera

DIRECTORA DE TESINA:

Mtra. Rosa Elena Lobera Sánchez



Ciudad Universitaria, Cd. Mx., 2019

*Para Mary Carmen, quien me hace descubrir nuevos límites.
Para mi familia, cuyos sacrificios me están haciendo llegar tan lejos.
Para cualquiera que quiera cambiar al mundo.*

Agradecimientos

A mi novia Mary Carmen, de quien he recibido incondicional motivación y fuerzas para nunca rendirme.

A mis padres, Emma y Eduardo por el apoyo de tantos años, siempre manteniéndonos alertas para que podamos lograr lo que deseemos.

A mis hermanos, Anel y Axel por siempre tener algo nuevo que compartir.

A Ricardo Ruíz, cuyo esfuerzo y trabajo también está plasmado en el producto final y fue él quien me invitó a tan grande proyecto.

A todos mis amigos por todo lo que vivimos juntos y el constante respaldo ante los problemas.

A la Ing. Rosa Lobera, que nos tuvo confianza y paciencia para entregar un programa útil y completo.

A todos los profesores de la Facultad de Ingeniería que todos los días me guiaron por el duro, complejo pero gratificante camino del Ingeniero.

Al equipo de terapeutas del lenguaje del Instituto Nacional de Rehabilitación por la oportunidad de crear algo para apoyarlos y por la ayuda a lo largo de este proyecto.

ÍNDICE

INTRODUCCIÓN.....	1
OBJETIVOS.....	1
METAS.....	2
CAPÍTULO 1 - COMUNICACIÓN HUMANA.....	3
1.1 CONCEPTOS.....	3
1.1.1 <i>Comunicación</i>	3
1.1.2 <i>Lenguaje</i>	3
1.1.3 <i>Componentes del lenguaje</i>	4
1.1.4 <i>Habla</i>	4
1.1.5 <i>Alteraciones del lenguaje</i>	5
1.1.6 <i>Retraso del lenguaje</i>	5
1.2 DESARROLLO DEL LENGUAJE.....	5
1.2.1 <i>Desarrollo cognitivo</i>	5
1.2.2 <i>Desarrollo social</i>	6
1.2.3 <i>Desarrollo afectivo</i>	7
1.2.4 <i>Desarrollo semántico</i>	7
1.3 CLASIFICACIÓN DE TRASTORNOS DEL LENGUAJE.....	8
1.3.1 <i>Clasificación sobre la base de criterios neurolingüísticos</i>	9
1.3.2 <i>Clasificación sobre la base de criterios etiológicos y psicopatológicos</i>	9
1.3.3 <i>Clasificación propuesta por Narbona y Chevrie-Muller</i>	9
1.3.4 <i>Clasificación de las alteraciones del lenguaje en base al componente del lenguaje alterado y de la etiología</i>	12
CAPÍTULO 2 - TIPOS DE SOFTWARE.....	15
2.1 CONCEPTOS.....	15
2.1.1 <i>Software</i>	15
2.2 SOFTWARE DE ENTRETENIMIENTO.....	16
2.2.1 <i>Videojuegos</i>	17
2.2.2 <i>Clasificación de los videojuegos</i>	17
2.3 SERIOUS GAMES Y GAMIFICATION.....	19
2.3.1 <i>¿Por qué juegos?</i>	19
2.3.2 <i>Juegos en la educación</i>	20
2.3.3 <i>Juegos en el ejército</i>	23
2.3.4 <i>Juegos en medicina</i>	24
2.3.5 <i>Juegos para motivar el ejercicio</i>	25
2.4 DESARROLLO DE SOFTWARE.....	26
2.4.1 <i>Diseño de software</i>	27
2.4.2 <i>Desarrollo en cascada</i>	27
2.4.3 <i>Modelo de prototipos</i>	28
2.4.4 <i>Desarrollo rápido de aplicaciones</i>	29
2.4.5 <i>Modelo en espiral</i>	30
2.4.6 <i>Desarrollo de juegos</i>	31

2.4.7 Herramientas de desarrollo	34
2.4.8 Game engines	34
2.4.9 Gráficos en computación.....	35
CAPÍTULO 3 - CREACIÓN DE FANTASMAGIA	43
3.1 DESARROLLO.....	43
3.1.1 Definición del problema	43
3.1.2 Objetivo del proyecto.....	44
3.1.3 Metodología de trabajo.....	44
3.1.4 Herramientas	44
3.2 CONCEPTO DEL JUEGO.....	45
3.3 PLANEACIÓN.....	49
3.4 DISEÑO	50
3.4.1 Diseño del Gameplay	50
3.4.2 Diseño de la interfaz.....	51
3.4.3 Diseño del nivel	55
3.5 ARTE.....	56
3.5.1 Arte de la interfaz	57
3.5.2 Texturas.....	58
3.5.3 Modelos 3D.....	59
3.6 PROGRAMACIÓN	61
3.6.1 Programación de contenido.....	62
3.6.3 Programación de componentes.....	62
3.6.4 Programación de lógica.....	63
3.7 REQUISITOS FUNCIONALES Y NO FUNCIONALES	63
CAPÍTULO 4 - MODIFICACIONES Y PRUEBAS	65
4.1 PRUEBAS CON LA VERSIÓN ALPHA	65
4.1.1 Modificaciones y errores encontrados.....	66
4.2 PRUEBAS CON LA VERSIÓN BETA.....	66
4.2.1 Modificaciones y errores encontrados.....	67
4.3 PRUEBAS CON LA VERSIÓN FINAL	68
4.3.1 Modificaciones y errores encontrados.....	68
4.4 ENTREGA FINAL.....	69
CAPÍTULO 5 - CONCLUSIONES	71
5.1 CONOCIMIENTO ADQUIRIDO	72
5.2 RESULTADOS OBTENIDOS	73
5.3 TRABAJO A FUTURO.....	74
BIBLIOGRAFÍA.....	77

Introducción

La idea de que los videojuegos pueden cambiar al mundo para bien puede lucir radical, pero somos muchas las personas que hemos decidido tener ésto como nuestra misión, ya sea por gusto, por fe o por ver obstáculos de nuestra vida cotidiana ser superadas con las lecciones que tuvimos, sin saberlo, cuando preferimos no soltar un control para seguir jugando.

Los videojuegos han estado con nosotros por varios años y han pasado por bastantes cambios. Como toda nueva idea o nuevo arte, fueron recibidos con escepticismo por unos, rechazo por otros, y aceptación por el resto. Hoy en día (2017), los videojuegos ya son su propia cultura, tienen mayor grado de aceptación social y están cerca de ser considerados arte; pero más allá de volverse otro medio de expresión, también tienen la capacidad de ser herramientas de aprendizaje, de apoyo médico, medios de información o, su propósito original, una forma de entretenimiento más.

Ahora, las personas tienen acceso a una infinidad de juegos digitales: en consolas dedicadas a los videojuegos, instalados en una computadora, en páginas de internet e incluso en sus teléfonos inteligentes. Los niños ya crecen con la idea de que estos aparatos y su contenido son esenciales en la vida de las personas, entonces presentan una curiosidad natural por los dispositivos electrónicos. Aprovechando esto, más su gusto y constante búsqueda por los juegos y la diversión, los videojuegos son el perfecto medio para mantener su interés y que puedan aprender nuevas habilidades, ejercitar conocimiento o, como es el caso de este proyecto, facilitar un proceso de rehabilitación.

En esta tesina se encuentra plasmado el desarrollo de un programa de entretenimiento para computadora, que es hoy usado como material de apoyo en la terapia del lenguaje en el Instituto Nacional de Rehabilitación (INR).

Objetivos

Este juego para computadora, es el resultado del programa de servicio social llamado **DISEÑO E IMPLEMENTACIÓN DEL SISTEMA PARA LA REHABILITACIÓN DEL PACIENTE AFÁSICO**; tiene 3 objetivos principales:

1. **Mantener la atención de los niños en sus ejercicios de rehabilitación.** Es normal que la curiosidad de los niños los tenga en constantes distracciones, sin embargo esto puede representar un problema para los terapeutas que buscan que estén enfocados en ejercicios para trabajar sus alteraciones del Lenguaje.
2. **Reducir el tiempo de rehabilitación.** Es muy probable hacer que la efectividad de la terapia aumente.
3. **Ayudar en el trabajo a las terapeutas del lenguaje mediante la inclusión de nuevas tecnologías.** Uno de los principales beneficios es brindar nuevas formas de trabajo y facilitar la carga del mismo.

Se realizaron pruebas en un grupo reducido de niños dentro del INR y extendiendo en este documento mis observaciones. Faltaría que validen la efectividad del proyecto en grupos más numerosos y con metodologías de medición de parte del área médica.

En este trabajo se encuentra un caso de uso de un videojuego diseñado especialmente como herramienta en el tratamiento de niños con problemas de lenguaje; en específico, trastornos en el desarrollo semántico del lenguaje. En el primer capítulo, se habla de forma general sobre el Lenguaje Humano, clasificaciones y problemas. Se entra en detalle sobre el área del lenguaje en el que se enfoca este programa. Dentro del segundo capítulo se encuentra información sobre el desarrollo de software, metodologías, ventajas y desventajas, así como el enfoque a desarrollo de videojuegos, herramientas y metodologías de desarrollo; de igual manera se enlistan usos de los videojuegos en diferentes campos y algunos ejemplos de proyectos exitosos de extender los beneficios del software de entretenimiento.

Todo sobre la creación del juego **Fantasmagia** que se programó para su uso en el INR se encuentra en el capítulo 3, donde se detalla cada parte del videojuego y las diferentes etapas de su desarrollo: la metodología de trabajo, técnicas de optimización de gráficos, arte y programación. Posteriormente, se encuentran las pruebas que se realizaron con niños del mismo Instituto, éstas se tratan de observaciones personales y la de los terapeutas más descripción de sus reacciones y la de varios médicos del mismo Instituto.

Metas

Con este trabajo se pretende transmitir el proceso y lo necesario para la creación de un *juego serio*. Se plantean las bases del área médica y de la ingeniería con la idea de que el lector vea el origen de cada decisión del diseño final del juego. Espero poder inspirar a más profesionistas a innovar en sus campos usando nuevas ideas, conceptos y tecnologías.

Las metas del proyecto son generar un videojuego completo que responda a las necesidades de la terapia del lenguaje, usando los conocimientos obtenidos en la carrera de Ingeniería en Computación de la UNAM. El producto final debe seguir la idea de los *juegos serios* para que pueda ser usado como herramienta de rehabilitación y abra las puertas a nuevas ideas e implementaciones de los videojuegos en el área médica.

Quiero mostrar las ventajas y el alcance real que puede tener un videojuego en el comportamiento humano; quiero cambiar el concepto que se tiene sobre los videojuegos hasta que sean tratados como una nueva forma de expresión, entretenimiento o un escalón para el crecimiento de las personas.

“La buena comunicación es tan estimulante como el café, y tan difícil de dormir después de.”

-Anne Morrow Lindbergh

Capítulo 1 - Comunicación humana

1.1 Conceptos

1.1.1 Comunicación

La palabra Comunicación es un término bastante difícil de definir. De manera sencilla, Dioses (2004) lo define como “Intercambio de información” (p. 3); esta vaga descripción sugiere cambiar datos entre dos entidades pero no ofrece más detalle del cómo se logra. Froemling (2011) dice que es “el proceso de compartir significado mediante un flujo continuo de mensajes simbólicos” (p. 5). Entonces, podemos decir que la comunicación es la transmisión de información entre individuos, pero lo que nos falta es la identidad de estos seres. Dioses prosigue mencionando que aunque todos los animales se comunican entre ellos, los seres humanos lo hacemos de una forma más compleja a la que llamamos lenguaje.

1.1.2 Lenguaje

Entendemos al lenguaje como la forma de comunicación única del ser humano. Bronckart (1980) habla de él como “la suma de los hábitos verbales del individuo” (p. 23), y hace la distinción entre el lenguaje y el pensamiento, poniendo al último como las ideas que tenemos con la boca cerrada y lo define como “un lenguaje subvocal”. Piaget (1924), con el enfoque psicológico, toma al lenguaje como el “revelador” del pensamiento y que con los enunciados y estructuras con los que se habla son representantes de la inteligencia. Lecours y Lhermitte (1979) parecen juntar estas dos ideas en su definición del lenguaje:

Es el resultado de una actividad nerviosa compleja que permite la comunicación interindividual de estados psíquicos a través de la materialización de signos multimodales que simbolizan estos estados de acuerdo con una convención propia de una comunidad lingüística. (Lecours y Lhermitte, 1979, extraído de Peña, 1988).

Con estas definiciones podemos concluir que el lenguaje es un proceso complejo característico del ser humano para transmitir ideas, emociones, experiencias y cualquier “estado psíquico”. Se dice que es una función única del ser humano ya que, aunque los animales tienen un nivel de comunicación, no llega a ser tan complejo como el lenguaje humano porque no sólo se limita al nivel hablado, sino que también involucra signos gráficos o acústicos, señas físicas o gestos corporales, sin tomar al pensamiento como un “lenguaje subvocal”. Sabemos que sigue un conjunto de reglas definidas por la lengua para simbolizar lo que queremos transmitir. Además necesitamos poder materializar el sistema de reglas que dan pie a la escritura. El propósito del lenguaje es la comunicación y la representación. Se compone de signos o símbolos que se usan en una codificación determinada, para representar objetos e ideas.

La adquisición del lenguaje amplía las posibilidades de comunicación de una persona consigo mismo y con los demás; potencia el desarrollo del conocimiento, llamado cognitivo, y viceversa. Los niños lo adquieren por una necesidad de querer comunicarse para conseguir diferentes objetivos, y los padres ayudan en dicho proceso para que el niño pueda desarrollar este proceso de manera adecuada; por lo tanto, el desarrollo del lenguaje se puede ver afectado por carencias afectiva (Dioses, 2004).

1.1.3 Componentes del lenguaje

Acosta y Moreno (2005) enlistan los siguientes como los componentes del lenguaje:

- Pragmática. Es el estudio del lenguaje aplicado en contextos sociales, se interesa por el uso de las diferentes reglas que lo definen empleado en encuentros con otras personas.
- Fonología. Estudia los sonidos de la expresión lingüística de una lengua determinada desde un punto de vista funcional y abstracto.
- Fonética. Estudia el material sonoro, tratando de obtener la mayor información posible a partir del sonido y sus propiedades. Tiene 3 puntos de vista de estudio: producción (fonética articuladora), transmisión (fonética acústica) y percepción (articuladora auditiva).
- Semántica. Estudia el significado de las palabras y sus posibles combinaciones.
- Morfosintaxis. Se ocupa del estudio de las reglas que intervienen en la formación de las palabras y de las posibles combinaciones que pueden tener en las diferentes estructuras de las oraciones.

Este proyecto es el reflejo digital de un ejercicio realizado por las terapistas del INR para trabajar el componente de la Semántica, se enfoca en varios objetos de la vida cotidiana que muchos niños pueden encontrar en su casa. Las combinaciones con las que se trabajan son las características de las cosas, por ejemplo la palabra “manzana” puede combinarse con las palabras “redonda”, “roja”, “pequeña” o “dulce”. También se hace referencia a las características de las cosas que no les pertenecen, en el ejemplo de la “manzana”, ésta no puede ser “plana”, “enorme” o “suave”.

Los niños que presentan alguna alteración en la Semántica, tienen problemas al discriminar objetos de acuerdo a una característica dada, esto lo vi trabajando con las terapistas en el proceso de análisis del problema.

1.1.4 Habla

Saussure (2011) se enfoca mucho en distinguir el “lenguaje” del “habla”. El “habla” es la lengua en acción. Es la transformación del pensamiento, los signos o códigos de una lengua en una representación externa. Cada individuo selecciona de su mente los signos y reglas que la lengua pone a su disposición.

El habla generalmente es asociado con la boca, pero Saussure menciona que no hay evidencia que sugiera que el humano tenga la disposición natural de usar su boca como instrumento para expresarse. Whitney, citado por Saussure hace notar que aunque la mayoría asociamos el habla con la boca, existen también otro tipo de lenguajes que transmiten las ideas de las personas usando extremidades sin realizar ningún tipo de sonido.

1.1.5 Alteraciones del lenguaje

Son un conjunto de desórdenes en el desarrollo o adquisición del lenguaje, que se presentan en la comprensión, producción o el uso del mismo. Algunas de estas alteraciones pueden ser de largo plazo hasta durar toda la vida aunque los síntomas o problemas cambian con el tiempo (Dioses, 2004, p. 4).

Estas alteraciones pueden clasificarse en dos grupos propuestos por Martínez (2008):

1. Orgánica/Funcional
 - a. Orgánica. Se sabe la causa física que lo genera.
 - b. Funcional. No se sabe la causa física y se debe indagar en el pasado fisiológico o neurológico de la persona.
2. Producción/Recepción
 - a. Producción. Patologías que afectan a todo lo que se requiere para generar el habla, neurológica, fisiológica y anatómicamente.
 - b. Recepción. Cuando no se puede decodificar el mensaje recibido.

1.1.6 Retraso del lenguaje

La palabra retraso sugiere un desfase en el tiempo, aplicado al lenguaje se toma como una falta del desarrollo del lenguaje a la edad en la que se debió presentar dicho desarrollo. En niños, se aplica cuando un niño presenta patrones lingüísticos correspondientes a niños de una menor edad (Peña, 2002, p.229).

1.2 Desarrollo del lenguaje

El objetivo del lenguaje es la expresión de nuestros pensamientos de tal manera que podamos compartirlo con alguien más y la de poder mantener una comunicación. El lenguaje está ligado al pensamiento humano, eso quiere decir que aumenta el conocimiento cuando se ejercita y amplía el lenguaje.

En el 2009, el IMSS informó que 200 mil niños en México, de 2 millones nacidos al año, lo cual representa casi el 20%; presentan algún nivel de trastorno del lenguaje al llegar a los 2 años, donde deben ya de contener un vocabulario de alrededor de 50 palabras. Las principales causas son desórdenes neurológicos, daño cerebral, infecciones en el sistema nervioso y traumatismos craneoencefálicos. La fuerte relación que existe entre el lenguaje y el aprendizaje hace que alteraciones del primero obstaculicen al segundo. (Rodríguez, 2009)

La forma en la que los padres identifican el problema es cuando los niños no entienden cuando se les habla, habla poco, habla mal, sólo sus familiares le entienden o no relata hechos. El problema se puede agravar si no se le da un seguimiento adecuado ya que se espera que el niño adquiera el lenguaje con el tiempo o ni siquiera se considera como un problema por completo.

1.2.1 Desarrollo cognitivo

Como se mencionó anteriormente, un objetivo del lenguaje es la expresión del pensamiento, esto requiere un puente entre la mente y la lengua. Cuando le enseñamos a un

niño el nombre de un objeto o un concepto nuevo se aumenta su capacidad para usar dicho objeto o concepto según lo requiera. En futuros escenarios podrá recordar la idea y jugar con ella basándose en su significado.

Existen varias teorías sobre el desarrollo cognitivo, es decir el desarrollo de la conexión entre el pensamiento y la lengua y cómo generamos y empleamos la relación idea-lenguaje.

- **La teoría conductista.** Publicada en la obra de Skinner (1981), esta teoría toma al habla como una conducta regida por las mismas variables controladoras a las que están sujetas todas las diferentes conductas del ser humano. El lenguaje es una conducta adquirida mediante la imitación de los estímulos externos de la sociedad. Como el pensamiento es interno, éste se diferencia del habla en que éste último es influenciado por un mediador ajeno, en general, otra persona.
- **Relativismo y determinismo lingüístico.** Basado en los estudios de Sapir y su discípulo Whorf (1971). Establecieron varias hipótesis, siendo el determinismo lingüístico el de mayor peso en donde afirman que la formación del pensamiento depende del idioma que aprenda la persona. Dicho de otra manera, la manera en la que percibimos nuestro entorno y los conceptos se basan en el acuerdo del lenguaje establecido por la comunidad. Por último, la capacidad del pensamiento está limitado al conocimiento del lenguaje de la persona.
- **La gramática generativa.** Esta teoría, iniciada por Chomsky dice que el pensamiento está separado del lenguaje por completo. El humano tiene mecanismos en su naturaleza para adquirir una forma de comunicación y el pensamiento no forma parte de dicha formación. La gramática generativa forma parte del conjunto de teorías innatistas (Dioses, 2004).
- **Teorías cognitivas.** Sostiene que el desarrollo del lenguaje se efectúa a lo largo de varias etapas. El pensamiento es parte de estas etapas y viene antes del propio lenguaje. Un bebé obtiene información y lo procesa pero sólo puede obtener tanto de la absorción por lo que requiere de usar el lenguaje para separar el pensamiento de la acción, así como para representar objetos y eventos pasados (Piaget, 1972).
- **Teoría interaccionista.** Esta última teoría menciona que el aprendizaje del lenguaje se da mientras el individuo interactúa y experimenta con su medio ambiente. El niño va construyendo estructuras sobre estructuras conforme aprende nuevas cosas y éstas definen su grado de madurez, motivación, autodisciplina y estilos de aprendizaje. (Dioses, 2004).

El doctor Miguel Pérez Pereira, profesor de desarrollo psicológico en la Universidad de Santiago de Compostela, sostiene que para construir un lenguaje es necesario que se dé un desarrollo en las áreas de lo cognitivo, lo comunicacional y lo verbal; el desarrollo lingüístico no puede ser ajeno a dichos aspectos.

1.2.2 Desarrollo social

El lenguaje se adquiere gracias a que la persona es parte de un grupo social que le enseña cómo usarlo y le transmite a través de él sus valores culturales para poder así tener acceso a las funciones que su uso permite. El medio social en el que nace el niño tiene una gran influencia en el tipo de lenguaje que el niño va a manejar. De igual manera, el nivel educativo de los padres va a influir en el tipo de lenguaje que dirijan a sus hijos y que más tarde ellos van a manejar.

1.2.3 Desarrollo afectivo

El lenguaje y el desarrollo afectivo están íntimamente relacionados ya que las carencias y/o maltrato afectivo influyen en su nivel de lenguaje, retrasándolo y empobreciéndolo.

1.2.4 Desarrollo semántico

Pérez Pereira habla sobre este tipo de desarrollo que comienza cuando los niños empiezan a relacionar las palabras que van adquiriendo durante los primeros meses de vida. El desarrollo del vocabulario es lento al inicio; se puede dividir en tres periodos la adquisición de nuevas palabras (Barrett, 1986).

El primer periodo, donde el niño adquiere al menos 10 palabras, se da entre los 11 y 15 meses. El periodo siguiente, entre los 15 y 19 meses de edad, el vocabulario aumenta a 50 diferentes palabras; el ritmo de aprendizaje es mayor aunque se sigue considerando lento. El último periodo es conocido como una “explosión de vocabulario” y es donde el niño tiene una velocidad de adquisición de palabras bastante elevada, aprendiendo entre 5 y 9 palabras al día; esta etapa dura hasta los 6 años.

Una característica de las primeras etapas es el uso de las pocas palabras adquiridas. Las palabras que el niño aprende son relacionadas con contextos especiales, por ejemplo un niño puede relacionar la palabra *agua* con el líquido que se toma cuando tiene sed pero no lo relaciona con el líquido en el que se baña o el agua que corre por un río. Conforme avanza en su desarrollo y aprende nuevas palabras, empieza a darles una representación conceptual a las palabras en lugar de una relación contextual.

La descontextualización del significado de las palabras comienza durante la segunda etapa, cuando aumenta su aprendizaje de nuevas palabras hasta que llegan a las 50 palabras. Se pueden encontrar problemas de desarrollo en este periodo, aunque son comunes al desarrollo de todo niño. Un primer error es la llamada infraextensión, donde los niños conocen y emplean una palabra correctamente hasta que son presentados con un objeto que presenta diferencias con el objeto con el que están familiarizados. Por ejemplo, conoce lo que es una *mesa*, pero puede sólo relacionar la palabra con mesas cuadradas o circulares y que sean de madera; y cuando se le presenta otro tipo de mesa con una figura diferente como un hexágono y/o hecha de un material como el cristal, el niño ya no podrá relacionar ese objeto extraño para él con la palabra *mesa*. Este error se da por la forma en la que los niños comienzan a generar los conceptos de cada palabra de su vocabulario; requieren de un ejemplar que establezca las características principales del objeto. Conforme van conociendo nuevos ejemplares van ampliando los horizontes para generar una semántica similar a la adulta. El error de infraextensión se supera con la experiencia.

Otro error es la sobregeneralización o sobreextensión, éste ocurre cuando se le da un uso inadecuado del vocabulario para nombrar nuevos objetos. Un ejemplo puede ser el uso de la palabra *gallina* no sólo para referirse a una gallina, sino también para referirse a los patos, gansos, avestruces, pelícanos, etc. Este error se da por la generalización de las características comunes de los objetos que ya conoce. La *gallina* la conoce como un animal con pico y dos patas, por eso al ver un pelícano ve que cumple con las características generales que ha conceptualizado de la palabra *gallina*. Barret (1995) entonces sugiere que la sobregeneralización viene después de la superación de la infraextensión. La corrección del problema es la adquisición gradual de nuevos atributos semánticos, como pico grande, pico chico, crestas, volador, etc. De esta manera el niño va generando sus propios campos semánticos, o conjunto de palabras con significado en común.

Éste es el camino común para generar un catálogo léxico: palabras vinculadas a un contexto y eventos, significados prototipos y aparición de la infraextensión, identificación de los rasgos principales de los prototipos y aparición de la sobreextensión, y por último, formación de campos semánticos. Sin embargo, no todos los niños forman su lenguaje de la misma manera, hay muchas variables que entran en juego. Se puede dividir en dos grupos de acuerdo a las palabras que favorecen en emplear: referencial y expresiva. El estilo referencial es cuando se da por emplear más nombres comunes por sobre palabras con función gramatical y de uso personal-social (como 'hola', 'adiós', 'gracias', 'lo siento', 'sí', 'no', etc.). El grupo contrario son los niños con la forma expresiva, aunque los niños expresivos usan con mayor frecuencia pronombres demostrativos, como esto, este; para poder darse a entender en la referencia a objetos con su interlocutor.

Entonces para que existan diferentes formas de adquisición del lenguaje, no puede ser algo tan simple. Los niños deben tener una forma de discriminar significados al encontrarse con una nueva palabra, por ejemplo si a un niño se le presenta un animal nuevo y es señalado por alguien por su nombre, como una "ardilla", debe existir una forma en la que el niño sepa que "ardilla" se refiere al ser vivo y no a los objetos del ambiente como un arbusto, un árbol o en su defecto, una característica del mismo animal. La forma en la que es capaz de descartar todos los significados que le puede dar a un léxico es usando tres restricciones: La del objeto completo, la taxonómica y la de la mutua exclusividad.

La primera restricción quiere decir que cuando llega una nueva palabra al vocabulario de un niño, ésta se le da al objeto completo al que se le asoció. Así se evita que el niño referencie la palabra con una sola característica del mismo objeto. La segunda restricción, taxonómica, hace que use una palabra para asociar objetos o acciones del mismo tipo, esto evita que use la palabra "ardilla", del ejemplo, para referirse también al lugar en el que se encontraba el objeto al momento que aprendió la palabra o la acción que realizaba. Por último, la tercera restricción hace que cuando una palabra nueva aparezca aplicada a un objeto cuyo contexto ya le es familiar al niño, será asociada a un atributo, característica o acción del objeto.

No todo lo que ayuda en el aprendizaje del lenguaje es interno al niño, obviamente también la adquisición de significados depende del contexto social en el que se encuentra. La reafirmación que proporcionan los familiares al significado que ha adquirido más las herramientas internas para trabajar con el lenguaje contribuyen a la diferenciación de la forma de aprendizaje. Aún no se puede explicar por completo cuál es el proceso de adquisición de significados y hasta el momento no se ha aceptado ninguna propuesta como una completa.

1.3 Clasificación de trastornos del lenguaje

En la actualidad existen varias formas de clasificar los trastornos del lenguaje y es importante para la intervención como para la investigación estar organizados en base a un modelo que guíe el análisis de la patología lingüística. A continuación se presentan varios ejemplos de la clasificación, incluyendo la que se tomó como base para el trabajo descrito en este documento (Clasificación de acuerdo al componente del lenguaje alterado).

1.3.1 Clasificación sobre la base de criterios neurolingüísticos

Presentados por Dioses (2004) los cuadros clínicos que se pueden presentar se deben a las siguientes razones:

- Un cuadro clínico corresponde a la deficiencia de alguno de los componentes lingüísticos, tanto a nivel comprensivo como expresivo.
- Las asociaciones que se producen entre estas deficiencias.
- Las patologías relacionadas con los defectos físicos y el déficit de los procesos gnósticos (procesos de reconocimiento de objetos mediante un sentido) y práxicos (movimientos o acciones; sistema de movimientos coordinados con un propósito o función).

1.3.2 Clasificación sobre la base de criterios etiológicos y psicopatológicos

Es necesario establecer el criterio que se tendrá para indicar si una patología es congénita o adquirida. Dioses nos presenta en la tabla 1.1 el criterio de varios autores para determinar entre ambos tipos de patologías.

Autor	Criterio
Bates y Cols.	Agrupa sin distinción trastornos del lenguaje por lesión cerebral desde los 6 meses a los 6 años.
Van Hout	Se trata de una patología adquirida desde fines del segundo año, cuando aparecen las primeras manifestaciones de sintaxis.
Chevrie - Muller y Narbona	Se trata de una patología adquirida a partir de los 18 a 20 meses de los primeros indicios de comunicación pre-verbal.

Tabla 1.1. Distinción entre patologías congénitas o adquiridas.

Sin embargo, la etiología y fisiopatología no son variables suficientes para apoyar una clasificación de los trastornos de lenguaje, aún es muy difícil establecer una correlación significativa entre los síntomas de trastornos lingüísticos específicos y una etiología o disfunción neurológica determinada.

1.3.3 Clasificación propuesta por Narbona y Chevrie-Muller

Estos autores (2003) plantean el lenguaje como resultado del funcionamiento conjunto de los aspectos cognitivo y cerebral. Señalan que la clasificación debe tomar en cuenta los criterios etiológicos y lingüístico, y proponen diferentes tablas para facilitar la clasificación de acuerdo a la causa del cuadro clínico. La tabla 1.2 muestra los problemas en base a déficits físicas de las persona. En la tabla 1.3 se enlistan los trastornos derivados de problemas neuronales que afectan al lenguaje. Por su parte, la tabla 1.4 muestra la tartamudez y la tabla 1.5 tiene los trastornos relacionados con patologías y carencias en el entorno del niño. Cada una de ellas organiza los problemas del lenguaje en Congénito y Adquirido.

Patologías		ETIOLOGÍA	
		<i>Congénito</i>	<i>Adquirido</i>
Déficit anatómicos		Trastornos de la articulación de sonidos por malformaciones en el velo del paladar, labios, lengua, laringe y maxilodentarias (hueso maxilar y dientes).	
Nivel Primario	Déficit sensoriales auditivos	<i>Hipoacusias</i>	
		Déficit de transmisión (malformaciones). Déficit de percepción (genético, tóxico, infeccioso).	Déficit de transmisión (inflamación o infección en el oído). Déficit de percepción (infeccioso, tóxico).
	Trastornos del orden y control de la motricidad faringobucal	Disartrias (alteración en articulación de palabras) y afemias (pérdida del habla). Parálisis labio-gloso-faríngeas (síndrome pseudobulbar aislado, o formando parte de Parálisis Cerebral Infantil). Disartrias cerebelosas. Disartrias graves. Parálisis Cerebral Atetósica.	Lesiones neurológicas periféricas o centrales: traumáticas, vasculares. Enfermedades progresivas (distonias secundarias).

Tabla 1.2. Déficit en los instrumentos del ser humano para el lenguaje.

Patologías		ETIOLOGÍA	
		<i>Congénito</i>	<i>Adquirido</i>
Nivel Secundario	Déficit gnósicos (Reconocimiento)	Benignos: Consecuencias en articulación y ortografía. Severos: Agnosia (dificultades de reconocimiento) verbal congénita.	Síndrome de afasia-epilepsia. (Landau-Kleffner).
	Déficit prácticos (Movimiento)	Benignos: Trastornos de la articulación (Dislalias). Severos: Debilidad de movimiento en boca y rostro (Dispraxia bucofacial).	Dificultad en el movimiento en boca y rostro (Apraxia bucofacial).
Nivel Terciario	Déficit lingüísticos (comprensión - expresión)	Benignos: Retraso simple de palabra (fonología) o de lenguaje (fonología, sintaxis, léxico). Severos: Disfasias en la lengua (fonológicas), sintácticas o al buscar el nombre de las palabras (disnómicas).	Afasia por lesión o disfunción uni o bihemisférica (infecciones, traumatismos, epilepsia).
	Déficit psicolingüísticos. Patología del lenguaje escrito	Trastorno semántico-pragmático. Dificultad de aprendizaje de lectura y escritura: dislexia (discrepancia en capacidad de aprendizaje y nivel de rendimiento), disortografía (dificultad de correcta escritura).	Pérdida de capacidades anteriormente adquiridas: alexia (pérdida parcial o total de la capacidad de leer), agrafia (pérdida parcial o total de la capacidad de escritura).

Tabla 1.3. Trastornos neurolingüísticos, incluye el lenguaje escrito.

Patologías	ETIOLOGÍA	
	<i>Congénito</i>	<i>Adquirido</i>
Tartamudeo	Tartamudeo (disfemia).	Disfluencias (lesiones corticales o subcorticales).

Tabla 1.4. Tartamudez.

Patologías	ETIOLOGÍA	
	<i>Congénito</i>	<i>Adquirido</i>
Estados deficitarios	Deficiencia intelectual: Homogénea. Disarmónica (déficit verbal más marcado que las capacidades cognitivas no verbales).	
Trastornos de la comunicación	Autismo infantil y su espectro. Síndrome de Asperger. Trastorno desintegrativo de la infancia.	Mutismo selectivo. Privaciones físicas, afectivas y culturales.

Tabla 1.5. Trastornos del lenguaje en psicopatología y carencias del entorno.

1.3.4 Clasificación de las alteraciones del lenguaje en base al componente del lenguaje alterado y de la etiología

Esta clasificación fue presentada por Puyuelo (2005) quien propone clasificaciones en base a las alteraciones de los componentes del lenguaje o la etiología.

a. En función al componente de lenguaje alterado:

- Problemas a nivel semántico. Es cuando existe retardo en la adquisición de palabras y significados.
- Problemas a nivel morfológico. Se presentan dificultades en el uso de formas gramaticales.
- Problemas a nivel sintácticos. Existe dificultad en la construcción de frases.
- Problemas pragmáticos. Éstos son los problemas que existen en el uso social del lenguaje.

- Problemas en la interacción de los componentes del lenguaje. Existe déficit entre las relaciones de los componentes del lenguaje.

b. En función de la etiología.

En esta parte, Puyuelo plantea que todo problema en el desarrollo del lenguaje es denominada etiología, algunos de los cuales han sido identificados, como la limitación cognitiva, déficit sensoriales y auditivos, problemas físicos, problemas en las relaciones sociales y falta de oportunidades lingüísticas en el entorno de desarrollo. Todos estos diferentes problemas son clasificados en cuatro principales ramas: Factores centrales, Factores periféricos, Factores del entorno y emocionales, y Mixtos.

Los factores centrales son problemas que evitan el desarrollo del lenguaje a nivel cognitivo y lingüístico. Entre ellos se encuentran:

- Trastorno específico del lenguaje.
- Retraso mental.
- Autismo.
- Déficit de atención e hiperactividad.
- Traumatismo craneoencefálico (TCE).

En el caso de los factores periféricos, son aquellos que tienen que ver con problemas en los sentidos que limiten el lenguaje:

- Problemas auditivos.
- Problemas visuales.
- Sordoceguera.
- Problemas físicos.

Los factores del entorno y emocionales son todos los problemas relacionados con el ambiente de desarrollo de la persona o aspectos de la misma persona. Y por último los mixtos son alteraciones del lenguaje vinculada a problemas en varios campos.

“Los juegos son la forma más elevada de la investigación”

-Albert Einstein

Capítulo 2 - Tipos de software

2.1 Conceptos

2.1.1 Software

Conjunto de instrucciones que al ejecutarse proporcionan cierta funcionalidad esperada por el usuario. El software no es un objeto físico, como lo es el hardware, es un elemento en un sistema lógico, es leído por la computadora para mostrar el resultado. Al no ser un elemento tangible, no se “desgasta”, es decir que con el tiempo no se generarán fallas en el programa por culpa de cambios que generan las condiciones ambientales en el software como sucede con el hardware. Con el tiempo, todo sistema físico sufre desgastes por culpa del uso, vibraciones, temperatura, entre otros; durante el inicio del uso del hardware, puede presentar fallas de fábrica que con el tiempo dichas fallas se van corrigiendo hasta que llega a un nivel estable hasta que los daños por desgaste empiezan a ocurrir y el número de fallas comienza a incrementarse de nuevo. El software también presenta fallas creadas por problemas presentes durante su creación que con el tiempo los programadores corrigen hasta llegar a un nivel estable del cual, teóricamente, no puede cambiar. Eso no quiere decir que el software no puede incrementar su número de fallas conforme pase el tiempo; las fallas incrementan cuando se le agregan nuevas funciones al software, esas nuevas funciones pasarán de igual forma por un proceso de estabilización (Pressman, 2010, p. 3).

2.1.2 Categorías del software

La clasificación de programas es bastante extenso, todo comienza con sólo 3 tipos base de software: Software de aplicación, Software de sistema y Software de desarrollo. (Lemley, 2004)

- a. **Software de aplicación.** Se trata de todo programa que agrega funcionalidades más allá de las capacidades básicas de la computadora que le son de utilidad al usuario de la computadora. Existe software de aplicación para uso general, como pueden ser los procesadores de texto o el navegador de internet, y también existen los de uso específico, como software especial para contaduría o el diseño arquitectónico. El software de aplicación tiene una enorme variedad de subcategorías de las cuales se enlistan algunas a continuación según (Shethna, 2015):
 - o Software de acceso a contenido. Que pueden ser los reproductores de audio y video o los navegadores de internet.
 - o Software de entretenimiento. En donde entran los videojuegos.
 - o Software educativo. Cualquier programa que tenga como fin apoyar en la educación de cualquier área y nivel educativo.

- Software de infraestructura de empresas. Son todos aquellos programas que manejan información sobre empresas como las bases de datos, servidores o sistemas de red.
 - Software de simulación. Software que recrea un ambiente y se usa en entrenamientos y aprendizajes de algunas actividades, o apoyan la investigación simulando fenómenos físicos; como ejemplo tenemos los simuladores de vehículos o simuladores de eventos astronómicos.
 - Software de desarrollo de medios. Aquí se encuentra todo software para la creación de imágenes, audio, video, videojuegos, animaciones y todo lo que requiera contenido multimedia.
- b. Software de sistema.** Este tipo de software es creado para conectar el hardware con otras aplicaciones y de proveer una plataforma para que el software de aplicación tenga un lugar donde pueda ejecutarse (Margaret, 2006). El software de sistema se divide en dos principales grupo: Sistemas operativos y Software de utilería.
- Sistemas operativos. Permite la administración de los componentes físicos de la máquina, manejo de la memoria y despliegue de información en los dispositivos de salida como el monitor. También sirve como plataforma para ejecutar software de aplicaciones.
 - Utilerías. Son programas que le dan al usuario una ventana para monitorear y administrar los recursos y el buen funcionamiento de la computadora.
- c. Software de desarrollo.** Son el conjunto de software que contiene código escrito en un lenguaje de programación (código fuente) y de librerías para generar programas que pertenecen a uno de los 3 grupos principales de software.

El proyecto de servicio social requiere herramientas para el apoyo de terapias y tareas relacionadas con el área de Comunicación Humana en el Instituto Nacional de Rehabilitación, por lo tanto el software tiene que ser una aplicación. En mi caso, el programa va enfocado al área de terapia del lenguaje de niños, quienes prestan un interés natural a los juegos y es por eso que decidí que el mejor tipo de software que llamaría la atención de los niños y les ayudaría en su terapia, sería uno de entretenimiento.

2.2 Software de entretenimiento

La definición del software de entretenimiento por Sommerville es: todo aquel que tiene como fin el brindar tiempo de recreación o lúdico (p. 11). Normalmente los videojuegos son considerados los únicos en pertenecer a esta categoría, sin embargo aquí también entrarían los videos y otros tipos de software que formen parte de hobbies o diversión. La definición vaga que se tiene hace que muchos otros software puedan entrar a esta categoría y es por eso que sólo se asocian a esta categoría los juegos interactivos.

Todos los tipos de software anteriores en general tienen un uso profesional. En contraste, el software de entretenimiento normalmente no es visto como posible herramienta para resolver problemas o como software de apoyo en tareas específicas; sin embargo poco a poco ha ido demostrando su efectividad y ventajas como software de apoyo en varias áreas.

El ejemplo de software de entretenimiento por excelencia son los videojuegos; que han sido tratados como juguetes o una simple forma de entretenimiento. Algunas veces se han visto

involucrados en disputas sobre violencia, sexismo, bullying, entre otras. Con el pasar del tiempo, el desarrollo de videojuegos ha mejorado y se ha demostrado las posibilidades que trae este tipo de software. Jane McGonigal (2010) se ha dedicado a demostrar el impacto que tienen los videojuegos en las personas. La filosofía de McGonigal se centra en juegos de realidad alternativa y multijugadores como medios para generar soluciones a grandes problemas usando inteligencia colectiva de todos los jugadores. Los videojuegos brindan motivación al jugador para realizar todo tipo de actividades; podemos traer este tipo de placeres y retos con la que se crean los juegos a actividades de la vida real para motivar a las personas a realizar trabajos y tareas no deseadas o aburridas.

2.2.1 Videojuegos

Me limitaré a hablar de los videojuegos, definición y clasificación, ya que por sí mismo son muy extensos y abarcan un campo muy amplio. Los videojuegos involucran interacción del usuario con una interfaz que es el software y espera cierta retroalimentación por parte del programa. Su principal uso es para la recreación aunque también se le han dado usos informativos y educativos (Granic, Lobel y Engels, 2014).

La interactividad brindada por estos programas debe cumplir y lograr ciertos aspectos como la inmersión del usuario a un mundo nuevo o experiencias que son imposibles de vivir en la vida real. La inmersión se logra brindando estímulos a los sentidos de una persona ya que son el puente de la realidad con la misma persona usando diferentes tecnologías para cada sentido del cuerpo, como la realidad virtual, realidad aumentada, audio 3D, por mencionar algunos ejemplos. Esto nos lleva a la principal característica que brindan los juegos, que no tiene ningún otro medio: el control que tiene el usuario de manipular ese nuevo mundo y lo lleva a vivir experiencias únicas.

La forma en la que el usuario se puede comunicar con los juegos se le llama la mecánica del juego, que define todas las acciones que puede realizar el jugador para poder interactuar con el mundo virtual. El conjunto de mecánicas forman el *gameplay*, o jugabilidad, del juego, que ya es lo que define el objetivo, la forma de juego y las reglas.

2.2.2 Clasificación de los videojuegos

Existen varias clasificaciones de los videojuegos, Crawford (1997) propone dos categorías principales: Habilidad-y-Acción y Estrategia; sin embargo la complejidad de las posibilidades de los juegos demanda algo más completo; se enlistan algunas a continuación:

- a. **Clasificación por género.** Así como las películas, novelas o series, los videojuegos se pueden categorizar por su género, sin embargo otros medios usan como base el mundo en el que se desarrolla su contenido o las experiencias o vivencias que quieren transmitir, en el caso de los videojuegos la base para definir un género es por el *gameplay*. Los diferentes géneros contienen a su vez subgéneros que hablan sobre las mecánicas de manera más específica. Entre los géneros principales tenemos:
 - *Acción.* Son juegos con mecánicas que requieren que el jugador responda de manera rápida, precisa y con buen tiempo para superar los diferentes obstáculos. Este género es considerado el más amplio, se enfocan principalmente en mecánicas de combate y tiene muchos subgéneros como Los *first-person shooters*, juegos de pelea, *hack and slash*, entre otros.
 - *Aventura.* Los juegos de aventura son aquellos cuyo *gameplay* no contiene retos de reflejos rápidos o acción. El objetivo de muchos de estos juegos es resolver

acertijos presentados de diferentes formas. Entre sus subgéneros se encuentran, pero no están limitados a, *text adventures*, *graphic adventures*, novelas gráficas, narrativos, de exploración e híbridos.

- *Acción-Aventura*. Combina elementos de los juegos de aventura y acción.
- *Role-Playing*. Son juegos basados en los juegos de rol como “Calabozos y Dragones” en donde el jugador toma el papel de un personaje y éste va creciendo en habilidades.
- Simulación. Son juegos diseñados para simular aspectos de una actividad de la vida real o realidad ficticia. El género de simulación es muy grande y tiene bastantes subgéneros como simulación de vehículo, simulación de vida y simulación de construcción o administración.
- *Estrategia*. Este género incluye juegos que requieren planeación amplia y cuidadosa para sobreponerse a los obstáculos que presente el juego.
- *Deportes*. Son todos aquellos juegos que están basados en actividades deportivas de la vida real o que son competitivos por equipos y se pueden jugar en equipos formados por personas y no por el mismo sistema.

Chris Crawford plantea que esta clasificación por género es propensa a cambiar por las nuevas técnicas de desarrollo de juegos ya que hay casos de juegos que caen en varios géneros como acción y *role-playing* (Crawford, 1997).

b. Clasificación por contenido. Este tipo de clasificación es similar a la clasificación por edades de otros medios. Existen diferentes tipos de clasificación dependiendo de la región y el país, la usada en México es la establecida por el *Entertainment Software Rating Board* (ESRB). Las clasificaciones dadas por esta organización son:

- *Early Childhood (EC)*. Para juegos cuyo contenido está enfocado para niños de 3 años hacia arriba.
- *Everyone (E)*. Con contenido para el público en general.
- *Everyone 10+ (E10+)*. Son juegos con contenido para público en general y edades mayores a 10 años.
- *Teen (T)*. Clasificación para juegos con contenido para edades de 13 años o más.
- *Mature (M)*. Esta clasificación es reservada para juegos con una mayor cantidad de violencia y otro contenido más explícito apto para personas mayores a los 18 años.

c. Clasificación por propósito

- *Core Games*. Aquí entran todos los videojuegos tal y como los conocemos sin agregar ningún objetivo extra al entretenimiento.
- *Casual Games*. Se tratan de juegos con mecánicas simples y fáciles de entender. Tienen la posibilidad de poder entrar y salir al juego en casi cualquier momento ya que su principal fin es entretener por un tiempo corto. Por estas características es por lo que estos juegos generalmente se encuentran en teléfonos celulares.
- *Serious Games*. Éstos son los juegos cuyo objetivo es más que ofrecer entretenimiento; tienen un fin educativo, informativo o como apoyo a proyectos de

investigación u otras áreas que puedan verse beneficiadas por el uso de juegos. El ejemplo por excelencia para este tipo de juegos son los simuladores, como los simuladores de vuelos de aviones, que son usados por la fuerza aérea estadounidense para entrenar a sus pilotos. También entran en esta categoría los juegos que ayudan en terapias, como en el tratamiento de quemaduras o terapias psicomotrices.

2.3 Serious games y *gamification*

Los juegos serios, *serious games* o *serious play*, son una tendencia que está surgiendo y está ganando gran aceptación en cada vez más áreas de trabajo. El término ganó mucho auge en los últimos años con los videojuegos pero la idea ya se tenía desde antes de la era digital, se aplica también para juegos de mesa o juegos de cartas cuyo propósito principal era algo diferente al entretenimiento de los participantes.

Michael Schrage, investigador del MIT y autor del libro *Serious Play* (2000), lo define como cualquier herramienta, tecnología o juguete que les permita a las personas mejorar su forma de jugar en serio con la incertidumbre y que garantice el aumento de calidad de la innovación.

La definición más socorrida y usada, incluso en la actualidad, es la dada por Clark Abt (1970) en su libro *Serious Games*: “Son juegos que tienen un propósito educativo explícito y cuidadosamente pensado y no son destinados para el entretenimiento. Esto no quiere decir que los juegos serios no son, o que no deberían, ser divertidos.”

Conforme la tecnología ha ido avanzando y se ha expandido su uso a varias áreas, desde la educación, pasando por medicina, mercadotecnia, las ciencias exactas, administración, y muchas más; el concepto de “juegos serios” se ha limitado para el software cuyo fin es diferente al entretenimiento y entregado como un videojuego.

2.3.1 ¿Por qué juegos?

En la actualidad se ve a los videojuegos como un escape a la realidad, como un pasatiempo para niños y que no tienen un uso. Sin embargo, conforme la cultura de los videojuegos crece en popularidad, el estudio del efecto de los videojuegos en las personas aumenta en diferentes áreas, desde psicológicas hasta motrices.

Jane McGonigal (2011) habla de cómo los videojugadores desarrollan habilidades como si estuvieran ejercitándose físicamente. En conjunto, las personas dedican 3 mil millones de horas de juego a la semana. Un jugador promedio en los Estados Unidos juega 1 o 2 horas al día pero existen jugadores “extremos” que dedican en promedio 45 horas de juego a la semana. Estadísticas del Instituto Nacional de Estadística y Geografía (2016), dicen que el promedio de horas dedicadas a la semana por los jóvenes mexicanos entre los 15 y 29 años en el año 2014 es de 4.3 horas.

Lo que los jugadores encuentran en los mundos extraordinarios que brindan los videojuegos son retos y placeres que la vida real no ofrece. Esos retos afinan su potencial lo cual desarrolla a una persona con un conjunto especial de habilidades. Según McGonigal los juegos

brindan 4 elementos que los hacen tan llamativos y populares; esos elementos son: satisfacción por el trabajo, esperanza real de triunfo, conexiones sociales fuertes y la oportunidad de ser parte de algo más que nosotros mismos. Dentro de un videojuego, el jugador siente una satisfacción al cumplir metas, siente un compromiso por cumplir y hasta tiene una causa heroica. Esto fomenta las ganas de superación de los jugadores, quizá reflejándose en metas de vida o profesionales.

La esperanza de un triunfo es lo que hace a los jugadores muy optimistas. Ante las derrotas en los diferentes desafíos de un videojuego, la mayoría de la gente lo reintenta y no descansa hasta lograr vencer el problema. Y no sólo se trata de un alto optimismo, sino que también tienen el reflejo de actuar en el momento en el que se les presenta el problema. Se puede aprender a tener esta perseverancia y urgencia de actuar para aplicarlo a problemas y conflictos en la vida real.

Los lazos sociales entre los jugadores también son algo característico de los videojuegos. Aunque muchos creen que los videojuegos aíslan a las personas del contacto humano o de las relaciones sociales, McGonigal afirma que nos gusta y confiamos más en una persona con la que hayamos jugado, aunque nos haya vencido. Sabemos que estarán en el juego, que las reglas serán respetadas y que se quedarán hasta el final del juego, así es como se forman mejores lazos y confianza entre las personas.

Los juegos están especialmente diseñados para que las personas dediquen el mayor tiempo posible, siempre tienen algo que ofrecer y siempre tienen algo en lo que los jugadores puedan hacer algo. El jugador promedio siempre está a la búsqueda de querer hacer algo nuevo, no está perdiendo el tiempo, está en constante búsqueda de ser productivo porque se siente bien y feliz resolviendo problemas que si no estuviera haciendo algo, aunque sea en un mundo virtual.

La satisfacción que tiene al realizar cosas, optimismo y urgencia por resolver problemas, los fuertes lazos sociales y las ganas de querer ser parte de algo grandioso son ahora aptitudes que forman parte de la gente que dedica un tiempo importante a trabajar y esforzarse en mundos virtuales, y los hacen sin dudar porque es algo que les encanta y obtienen mucha satisfacción. El único inconveniente es que lo hacen en mundos virtuales; debemos de poder poner esos 4 elementos que hacen a un videojuego enganchable en otras actividades y tareas que hacemos en la vida real. De ahí nace la idea del *Gamification*, es el uso de todo lo que hace un videojuego algo llamativo y adictivo en lugares que normalmente se consideran fuera del contexto de un juego para que los usuarios tengan una mayor participación y un incremento en el desempeño.

En otras palabras, el *Gamification* se trata de hacer una actividad normal un juego, usando elementos como tabla de puntajes, logros, niveles, recompensas o competitividad. En un rango más amplio, los Juegos Serios forman parte del concepto de *Gamification* y es algo que cada vez es más usado por más grupos de desarrolladores de software.

2.3.2 Juegos en la educación

Quizá la educación sea el área que tiene más ejemplos del uso de videojuegos. En la parte académica tenemos juegos como Class Dojo (Fig. 2.1), que se trata de una plataforma online donde el profesor registra a su clase con sus alumnos, cada uno con un avatar el cual puede ser personalizado por ellos mismos. La mecánica se enfoca en las acciones que realizan los alumnos en la clase, y se reflejan en puntos positivos o negativos para cada jugador. Por ejemplo, si el alumno llega temprano a la clase, le puede ameritar un punto, si ayuda a sus compañeros a resolver dudas puede obtener 3 puntos; y de igual manera se pueden restar

puntos por acciones negativas como puede ser hacer ruido o distraer a sus compañeros de clase. El profesor ajusta las acciones y los puntos que se ameritan por cada una de ellas. Agregado al sistema de puntos, Class Dojo también brinda un mecanismo de comunicación entre el profesor y los padres de familia, que pueden recibir mensajes personales o globales sobre el comportamiento de los alumnos o reportes sobre el progreso de la clase. Esto logra también un acercamiento mayor de los mismos padres de familia a lo que ocurre en la clase.

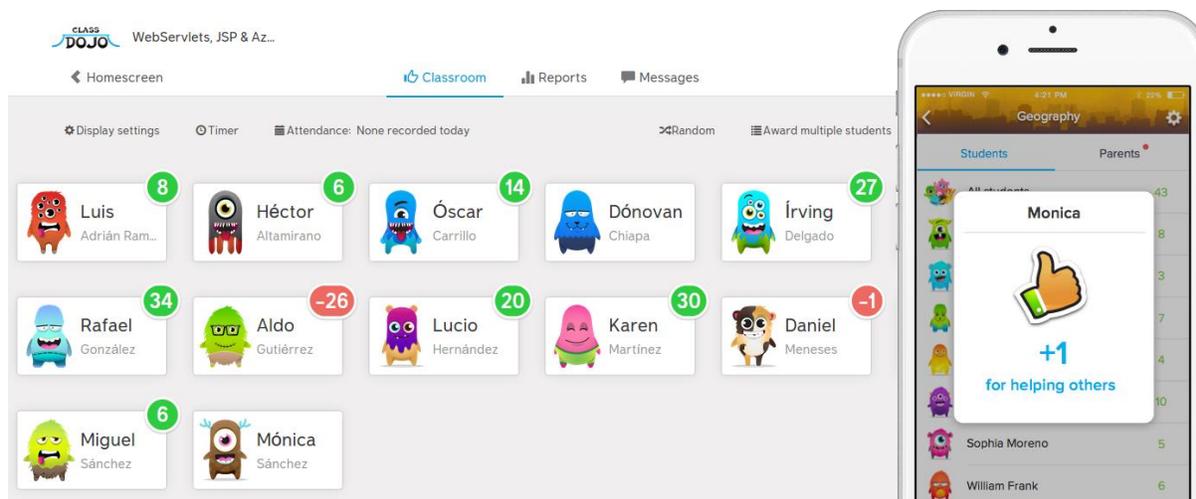


Fig. 2.1. Plataforma Class Dojo (<https://www.classdojo.com/>) (2015).

Similar a Class Dojo, también existe Classcraft (Fig. 2.2), otra plataforma para aplicar mecánicas de juegos en la clase. De igual manera, se registran los alumnos y alumnas de la clase pero a diferencia del Class Dojo, los alumnos forman equipos, eligen avatares y personalizan sus habilidades especiales para apoyarse durante la clase en equipo o individualmente. Las habilidades les ayudan a ganar puntos de experiencia, que se dan por el profesor como recompensa a acciones positivas en clase o por calificaciones aprobatorias en los exámenes. Las acciones negativas también tienen repercusiones en los avatares de alumnos los cuales pierden puntos de vida y en caso de perder todos sus puntos de vida se les asigna un castigo al azar. Cada una de estas opciones, acciones, castigos, habilidades especiales, recompensas y eventos son completamente personalizables por el profesor de la clase de acuerdo a sus necesidades y características de su clase.

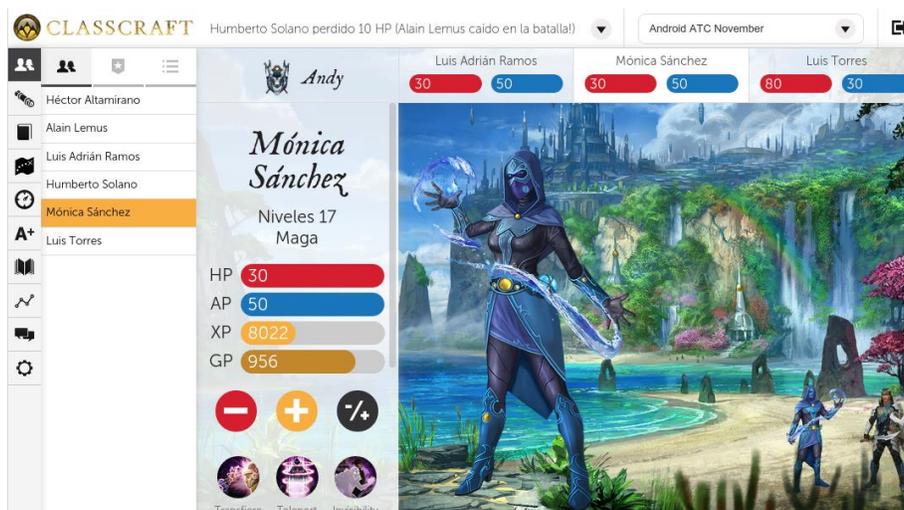


Fig. 2.2. Plataforma CLASSCRAFT (<http://www.classcraft.com/es/>)(2015)

Tanto Class Dojo como Classcraft están en constante actualización, así que con el tiempo van agregando nuevas características a sus sistemas de acuerdo a la retroalimentación que obtienen de sus usuarios y nuevas ideas de sus creadores. Ambos sistemas cuentan con comunidades que ya hacen uso alguna de dichas plataformas en salones de clases de escuelas en todo el mundo.

En México también existen proyectos de *Serious Games* aplicados en la educación. Caldera Estudio (<http://caldera-estudio.com/>) es un autoproclamado laboratorio mexicano dedicado a crear juegos, no limitado a videojuegos, enfocados en niños para fomentar su pensamiento y formación.

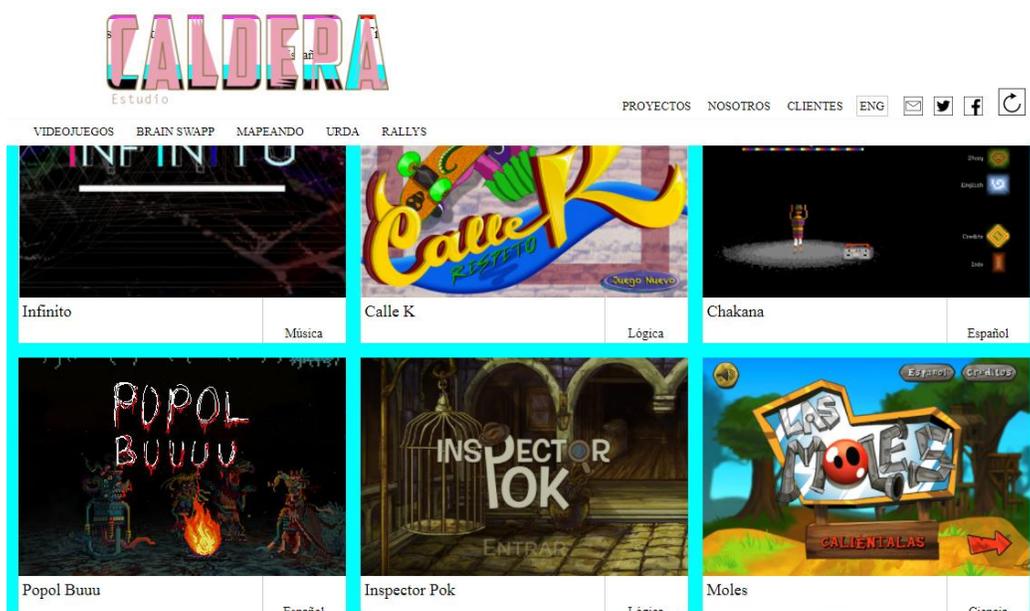


Fig. 2.3. Página de Caldera Estudio con algunos de sus juegos. (<http://caldera-estudio.com>)

Empresas que buscan experiencias lúdicas que enseñen o fomenten la cultura en las personas son los clientes de Caldera Estudio. “Dale vida al árbol Ramón” es un ejemplo de un juego que se encuentra en el Papalote Museo del Niño. En él, el jugador tiene a su cargo el cuidado del árbol Ramón, y sólo conociendo sus partes y colocándolas en los lugares correctos, es como va “creciendo” el árbol. En su página web se encuentra una larga colección de sus proyectos (Fig. 2.3).

También se pueden hacer uso de juegos que normalmente tienen un enfoque más de entretenimiento que académico, pero que se llevan a cabo en una época específica de la historia o hacen crítica social de alguna ideología política o pueden iniciar un debate sobre la psicología del comportamiento humano; así como un profesor podría usar una película, un libro o algún otro material de apoyo.

2.3.3 Juegos en el ejército

Los ejemplos más comunes de juegos serios son los simuladores, los cuales en el ejército pueden ser simuladores de vuelo o de batalla. Su propósito es el de entrenar a pilotos o soldados antes de enfrentar escenarios en la vida real. Esto reduce el riesgo de pérdidas humanas y materiales en ejercicios reales ya que los candidatos deben de cumplir varias horas en los simuladores antes de que se les confíe con equipo de verdad.

El ejército estadounidense hace uso de simuladores porque son más baratos y fáciles de instalar o transportar que tener que enviar a los soldados a una instalación de entrenamiento, darles equipo y contratar actores. Entre los simuladores que usan está Virtual Battlespace 2 U.S. Army (Fig. 2.4) que es una plataforma de entrenamiento que simula ambientes realistas con terrenos dinámicos y cientos de soldados y civiles virtuales. Soporta hasta 100 personas conectadas en red por cada ejercicio.



Fig. 2.4. Virtual Battlespace 2: U.S. Army (VBS2)

También existe una versión de VBS2 llamado VBS2 Fires, que se enfoca en la simulación del llamado a fuego de artillería, morteros, MLRS y apoyo naval. Las órdenes son enviadas desde una simulación del VBS2 hechas por otros participantes.

Otro simulador empleado se llama Bilateral Negotiation Trainer (BiLAT), crea un escenario para practicar la habilidad de llevar a cabo una reunión y negociación en un contexto cultural en específico. Los estudiantes toman el papel de oficiales militares y se reúnen con simulaciones de líderes locales y tienen que interactuar con ellos para lograr objetivos establecidos (Fig. 2.5).



Fig. 2.5. Bilateral Negotiation Trainer (BiLAT)

2.3.4 Juegos en medicina

Existen empresas dedicadas en su mayoría al desarrollo y estudio de los resultados brindados por los juegos serios. Un ejemplo es la empresa FirstHand Technology que ha desarrollado juegos con diferentes propósitos y para diferentes áreas de trabajo. No se limitan a entregar programas de computadora, sino que buscan emplear nuevas tecnologías como la realidad virtual.

El juego SnowWorld es un gran ejemplo de uno de los trabajos de FirstHand Technology. SnowWorld es un juego de realidad virtual empleado en el tratamiento de quemaduras de un hospital. El paciente se adentra a un mundo de nieve en realidad virtual durante su tratamiento de quemaduras. Durante el proceso médico, el paciente vive en un mundo cubierto de nieve y juega a lanzar bolas de nieve. Los tratamientos pueden ser muy dolorosos para el paciente y hay ocasiones en las que no se puede hacer uso de calmantes para aliviar el dolor sufrido. El propósito que tiene SnowWorld en este escenario es el de brindar una alternativa al uso de drogas para suprimir el dolor.

Otro ejemplo, llamado IraqWorld, se realizó entre FirstHand junto con la Dr. Sarah Miyahira y el Dr. Ray Folen del Centro Médico Militar Tripler en Hawaii; y el Dr. Hunter Hoffman de la Universidad de Washington IraqWorld consta de una serie de simulaciones de realidad virtual para ayudar a soldados a tratar el Síndrome de Estrés Post-Traumático. La simulación consta de un recorrido en un vehículo militar patrullando una ciudad iraquí. El terapeuta va

eligiendo niveles de estrés para el paciente, puede empezar con niveles bajos, con soldados patrullando, niños en las calles, sonidos de islámicos de rezos, etc. Mientras el terapeuta le pide que vaya relatando eventos similares que le hayan ocurrido con la idea de encontrar la fuente de su trauma. Con el avance en las sesiones, los niveles de estrés dentro de la simulación van aumentando; explosiones en la calle, disparos o atentados. El soldado enfrenta sus miedos y recuenta sus experiencias para que se cure.

Un programa que también tienen pero que sirve para el tratamiento de la Aracnofobia se llama SpiderWorld (Fig. 2.6). Creado por el Dr. Hoffman, ponen a personas con un severo caso de Aracnofobia a tratar con arañas virtuales para afrontar sus miedos empezando con algo irreal. La esencia virtual le da mayor control al terapeuta sobre la intensidad de las sesiones que si se tratara de un tratamiento con animales de verdad. La meta es lograr que el paciente supere el miedo y pueda tener una vida normal.



Fig. 2.6. SpiderWorld. 2005 Hoffman and Firsthand

En su página de internet (www.firsthand.com) muestran la información sobre éstos y más de sus trabajos y los resultados obtenidos tras su aplicación en la vida real.

2.3.5 Juegos para motivar el ejercicio

Una actividad que cuesta trabajo que sea realizada por mucha gente es el ejercicio. Todos deben ejercitarse regularmente, la vida sedentaria es causante de graves problemas para la salud y aun sabiendo eso a muchos les cuesta trabajo pararse y ejercitarse de vez en cuando. El problema es que para que se vayan viendo los resultados y ventajas del ejercicio, debe pasar mucho tiempo y debe haber constancia y disciplina por parte de la persona; muchas veces se desea ver resultado de nuestras acciones de inmediato.

Aquí es donde los videojuegos nos ayudan a adoptar a las rutinas de ejercicio en nuestra vida. Podemos incluir juegos para motivarnos a realizar una actividad física, con pequeñas metas o competiciones y tener ciertas recompensas inclusive para que nos “engañen” a hacer ejercicio. Tal es caso de *Zombies, Run!* (Fig. 2.7), un juego para teléfonos móviles donde el jugador debe correr para avanzar su historia. En *Zombies, Run!* el jugador se encuentra en un mundo donde los zombies son una realidad y debe correr para cumplir ciertas misiones como conseguir suministros para su comunidad de sobrevivientes, o para ayudar a gente en problemas o

simplemente para huir de los zombies. El juego usa sólo audio para ir narrando los eventos y va siguiendo el movimiento del jugador con el GPS del teléfono. Guarda también estadísticas y se pueden compartir con amigos para comparar resultados y fomentar la competitividad. Es una forma excelente poner a la gente a correr mientras se divierte.

Existen otros juegos para motivar a la gente a hacer algún tipo de ejercicio; los simuladores de baile son muy famosos y socorridos como ejemplos claros de videojuegos para ejercitarse. Juegos como *Dance Dance Revolution* o *Just Dance*, requieren que el jugador se mueva o haga pasos de baile al ritmo de la música para ir sumando puntos y ganarle a sus amigos. La *Dance Dance Revolution* ha logrado entrar a las escuelas norteamericanas como una actividad en sus clases de Educación Física para reducir la obesidad infantil. Forma parte de un proyecto llamado Let's Move! cuyo objetivo es fomentar la actividad física en los niños usando ejercicios poco comunes o diferentes a los acostumbrados -normalmente, videojuegos.



Fig. 2.7. Zombies, Run! 2015 Six to Start.

2.4 Desarrollo de software

Ya que se tiene la idea de qué tipo de software se quiere desarrollar hay que ver la forma de desarrollo que se siguió de acuerdo a lo definido por la Ingeniería de software. Primero se enlistan las metodologías de desarrollo de software y cuál es la adecuada para el desarrollo de un software de entretenimiento. La metodología de desarrollo de software es un marco usado para la planeación, estructuración y control del proceso del desarrollo de sistemas de información. La mayoría de las metodologías tienen los siguientes procesos dentro de su modelo de desarrollo:

- Análisis del problema
- Estudio de Mercado
- Toma de requisitos
- Planeación o diseño de la solución por software
- Implementación del software
- Pruebas
- Mantenimiento y corrección de bugs

Las diferentes metodologías tienen fortalezas y debilidades que las hacen adecuadas para el uso en ciertos escenarios pero todas hacen uso de los procesos enlistados anteriormente.

2.4.1 Diseño de software

Probablemente la parte más importante en el desarrollo de software, es cuando se abstrae la solución del problema a una solución en software que cumpla con las metas establecidas durante los análisis previos. El producto de la fase de diseño es una documentación que determina las especificaciones del software, como pueden ser la siguiente:

- **Software automatizado o centrado en el usuario.** Donde se determina si el programa trabaja independientemente o si requiere la participación de un usuario para que trabaje correctamente.
- **Independiente o dependiente de una plataforma.** Si el software puede ser ejecutado en un sistema específico o si puede ejecutar en cualquier sistema. Las variables que determinan esta especificación podrían ser el Sistema Operativo o la arquitectura de la computadora.

2.4.2 Desarrollo en cascada

Es una metodología secuencial en la cual el desarrollo se realiza como una serie de procesos en orden de manera escalonada o hacia abajo (como una cascada) a través de varias fases:

- Análisis de requerimientos
- Diseño de software
- Implementación
- Fase de pruebas
- Integración (sólo si hay varios subsistemas) o pruebas del sistema
- Mantenimiento

La idea de este modelo en cascada es que el desarrollo es secuencial con cierto grado aceptado de superposición y de retorno entre fases (Fig. 2.8). Se hace énfasis en la planeación, calendario de actividades, días meta, presupuestos e implementación de todo un sistema al mismo tiempo. Se tiene un control estricto durante la vida del proyecto a través de una documentación, revisiones y aprobaciones por el usuario o cliente que suele dejarse al terminar una fase y antes de comenzar la siguiente (Sommerville, 2011).

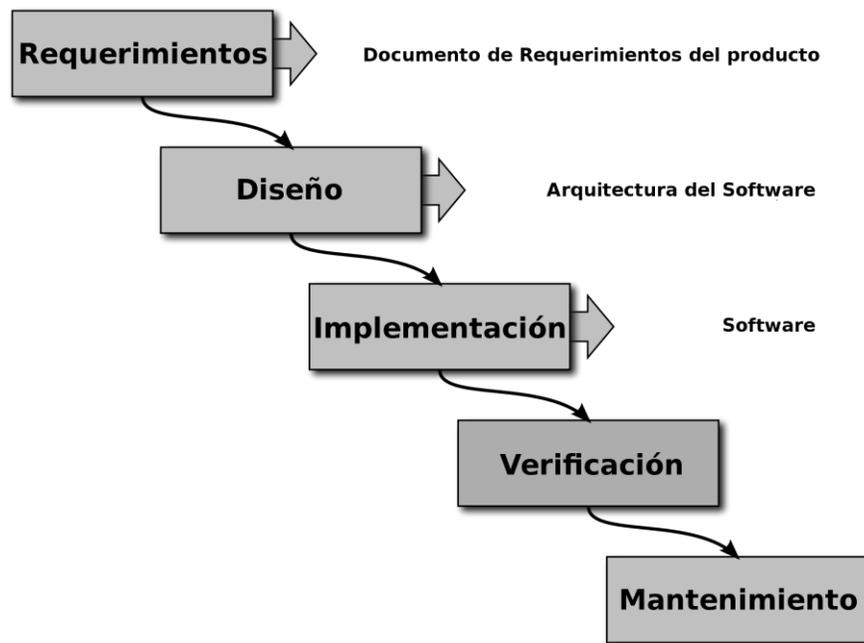


Fig. 2.8. Metodología de desarrollo de software en cascada.

Este modelo se aplica tradicionalmente al desarrollo de software común, aunque su ejecución pura limita el flujo del desarrollo “hacia abajo” sin la posibilidad de regresión a fases pasadas que ya hayan sido terminadas, esto lo hace un modelo poco flexible lo cual puede resultar en problemas de entregas tardías o problemas con el presupuesto dado para el proyecto.

2.4.3 Modelo de prototipos

Este desarrollo se basa en la creación de prototipos. Sommerville menciona que esta metodología debe ser usada en conjunto con otras para entregar un producto final. El objetivo es reducir los riesgos del proyecto mediante la segmentación de algunas fases de desarrollo para facilitar el proceso y sea más fácil la transición entre las diferentes etapas. Se debe involucrar al usuario más para que el producto final sea del agrado del mismo.

En general, se debe crear una versión pequeña del sistema en desarrollo y poco a poco se va modificando hasta cumplir con los requerimientos del usuario. Los prototipos se crean sólo para ayudar en el desarrollo, normalmente son descartados al concluir su uso, sin embargo puede ser utilizado para que evolucione al producto final.

Sommerville habla de las fases del desarrollo por prototipos y se pueden resumir en los siguientes pasos:

1. Identificar los requerimientos. Determinar los requerimientos básicos incluyendo la información de entrada y salida, aunque algunos detalles pueden ser omitidos, como la seguridad.
2. Desarrollar un prototipo inicial. El primer prototipo se crea, normalmente sólo incluye la interfaz gráfica.
3. Revisión. Los clientes, incluyendo los usuarios finales, prueban el prototipo y brindan retroalimentación sobre cambios y adiciones.

4. Repaso y mejora del prototipo. Se mejora el prototipo de acuerdo a la retroalimentación. Es probable que se tenga que evaluar qué se encuentra dentro del rango del contrato. Si se realizaron cambios, se repite el paso 3 y probablemente el 4.

Las ventajas de usar prototipos es que se reducen los tiempos y costos de producción, ya que brindan una recolección de requerimientos de mayor calidad y aumentan las posibilidades de entregar un programa con el que el usuario quede satisfecho. Otra ventaja es el aumento de la involucración del usuario con el proyecto. Los prototipos requieren que el usuario evalúe el progreso del proyecto antes de que se tenga un avance considerable del software y sea más difícil realizar cambios.

Cuando se realizan prototipos hay que tener cuidado de que el cliente no confunda el prototipo con el producto final o que crea que tendría funciones que no están dentro del contrato. Otro problema puede venir con los desarrolladores al momento de desechar el prototipo ya que pueden estar muy apegados con el software al cual el invirtieron tiempo y esfuerzo para crearlo.

Un punto muy importante de los prototipos es que deben ser creados de manera rápida, hay que tener cuidado de que el tiempo de desarrollo de los mismos no se exceda ya que sería pérdida de tiempo y dinero. Por último, el uso de la metodología de prototipos implica mayor recursos destinados para el equipo de desarrollo así que no cualquier empresa puede darse el lujo de hacer usos de prototipos.

2.4.4 Desarrollo rápido de aplicaciones

El Desarrollo Rápido de Aplicaciones, *Rapid Application Development* (RAD), es una metodología de desarrollo iterativa que se enfoca en la rápida creación de prototipos en lugar de la planeación. Surge en respuesta al rápido avance de la economía, con el fin de desarrollar software lo más rápido posible (Whitten y Bentley, 2007, p. 98). La planeación en este modelo se hace durante el desarrollo del software, esto permite que la escritura del programa sea más rápido y, quizá lo más importante, permite que los requerimientos del sistema se puedan ajustar de acuerdo al conocimiento que se obtiene conforme el proyecto avanza. También se enfoca en la flexibilidad de los procesos que se pueden adaptar de acuerdo a la evolución del programa.

El proceso para el desarrollo rápido de aplicaciones se divide en 4:

1. **Fase de planeación y requerimiento.** Como su nombre sugiere en esta fase se juntan los usuarios, administradores y miembros del equipo de desarrollo para discutir y ponerse de acuerdo en las necesidades del negocio, el rango del proyecto, límites y requerimientos.
2. **Fase de diseño.** Durante esta fase, los usuarios finales trabajan junto a los analistas de sistemas para diseñar modelos y prototipos que representan todos los procesos del sistema. La idea es que el usuario pueda entender, modificar y aprobar un modelo del sistema que cumpla con sus necesidades.
3. **Fase de construcción.** Esta ya es la fase completa de desarrollo donde se empieza la programación de la aplicación. El usuario sigue presente durante este proceso para brindar retroalimentación cuando se necesite ya sobre el sistema final.
4. **Fase transición.** Es la unión de muchos procesos, desde las pruebas. como la implementación, cambios y hasta la capacitación de los usuarios. La compresión de todos estos procesos en una sola fase hace que el proyecto se construya, instale y se entregue más rápido.

Las ventajas que trae la metodología RAD es una mejor calidad ya que la participación e interacción del usuario con la evolución de los prototipos hace que el producto final termine siendo más útil para la resolución de problemas del negocio. También reduce el porcentaje de riesgo, aunque a primer vista RAD se enfoque más en la velocidad de desarrollo, la metodología puede enfocarse desde un principio en factores clave de riesgo y se ajusta a ellos basados en evidencia empírica recolectada al inicio de los procesos. Por último, se entregan más proyectos terminados y a tiempo usando el RAD ya que este se enfoca en el desarrollo y la posibilidad de que un fallo grande de desarrollo retrase o detenga al proyecto se reduce (Whitten y Bentley, 2007, p. 100).

Para algunos desarrolladores, el cambio de una metodología común a la RAD puede resultar conflictivo con la forma en la que están acostumbrados a desarrollar. El enfoque RAD tiene más interacción con el cliente durante casi todo el ciclo de vida de desarrollo, esto implica que la empresa esté dispuesta a invertir tiempo y personal especializado para las reuniones con el cliente y además el desarrollo del producto.

Otra ventaja del RAD es que ofrece más flexibilidad en sus procesos de desarrollo; lo ideal es que pueda adaptarse de acuerdo a los problemas y oportunidades que surjan en el camino. Sin embargo, hay que tener un cuidado de que la flexibilidad y el control de proyecto se mantengan en un equilibrio ya que tener más de una sacrifica cantidad de la otra, es decir mientras más flexibilidad tenga un proyecto se tiene menos control y vice versa.

También hay que tener cuidado que los desarrolladores dejen de estar agregando pequeñas mejoras a los prototipos e ignoren lo importante del proyecto como las metas del diseño. Los proyectos RAD usualmente se asignan a equipos de desarrollo pequeños o medianos, no es recomendable para proyectos de gran escala.

2.4.5 Modelo en espiral

El modelo en espiral (Fig. 2.9) es una combinación entre aspectos importantes del modelo en cascada y la metodología de prototipos. Propuesto por Boehm en 1988, hace enfoque en análisis de riesgos, algo que ninguna otra metodología previa a esta había tenido en cuenta, lo cual es ideal para proyectos complejos y de gran escala.

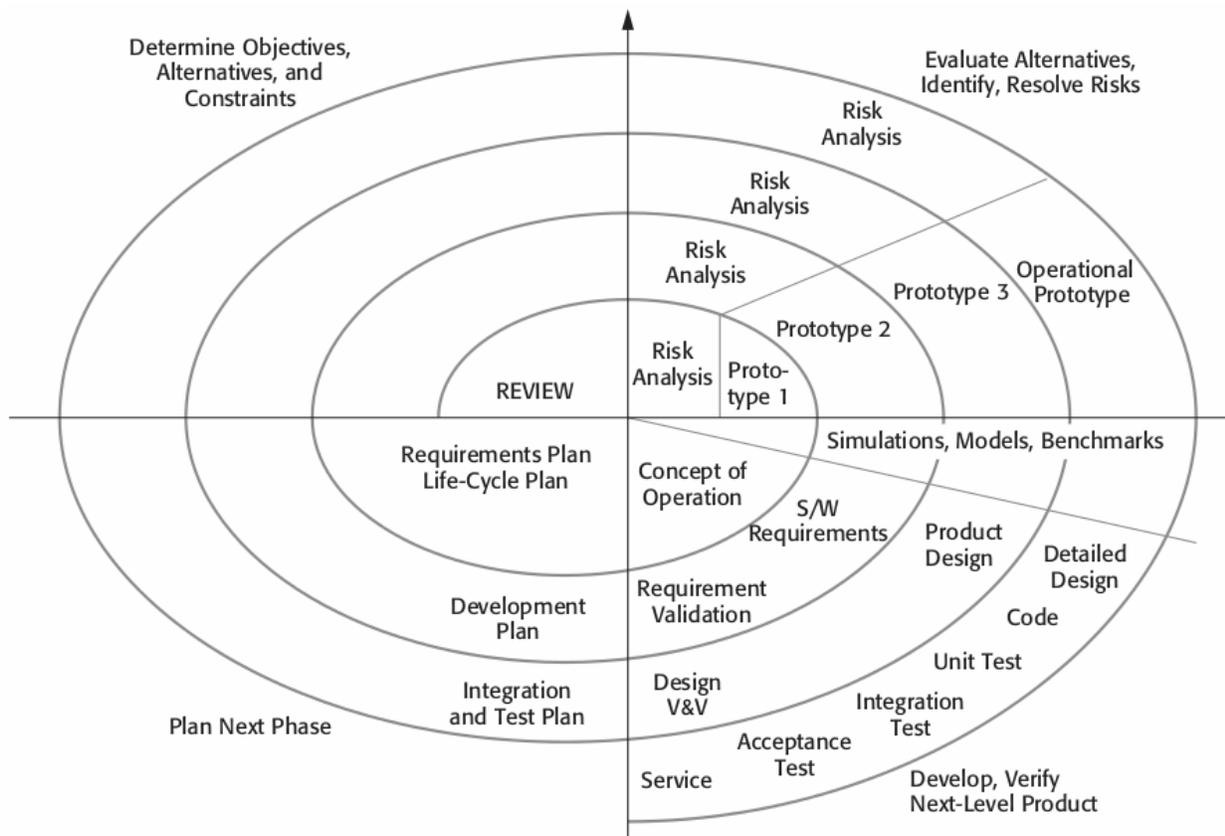


Fig. 2.9. Ejemplo del diagrama del desarrollo en espiral.

El desarrollo del software es representado como una espiral donde cada ciclo representa una fase de un proceso del desarrollo. Cada ciclo se divide en 4 partes:

1. El establecimiento de objetivos. Se definen los objetivos para la fase que se hará en el ciclo que está por iniciar. Se identifican las restricciones del proceso y del producto así como los riesgos. Se hace la planeación de desarrollo así como rutas alternas en función de los diferentes riesgos identificados.
2. Valoración y reducción de riesgos. Se evalúa cada riesgo identificado y se toman medidas para reducirlos.
3. Desarrollo y validación. Después de la valoración de riesgos se toma una metodología de trabajo para dicha fase.
4. Planeación. Se hace una revisión del proceso y se decide si se continúa con el siguiente ciclo. Si se decide avanzar, se hace la planificación del siguiente paso.

2.4.6 Desarrollo de juegos

Para el desarrollo de videojuegos usualmente no se usan los modelos tradicionales de desarrollo de ingeniería de software ya que este tipo de software involucra muchas otras áreas y disciplinas. Sin embargo, juegos desarrollados con una mala metodología de desarrollo terminan pasándose del presupuesto y del tiempo estimado de entrega y con una cantidad enorme de bugs.

Los videojuegos difieren de otros tipos de software en que lo importante del software para empresas o para uso comercial es el funcionamiento, que se encuentre bien estructurado y que haga lo que el usuario final requiere; mientras que los videojuegos buscan que el usuario invierta tiempo en entretenimiento, lo más importante es la experiencia que se lleva el jugador. Además, los juegos no se ven como software, sino que se engloban con otros medios como las películas, la televisión y hasta los libros. Lo que diferencia a los juegos de ellos es la interactividad del usuario con el medio. El jugador es colocado en un mundo donde sus acciones tienen consecuencias, aunque sean menores, dentro del mundo y tiene una retroalimentación a algunos o todos sus sentidos; el jugador es un actor más.

En el desarrollo de un juego se puede usar el Desarrollo Ágil, que es una colección de métodos en la que los requerimientos pueden ir cambiando, y se espera que cambien, durante todo el desarrollo del proyecto. Esta metodología funciona porque al inicio del proyecto casi siempre no se tiene una visión clara de toda la lista de requerimientos. Una de las metodologías ágiles más comunes y populares es la metodología Scrum.

El Scrum se trata de un marco de desarrollo iterativo e incremental, lo que quiere decir que se trata de un proceso que se repite y aumenta el producto con cada repetición. Schwaber (2004) habla de eso como el esqueleto y corazón de Scrum. Durante una iteración el equipo revisa lo que debe hacerse, lo que se ha hecho y en dónde hay oportunidades para realizar incrementos antes de empezar la nueva iteración. Schwaber dice que este proceso se repite hasta que se terminen los recursos del proyecto.

Otro método de trabajo usado es el Proceso Personal de Software (PSP por sus siglas en inglés) que requiere entrenamiento del equipo de desarrollo para que todos se encuentren en un mismo canal durante el desarrollo del juego. Este método requiere compromiso de todos los involucrados para que funcione bien.

Usualmente la creación de un videojuego requiere la combinación de varias metodologías para las diferentes áreas. En la programación se puede usar una metodología en cascada mientras que el diseño de juego puede hacer uso de prototipos. En general los juegos pasan por los siguientes procesos:

1. **Pre-producción o fase de diseño.** Es la fase de planeación en donde se enfoca en la idea o concepto principal y se generan los primeros documentos de diseño. También se establece un primer plan de las tareas, horarios y tiempo estimado de desarrollo.
2. **Documento de diseño de juego.** Este documento describe el concepto general del juego y los elementos del modo de juego. También puede contener sketches que ayuden con la descripción del concepto, escenarios y personajes, y un prototipo inicial del juego. El documento sirve como guía durante toda la vida del proyecto y puede tener varias modificaciones y cambios conforme se vaya necesitando.
3. **Producción.** Es cuando se desarrollan los elementos necesarios, como gráficos, audio y video, además del código fuente.
4. **Pruebas.** La fase de pruebas entra una vez que ya se encuentra desarrollado algo que ya sea jugable. En las primeras pruebas se encuentran los primeros errores mayores en el código y los errores de diseño en el gameplay o cambios en las mecánicas del juego. Lo cambios necesarios se hacen ya sea por el equipo de programadores o por el diseñador del juego, dependiendo de los problemas encontrados por equipo de pruebas.
5. **Post-producción.** Una vez terminado el desarrollo, el proyecto pasa a la fase de post-producción o de mantenimiento, donde puede que el equipo de programadores sea

requerido para corregir bugs y conflictos en el juego que surgen después de la publicación del juego.

Esta es una forma general de desarrollar un juego, depende mucho del personal que se tiene para las diferentes áreas necesarias. Durante el desarrollo del juego de este trabajo se siguió una metodología ágil en donde se realizó una entrevista con las terapistas para generar una idea general del concepto del juego. Posterior a eso se dio un tiempo para la planeación y durante el desarrollo hubo diversas entrevistas para ajustar los requisitos, el diseño y que las terapistas tuvieran una noción completa de cómo iba quedando el juego. Como ya se ha mencionado, el desarrollo de videojuegos involucra muchas disciplinas. En el capítulo 3 se encuentra todo lo mencionado con mayor detalle.

En un desarrollo ágil se manejan diferentes roles con una tarea especial para entregar metas establecidas, llamadas *sprints*, en un tiempo determinado. Los roles que se pueden encontrar normalmente en un desarrollo ágil son: el líder de equipo de desarrollo, los miembros del equipo, los dueños del producto y las partes interesadas. Siguiendo con la forma de trabajo descrita en el párrafo anterior, el rol de las terapistas del lenguaje eran “dueñas del producto”, ya que ellas representaban a las “partes interesadas” (los niños en tratamiento, otras terapistas, directivos del Instituto, etc.) y brindaban información al equipo de desarrollo. La ingeniera Rosa Elena Lobera Sánchez era nuestro líder del equipo, a ella reportaba mis actividades y coordinó las entrevistas y pruebas con los niños. Por último yo fungí como miembro del equipo quien tenía la responsabilidad de programar el juego según lo acordado en las juntas.

Dependiendo de la complejidad o *gameplay*, la programación de un juego puede involucrar muchas áreas:

- Física. Es la programación de todas las características físicas de los objetos del mundo virtual que le dan una mayor responsividad a la interacción del jugador con su entorno.
- Inteligencia Artificial. Los personajes del mundo virtual, tanto neutrales, amistosos o enemigos, requieren algún grado de inteligencia para desempeñar sus papeles dentro del juego. Los conceptos de la inteligencia artificial se usan para darles estos comportamientos que pueden ser muy complejos, como toma de decisiones o en extremo sencillos, como seguir el camino trazado por el jugador.
- Gráficos. El área de computación gráfica es muy importante en el desarrollo de un juego, sin importar si requiere gráficos potentes o un simple movimiento de imágenes. Se requiere de la optimización de modelos 3D o de las mismas imágenes para evitar que el juego sea lento o muy pesado.
- Sonido. La integración de sonidos también tiene su grado de dificultad, sobre todo si se desea meter sonido 3D realista para aumentar la inmersión del jugador. Los programadores de audio deben controlar los sonidos de acuerdo a las acciones del jugador teniendo cuidado de que se mantengan realistas y no sean molestos.
- *Gameplay*. La jugabilidad o *gameplay* es la colección de las distintas mecánicas que existen dentro del juego y que puede realizar el usuario dentro del mundo virtual.
- Interfaz. La interfaz es la cara que se presenta al jugador durante todo momento, desde los menús hasta información relevante al usuario sobre el estado en el que se encuentra: vida, puntaje, balas, objetivos, nivel y todo lo que pueda tener algún grado de importancia para el jugador.
- *Render*. Es el proceso de generar una imagen o video partiendo de un modelo 3D o algo definido en un código.

Existen otras áreas relevantes a la programación de un juego pero las anteriores pueden verse como las básicas o las más comunes. Todas involucran diferentes habilidades o especializaciones de programación así que normalmente se junta a un grupo grande de programadores para realizar el trabajo.

2.4.7 Herramientas de desarrollo

Los programadores cuentan con diversas tecnologías, metodologías y herramientas para la programación de videojuegos. Antes de elegir cuál herramienta se utilizará, deben ya tener bien definido el diseño inicial del juego, además de saber para qué plataforma se lanzará el juego, los requerimientos de la velocidad de ejecución y el lenguaje que usan otras herramientas.

Se elige el IDE, Entorno de Desarrollo Integrado, que es un software que sirve para la programación, depuración y desarrollo en general. Después se elige el lenguaje de programación a emplearse. El más común es el Lenguaje C++, que se trata de un lenguaje que emplea el paradigma de programación orientado a objetos, tiene gran compatibilidad con varios sistemas y sus tiempos de ejecución son muy rápidos. Otros lenguajes como Java o C son también empleados pero muchas veces no son tan aptos para varios proyectos ya que usan otros paradigmas de programación o su tiempo de ejecución es muy lento. El paradigma orientado a objetos es la forma de programación que usa el concepto de "objetos", que se tratan de estructuras de datos que contienen información, conocidos como atributos del objeto, y ciertos procedimientos conocidos como métodos. Los programas en el paradigma orientado a objetos se diseñan creando objetos que interactúan entre ellos. Esta idea de este paradigma es la que la hace ideal para la programación de juegos.

Esto no descarta a todos los lenguajes de programación ya que varios pueden ser empleados para otras áreas de programación. Muchas veces los lenguajes de scripts son usados en tareas más simples que no siempre se dejan a lenguajes como C++. Lenguajes como Python o Lua son usados para este propósito. Y para la creación de efectos avanzados de gráficos, llamados Shaders, como efectos de sombras, luces o más detalle en los objetos; se usan otros lenguajes como GLSL.

Existen librerías que hacen posible el render como OpenGL. Ésta permite el render de gráficos en 2D y 3D, y se puede usar la Unidad Procesamiento Gráfico (GPU). OpenGL fue creado por Silicon Graphics Inc y actualmente es mantenido por Khronos Group. y actualmente se usa en diferentes campos de la computación como el Desarrollo Asistido por Computadora (CAD), la Realidad Virtual, Visualización de Información Científica, simuladores y videojuegos. Está formado por una colección de funciones y valores constantes que son usados por un lenguaje de programación que quiera hacer uso de renders. OpenGL es independiente del lenguaje, lo que quiere decir que puede ser usado con cualquier lenguaje de programación, aunque sus funciones estén escritas en un formato similar al lenguaje C. Las implementaciones más comunes son con los lenguajes JavaScript (WebGL), con C (WGL, GLX, CGL), C para iPhone, y C con Java para Android. Además de ser independiente del lenguaje, es independiente de la plataforma, lo que significa que puede ser usado para diferentes sistemas operativos o hardware.

2.4.8 Game engines

Los game engines son programas diseñados especialmente para la fabricación de videojuegos. Son empleados para crear juegos para consolas de videojuegos, teléfonos móviles o para computadoras. Los *game engines* facilitan varias áreas del desarrollo ya que en su

mayoría proveen la funcionalidad para el rendering de gráficos en 2D o 3D, para la física del juego o detección de colisiones entre los objetos, el sonido, animación, inteligencia artificial, etc.

Varias compañías tienen sus propios engines de juego para facilitar su desarrollo y proteger sus proyectos, sin embargo la desventaja es que, al tratarse de un software propietario de la empresa, sus empleados deben capacitarse en el uso de dicho programa, lo cual implica tiempo que se debe invertir en la enseñanza del uso de dichas herramientas.

Los *engines*: proveen una interfaz gráfica para el desarrollo de juegos, con pequeños componentes de software reusable (Ward, 2008). En algunos casos, estos componentes se agregan a un IDE para habilitar una forma simplificada y rápida de construir un juego. El empleo de game engines reduce el costo, complejidad y tiempos de desarrollo y puede ser entregado a varias plataformas como consolas de juegos o computadoras sin la necesidad de hacer tantos cambios al código fuente. La arquitectura de diseño de algunos *engines* es usando componentes, los cuales pueden ser reemplazados o extendidos por otros componentes más especializados; por ejemplo, *Havok* es un componente especializado en la física que puede ser usado como componente de un *game engine*.

Los componentes que pueden formar parte de un *game engine* son:

- **Programa principal de juego.** Es toda la lógica del juego, que se distingue de otros componentes como el gráfico o el sonido.
- **Render engine.** Es el componente que se encarga del render, dependiendo del engine puede dar soporte sólo para el uso amplio de la Unidad de Procesamiento Gráfico (GPU), conocida también como la tarjeta de video.
- **Engine de audio.** Es un componente que contiene todos los algoritmos dedicados al sonido. Puede ejecutar los cálculos directamente en el CPU o en un Circuito Integrado especial.
- **Engine de física.** Es la parte responsable de darle al juego la sensación de la aplicación de las leyes de la física.

Las últimas tendencias de los game engines es adentrarse al mundo de las diferentes plataformas como los teléfonos móviles con Android o iOS; o que tengan la habilidad de exportar juegos que puedan correr en navegadores web. También se generan más engines que usan lenguajes de alto nivel como C#, Java o Python para que sea más fácil el desarrollo de juegos.

2.4.9 Gráficos en computación

Existen diferentes técnicas para mostrarle al jugador el ambiente en el que se encuentra o una retroalimentación de sus acciones mediante una pantalla. Saber qué técnica usar está en función de la experiencia que se quiere brindar, las limitaciones del hardware en el cual se ejecutará el juego y que existan los recursos para poder crear los gráficos deseados.

a. Basado en texto

Una de las primeras formas para mostrar gráficos en pantalla era a base de texto. Los primeros juegos usaban representaciones en texto en lugar de gráficos a los que estamos actualizados; el texto podía ser una breve descripción del entorno en el que se encontraba el jugador o una abstracción de un área en 2D usando diferentes tipos de caracteres. Juegos desde 1978 hacen uso de esta técnica siendo los más famosos los llamados Calabozos Multiusuario

(MUD por sus siglas en inglés, *Multi-User Dungeon*). Los MUD son una evolución de los juegos de rol donde el jugador recibía una descripción textual de su entorno e ingresaba acciones mediante comandos similares al lenguaje natural (Fig. 2.10). Son Multiusuario porque se conectan varios jugadores de manera remota a jugar el mismo juego; pueden leer descripciones de las habitaciones, objetos, otros jugadores, personajes virtuales y de las acciones realizadas. Mezclan elementos de los juegos de rol, ficción interactiva y chat online.

```

Telnet british-legends.com
*n
Path.
You are standing on a path which leads off a road to the north, to a cottage
south of you. To the west and east are separate gardens.
*w
Flower garden.
You are in a well-kept garden. There is an unexpectedly sweet smell here, and
you notice lots of flowers. To the east across a path there is more garden.
*w
Cliff.
You are standing on the edge of a cliff surrounded by forest to the north and
a river to the south. A chill wind blows up the unclimbable and unscalded
heights. At the base of the cliff you can just make out the shapes of jagged
rocks.
*w
As you approach the edge of the cliff the rock starts to crumble. Hurriedly,
you retreat as you feel the ground begin to give way under your feet!
*leap
You are splattered over a very large area, or at least most of you
is. The rest of your remains are, even now, being eaten by the seagulls
(especially your eyes). If you'd have looked properly before you leaped you
might have decided not to jump!
Persona updated.
Would you like to play again?
:

```

Fig. 2.10. "MUD screenshot" by Source. Licensed under Fair use via Wikipedia - http://en.wikipedia.org/wiki/File:MUD1_screenshot.gif#/media/File:MUD1_screenshot.gif

No todos los juegos basados en texto tienen que ser multiusuario, un ejemplo es Zork. Fue creado entre 1977 y 1979 por Tim Anderson, Marc Blank, Bruce Daniels y Dave Lebling; que eran miembros de Grupo de Modelado Dinámico del Instituto de Tecnología de Massachusetts. Zork se hizo famoso por su extenso contenido, tanto en historia como en el tipo de comandos que podían ser ingresados ya que no sólo reconocía enunciados simples de dos palabras sino que también enunciados con algunas preposiciones y conjunciones.

Por último, los primeros juegos del género conocido como *Roguelike*, que se caracterizan por calabozos generados de manera aleatoria, muerte permanente y por tener movimientos por turnos, representaban sus escenarios con caracteres especiales del ASCII. El juego *Rogue* (Fig. 2.11), de donde nació el término del género *Roguelike*, fue el que popularizó su estilo de jugabilidad. El objetivo de *Rogue* es adentrarse en un calabozo donde el jugador tenía que llegar al nivel más bajo de dicho calabozo, obtener un amuleto y salir por donde entró; fue desarrollado por Michael Toy y Glen Wichman en 1980. Su versión original usaba letras para representar todo, incluidos a los personajes.

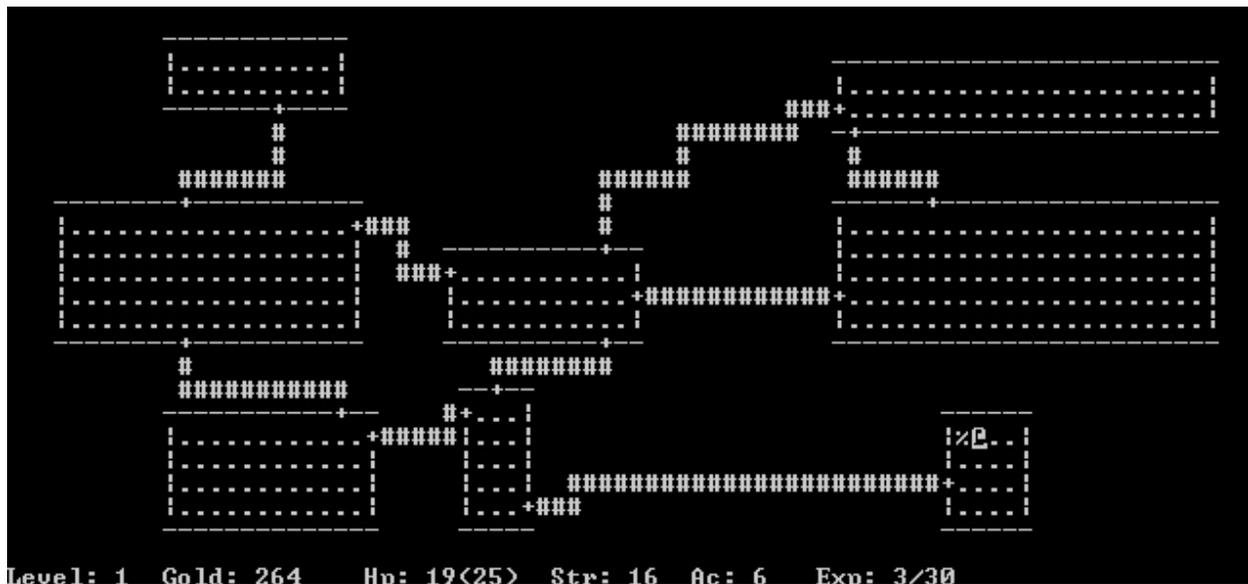


Fig. 2.11. "Rogue Unix Screenshot CAR". Licensed under Public Domain via Wikipedia - http://en.wikipedia.org/wiki/File:Rogue_Unix_Screenshot_CAR.PNG#/media/File:Rogue_Unix_Screenshot_CAR.PNG

b. Mapa de bits

También conocidos como imágenes ráster (*raster images*), es una matriz de puntos estructurada, normalmente en forma rectangular, que representan una cuadrícula de píxeles o puntos de color a ser desplegados en cualquier dispositivo de salida. Un mapa de bits se caracteriza por tener un ancho y una altura medida en píxeles y por un número de bits por píxel, conocido como la profundidad del color, el cual determina la cantidad de colores que puede representar.

La forma de representación en mapa de bits se usa en los monitores para desplegar la salida, donde cada píxel en pantalla corresponde a una localidad de memoria. La pantalla se dibuja por un proceso de escaneo a través de cada píxel y los colorea de acuerdo a los valores encontrados en cada localidad de memoria.

La forma más común de almacenamiento de imágenes es en mapa de bits, como lo son los formatos BMP, JPEG y PNG. Las imágenes dependen de la resolución, lo que quiere decir que no pueden ser escaladas a cualquier dimensión sin que exista una pérdida en la calidad de la imagen. Sin embargo este tipo de imágenes son la mejor opción para las fotografías, impresiones e imágenes fotorrealistas.

Éste es el tipo de graficación más usado en juegos de tipo 2D. De ella se derivan tipos de animación como los Sprites (Fig. 2.12), que son una colección de imágenes que forman una animación cuadro a cuadro. Normalmente las imágenes que forman la o las animaciones se juntan en una sola imagen llamada Atlas. Programáticamente, se eligen cada una de las imágenes y se reemplazan cuadro a cuadro para dar la ilusión de animación.

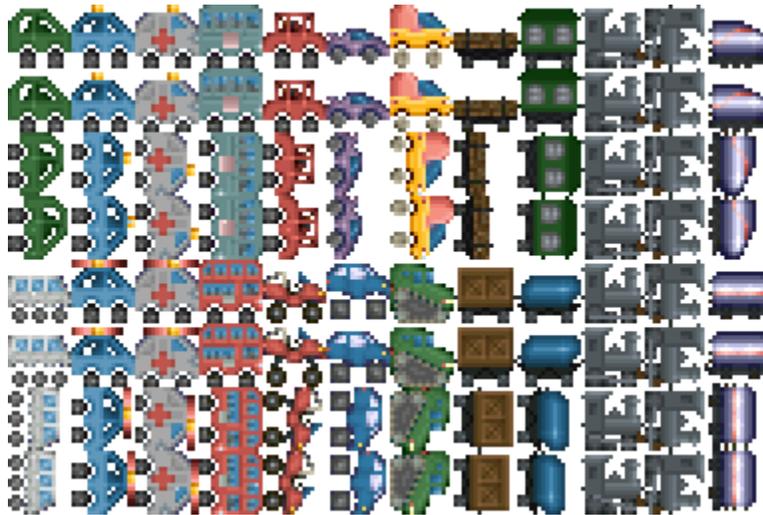


Fig. 2.12. "Sprite-example" by Kriplozoik - Own work. Licensed under CC BY 3.0 via Wikimedia Commons - <http://commons.wikimedia.org/wiki/File:Sprite-example.gif#/media/File:Sprite-example.gif>

c. Gráficos vectorial

Realizar gráficos de manera vectorial quiere decir hacer uso de expresiones matemáticas para crear figuras primitivas como puntos, líneas, círculos, etc. Las imágenes vectoriales pueden ser aumentadas en tamaño, o magnificadas, infinitamente sin que la imagen sufra de pixelación, es decir sin que se distingan los píxeles al aumentar la escala.

La herramienta representativa del uso de este tipo de graficación es Adobe Flash, un software para realizar animaciones y juegos. Los productos de Flash normalmente se encuentran en páginas para ser vistos o jugados en un navegador de internet. El formato de imagen vectorial que es el estándar del *World Wide Web Consortium* es el *Scalable Vector Graphics* (SVG). El uso de expresiones matemáticas para dibujar reduce el tamaño de los datos necesarios para guardar una imagen, por ejemplo si se crea un círculo de radio r en una imagen, la información que requiere guardar es:

1. Un indicador de que el dibujo se trata de un círculo.
2. El radio r .
3. La localización del punto central del círculo.
4. El estilo y color de la línea del contorno que bien puede ser inexistente.
5. El color y estilo del relleno, que puede ser transparente.

Las ventajas que trae el dibujo vectorial son:

- La información mínima para guardar un archivo vectorial es mucho menor que el de una imagen normal.
- Se puede agrandar la imagen una infinidad de veces y ésta no presentará ninguna distorsión y se mantendrá suave.
- Los parámetros de los objetos que forman la imagen se almacenan y pueden ser modificadas posteriormente, esto también quiere decir que si se escala, se mueve o gira los objetos la imagen no perderá calidad.
- El dibujo vectorial se puede usar en escenarios 3D para dibujar una sombra más realista de los objetos.

Por supuesto, que el dibujo vectorial trae consigo algunas desventajas en comparación con los mapas de bits:

- Cámaras y escáneres producen imágenes en mapas de bits que sería muy impráctico convertir a gráficos vectoriales.
- Algunas imágenes pueden tener curvas muy complejas que se vuelven muy difíciles de reproducir.
- Es muy complicado realizar imágenes con mucho detalle y realismo.

d. *Full Motion Video (FMV)*

Otra técnica de graficación para los videojuegos es el *Full Motion Video (FMV)* o Videos de Movimiento Completo. Este tipo de técnica hace uso de videos pregrabados para ser reproducidos durante el juego. Los FMVs son usados en muchos juegos para presentar información o escenas que avanzan la historia del juego, aunque existe una rama especial de juegos que son completamente creados usando FMVs.

Aunque los FMVs logran una mejor apariencia en cuanto a calidad de gráficos, hubo un descontento con los juegos que usaban esta técnica para todo ya que los jugadores se quejaban que no existía tanta interactividad, parecía más una película interactiva que un videojuego. Por esta razón, y por el desarrollo de mejores tecnología y técnicas de graficación, este tipo de juegos decayó rápidamente.

e. Gráficos 3D

Con la adición de las aceleradoras gráficas o tarjetas de video, se pudo pasar de los gráficos en 2D a representaciones más complejas y realistas. Usando representaciones geométricas en tres dimensiones, se realizan cálculos, un proceso conocido como *renderización*, que genera imágenes 2D para ser desplegadas en pantalla. Dichas imágenes se pueden almacenar para su futura revisión o, para el caso de los videojuegos, se generan y se despliegan en tiempo real.

La representación de imágenes 3D en computación hace uso de algoritmos de vectores para el modelado de las figuras geométricas en 3D, y de mapa de bits para el proceso de renderización. Objetos 3D pasan por un proceso de modelado, que generan archivos que guardan la información del objeto. Este objeto puede ser usado después por otros programas para colocarlo en lugares especiales o modificarlo, para después ser renderizado a la pantalla.

Los gráficos 3D pasan por 3 pasos en general:

1. **Modelado 3D.** Es el proceso de formar un modelo de un objeto en particular en la computadora.
2. **Plan y animación.** El movimiento y localización de los objetos 3D en una escena.
3. **Render 3D.** Cálculos realizados por la computadora para generar una imagen en pantalla. Se realizan varios algoritmos para la iluminación de la escena o efectos especiales de los objetos.

El modelado se puede hacer de dos formas: utilizando una herramienta de modelado 3D o escaneando objetos de la vida real a una computadora. Los modelos son una colección de vértices que definen la forma general y se conectan para formar polígonos. Los polígonos son

superficies formadas por la conexión entre 3 o más vértices. Una vez que se tienen los modelos, se deben colocar en la escena y establecer una relación espacial entre ellos, es decir su escala y posición relativa entre ellos. La animación es el movimiento y deformación de los objetos a través del tiempo. Por último, el render convierte los modelos en imágenes para obtener imágenes fotorrealistas o, si se desea, obtener un estilo diferente usando técnicas de render no fotorrealistas. Las dos operaciones básicas de cualquier render son la iluminación, llamada el transporte de la luz, y la reacción de las superficies de los modelos a la luz, llamada la dispersión. El render requiere de una proyección 3D para representar una escena en 3D en una imagen de dos dimensiones.

Todo el proceso de graficación en 3D, sobre todo el render, puede tomar mucho tiempo para representar una sola imagen. El tiempo de procesamiento depende del número de polígonos de los modelos, la cantidad de modelos en la escena, los cálculos de la luz, deformaciones o reflexiones de la luz que puedan generarse por la superficie de algunos objetos, entre otras. En el caso de imágenes fotorrealistas para representar un edificio terminado o para ser usado en revistas, efectos especiales de una película o la creación de una película animada; el proceso de render se realiza una única ocasión para generar el producto final, lo cual permite usar técnicas de render más complejas y con efectos muy realistas aunque tarde mucho tiempo en generarse la proyección. Sin embargo, en el caso de los videojuegos, el proceso del render debe realizarse en tiempo real, por lo que la proyección de escenas 3D a imágenes en pantalla debe ser muy rápido para que el usuario sienta que se encuentra en un mundo ajeno que responde inmediatamente a sus acciones como sucede en nuestro universo.

El render en tiempo real se vale de diferentes técnicas que tienen como fin agilizar el tiempo de proyección a pantalla. Los modelos 3D se crean usando un bajo número de polígonos pero usando pequeños trucos geométricos para darles una mejor apariencia, como el uso de vectores normales a los vértices en los cálculos de iluminación para darle a la superficie de los objetos una apariencia más suave (Fig. 2.13). Otra técnica es la limitación del cálculo del render a los modelos que se verán en la pantalla una vez que se haya realizado la proyección. Procesos más complejos pero que son rápidos y generan efectos visuales muy realistas o impresionantes son los *shaders*, que usan un lenguaje de programación propio y hacen modificaciones durante el proceso del render. Con *shaders* se pueden generar efectos como sombras más suaves, pequeños relieves en muros porosos, efectos de luz, superficies reflejantes, y muchos más (Fig. 2.14).

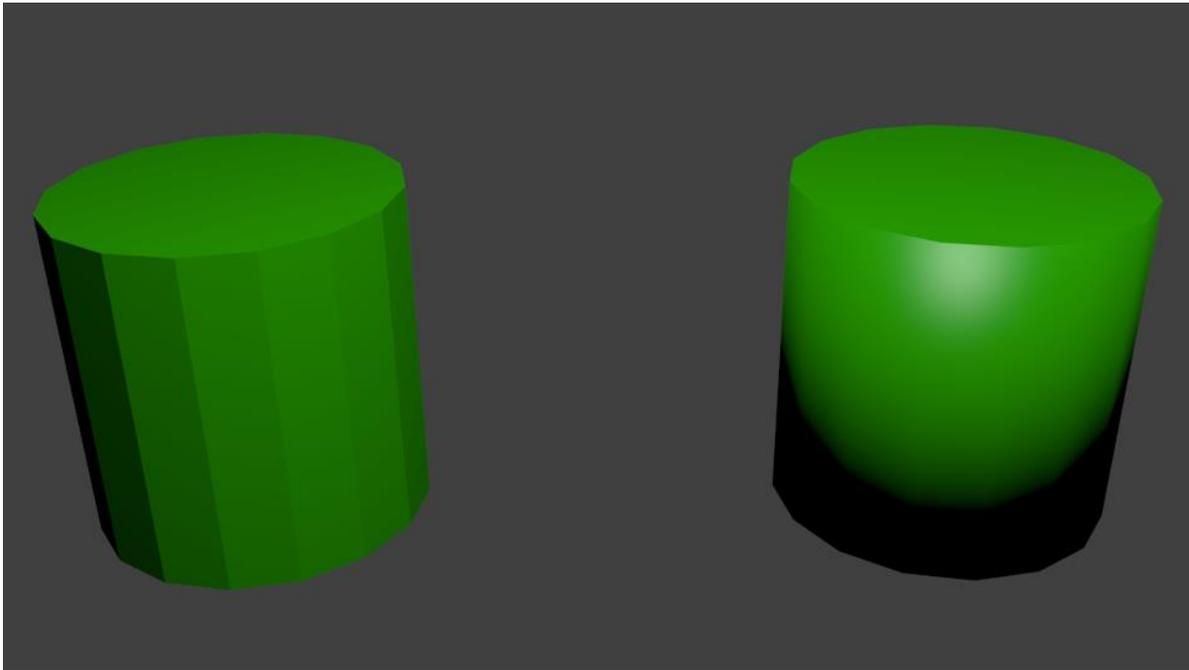


Fig. 2.13. Representación de dos cilindros con el mismo número de polígonos. El de la izquierda tiene un sombreado plano sin usar vectores normales a los vértices. El de la derecha usa un sombreado con vectores normales a los vértices para dar una apariencia más suave. - Render usando Blender 2.66a (Abril 2015).

No todas las ideas de render que existen se pueden aplicar a cualquier juego y esperar que cualquier computadora genere imágenes con grandes efectos y buena calidad en tiempo real, el uso de tarjetas gráficas se requieren para hacer renders más complejos con gráficos muy reales. Los cálculos necesarios para las mejores técnicas de render son demasiados como para que un procesador de computadora pueda realizar sin alentar todos los procesos que realiza comúnmente. Las tarjetas de video liberan al procesador de todos los cálculos relacionados con la proyección de imágenes, así, tener un procesador dedicado al proceso de render nos permite generar gráficos más complejos en tiempo real.

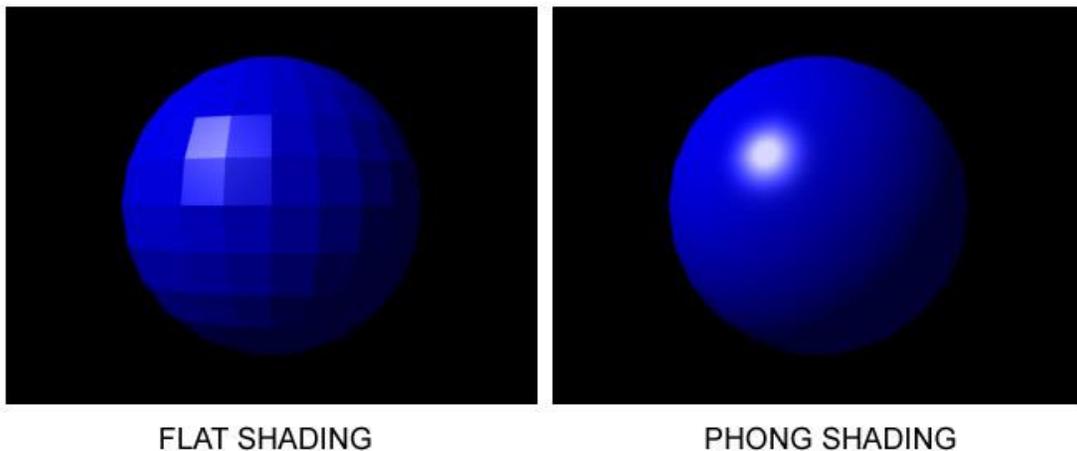


Fig. 2.14. Ejemplo de Shader llamado Phong. "Phong-shading-sample". Licensed under Public Domain via Wikimedia Commons - <http://commons.wikimedia.org/wiki/File:Phong-shading-sample.jpg#/media/File:Phong-shading-sample.jpg>

“No dejamos de jugar porque envejecemos; envejecemos porque dejamos de jugar”

-George Bernard Shaw

Capítulo 3 - Creación de Fantasmagia

Ya que el objetivo final del programa es ayudar a niños en rehabilitación, de inmediato surge como un mejor candidato cualquier software de entretenimiento para entregar un programa que capte la atención de los niños. El software tiene como base la idea de un juego de video ya que los videojuegos han demostrado ser motivadores para todo tipo de actividades; desde hacer ejercicio hasta la educación. Muy a menudo los niños en rehabilitación de lenguaje pueden mostrarse distraídos o con falta de interés en los ejercicios de terapia; es por eso que el videojuego pretende atraer la atención al ejercicio de terapia para que mientras el niño trabaja con los ejercicios de lenguaje, juegue y se divierta. Este software busca demostrar que el uso de videojuegos en el área de terapia ayuda en la rehabilitación y tratamiento de los pacientes; se busca enriquecer la selección de material para la terapia del lenguaje que sea útil para el terapeuta como para su paciente.

El programa que realizamos se hizo un videojuego porque su uso está enfocado en niños de entre 6 y 10 años; les llama mucho la atención este tipo de software y captura su atención de inmediato, agregado a eso es los resultados que han brindado el uso de juegos interactivos de computadora y otros dispositivos como apoyo en diferentes áreas de trabajo como se explica en este tema.

3.1 Desarrollo

El proyecto se realizó con la idea original de emplear tecnologías de la información en el área de terapia de comunicación humana del Instituto Nacional de Rehabilitación (INR). Los terapeutas de dicha área están buscando la adición de nuevas técnicas y tecnologías a su área de trabajo para mejorar la efectividad de sus sesiones.

Como parte de un servicio social, estudiantes de la Facultad de Ingeniería de la UNAM son invitados a realizar trabajos en conjunto con los terapeutas. A nosotros se nos invitó con la idea de querer desarrollar un software de entretenimiento para apoyar las sesiones con niños.

3.1.1 Definición del problema

Los niños de hoy en día están muy acostumbrados a tener un alto contacto con dispositivos móviles o computadoras, les llama mucho la atención y son parte de su vida cotidiana. Esto no es necesariamente algo malo, se puede emplear en áreas que donde normalmente es extraño ver el uso de este tipo de dispositivos para aumentar el interés y

participación de niños. El despliegue de las nuevas tecnologías a diferentes áreas no es una tarea simple, requiere de trabajo en conjunto entre ingenieros y los especialistas de las partes interesadas en adoptar nuevas herramientas de trabajo.

Las terapeutas del INR recibieron con buena cara la idea de emplear software de entretenimiento para ser usado como ejercicio de terapia con sus pacientes. Al tratar con niños en su día a día, ya cuentan con varios juegos con material didáctico que muchas veces era elaborado por los mismos terapeutas. En ocasiones, es difícil conseguir que los niños se concentren en la tarea que se les asignan y se dificulta cuando se trata con una cantidad mayor de niños.

En el INR están conscientes del interés de las nuevas generaciones en los nuevos dispositivos y nuevas piezas de tecnología, es por eso que buscan traerlos a su campo de trabajo. Desgraciadamente, en el INR no contaban con equipos de cómputo con los recursos necesarios para ejecutar software con alta demanda de gráficos.

3.1.2 Objetivo del proyecto

El juego tiene como objetivo general ser una herramienta de apoyo del terapeuta para la rehabilitación de niños entre 8 y 10 años de edad con problemas de lenguaje a nivel semántico. Se trata de un software de entretenimiento que pueda captar la atención del niño en ejercicios que ayuden en su rehabilitación en la clasificación de palabras por sus características físicas. Debe lograr que el niño encuentre diversión en sus sesiones con las terapeutas, aumente su desempeño y hasta reduzca el tiempo que debe invertir en su rehabilitación.

Se busca fomentar el uso de técnicas nuevas usando tecnologías de la información en diferentes áreas; medicina en este caso. Medir los beneficios, si los hay, del uso de juegos serios en la terapia de Comunicación Humana.

3.1.3 Metodología de trabajo

La forma de trabajo empleada para el desarrollo del proyecto fue usando una metodología de desarrollo rápido de aplicaciones (RAD). Muchos videojuegos son creados siguiendo RADs, mezclados con el prototipado; lo importante es crear algo y tener a los usuarios cerca para que evalúen el trabajo y poder actualizar los requerimientos según lo vaya requiriendo el desarrollo.

Como se partió de un ejercicio lúdico que ya tenían implementado como ejercicio de terapia, la planeación y diseño se balanceó más hacia el lado ingenieril. Se tomó un videojuego como proyecto porque es un software que ha probado tener buenos resultados en lo que necesitaban las terapeutas: una herramienta que incremente la atención e interés de los pacientes en sus sesiones en el instituto.

El límite de recursos gráficos en los equipos de cómputo en el INR nos sugirió usar herramientas de desarrollo de software que generen gráficos con poca demanda del hardware de las computadoras. No sólo se necesitó emplear software de desarrollo de bajo nivel sino también técnicas de optimización de modelos 3D para que los objetos en pantalla no alentarán el producto final.

3.1.4 Herramientas

El proyecto necesitó producción en muchas áreas, desde programación hasta musicales, a continuación se detallan las diferentes herramientas que se usaron para el desarrollo del software.

Al ser un juego es muy recomendable usar un lenguaje de programación orientado a objetos más una librería adecuada de gráficos. Los objetos son una estructura de datos en forma de atributos y en procedimientos llamados métodos. Los objetos, también llamados instancias, pueden tener varias copias de los mismos en existencia al mismo tiempo, pero con diferentes valores en sus atributos y realizando diferentes procesos. Adicionalmente y como se acaba de mencionar, el equipo de cómputo a disposición de los y las terapistas del área de Comunicación Humana no contaba con los componentes para ejecutar software con alta demanda de gráficos. Las computadoras contaban con el sistema operativo Windows XP con procesadores a 1.7 GHz, 512 MB en memoria RAM y sin ningún tipo de procesador gráfico. Por estas razones escogimos hacerlo en OpenGL 2.1 usando el lenguaje de programación C++, para entregar un software ejecutable en una computadora de escritorio para que pueda ser usado en las terapias grupales con niños y además que pudiera ejecutarse de manera eficiente.

Otras razones por las que se prefirió OpenGL sobre un *game engine* fue la experiencia del equipo de trabajo sobre la misma herramienta. Tampoco se contó con un presupuesto para cubrir alguna licencia para el uso de un *game engine* propietario.

El juego está hecho con gráficos en tercera dimensión para poder crear un modelo de casa muy similar a lo real. Como software de modelado en 3D se optó por Blender, el cual es de distribución libre, además de realizar los gráficos con técnicas de optimización de modelos, los cuales se especifican en el tema 3.5.3, para ayudar en la velocidad de ejecución del software ya que los programas de computación gráfica son muy demandantes de recursos y se debe tener siempre en cuenta una optimización en todas las áreas del programa, desde la programación hasta el diseño de modelos y texturas. Las imágenes y texturas necesarias fueron creadas en GIMP, software de edición de imágenes de distribución libre. Dichos gráficos también requieren ser entregados en dimensiones especiales y teniendo en cuenta la resolución necesaria para no saturar la memoria con tanta información de las imágenes. Y el audio se realizó en el programa Linux MultiMedia Studio (LMMS). Todas las herramientas son de código abierto, lo que permite su uso gratuito y crean producto de alta calidad y funcionalidad como sus contrapartes de código privado.

En este documento sólo se comenta el desarrollo en cuanto a la programación, sobre técnicas de computación gráfica para la creación de gráficos, toma de requerimientos, diseño del programa y pruebas; todo lo extraordinario se omite porque va más allá del objetivo de este escrito.

3.2 Concepto del juego

El concepto del juego nace ante las ganas de querer implementar nuevas tecnologías en el área de rehabilitación del lenguaje. Así surge el proyecto “DISEÑO E IMPLEMENTACIÓN DEL SISTEMA PARA LA REHABILITACIÓN DEL PACIENTE AFÁSICO VÍA INTERNET” creado por la Ing. Rosa Elena Lobera Sánchez cuyo objetivo es invertir el servicio social de estudiantes de ingeniería en proyectos que realmente brinden ese beneficio a la sociedad, en este caso enfocado a los esfuerzos realizados en el Instituto Nacional de Rehabilitación.

Ya que la rehabilitación del lenguaje consta de varias etapas y niveles, como ya se vio, es muy complicado hacer un solo software que sirva para todo nivel. Para definir en qué nivel iba a

trabajar nuestro proyecto recibimos el apoyo y asesoría por parte de terapeutas profesionales en comunicación humana: Licenciada Andrea Hernández Bueno, Licenciada María Iliana Linares Aviña y jefe de servicio el Licenciado Manuel Contreras Ordaz. Ellos nos ayudaron explicando sobre su forma de trabajo y ejercicios para tratar con diferentes deficiencias. También sirvieron de consulta durante cada etapa del desarrollo del software y como coordinadores de las pruebas con niños en el INR.

Durante la toma de requisitos hicimos entrevistas con las terapeutas para mostrar qué clase de software podíamos hacer y conocer más sobre su área. Conocimos varios ejercicios y juegos para terapias en diferentes áreas enfocándonos en el área de lenguaje de donde sacamos una primera idea. En el INR manejan la clasificación de trastornos del lenguaje según el componente del lenguaje alterado; el videojuego se basa en un ejercicio para terapia del lenguaje a nivel semántico, en donde a un grupo de niños se les muestra una colección de objetos de uso cotidiano y se les pide seleccionar uno de acuerdo a un grupo de características dadas por el terapeuta; por ejemplo, si se pidiera un objeto que cumpliera con las características de “suave” y “hecho de tela” el niño acertaría al seleccionar una almohada o un sillón o una camisa o cualquier objeto siempre y cuando cumpla con ambas características. La primera idea del juego constaba de una casa vacía y el objetivo era ir metiendo muebles y objetos a la casa en el orden en que se fueran pidiendo a través de sus características. Mientras más objetos se metieran y se colocaran en su posición correcta la casa se iluminaría y se llenaría de vida.

Tuvimos que diseñar el proyecto y planear su desarrollo, así que pensamos realizar una toma de requisitos, planeación, diseño, desarrollo y una colección de pruebas para poder sacar el software como herramienta para terapias. Ya que había quedado la idea del juego tuvimos 2 semanas para desarrollar un pequeño prototipo para mostrarlo durante el **“1er Curso Internacional y 2do Curso Nacional Manejo Interdisciplinario de los Trastornos de Comunicación Humana”**, en la conferencia **“El Modelo Cadena, una Propuesta para la Rehabilitación del Lenguaje en Niños”** que se realizó en el mismo hospital. Recibió buenas críticas e interés lo que permitió seguir su desarrollo. La etapa de desarrollo constó de programación de pequeños objetivos que inmediatamente hicimos pruebas con los terapeutas y evaluamos cambios y ajustes según los resultados de las pruebas. Con el tiempo las pruebas se realizaron con grupos de niños lo cual impulsó mucho el desarrollo ya que se encontraron más errores y mejoras que poco a poco se le agregaron al proyecto.



Fig. 3.1. Concepto final del juego.

Con el tiempo llegó un punto en el que no podíamos encontrar una conclusión al proyecto para terminar el desarrollo y tomamos la decisión de hacer un cambio en el concepto del juego. El problema fue que el tamaño del proyecto fue mayor al esperado y no se veía un final cercano. Se ideó una nueva temática para el juego, que pudiera usar gran parte de los recursos y código ya creado hasta el momento. La nueva idea se presentó a los terapeutas quienes aprobaron los cambios. El nuevo concepto, que quedó como el definitivo, consta de una casa embrujada de la cual se quieren mudar, sin embargo los fantasmas que habitan la casa quieren que los inquilinos se queden a jugar con ellos; esto lo intentan sacando los muebles del camión de mudanza de vuelta a la casa (Figura 3.1). Los muebles viajan en grupos de 4 de los cuales 3 son muebles falsos y uno es real. Es deber del jugador seguir las pistas, que constan de características del objeto, tomar los muebles correctos y llevarlos de vuelta al camión de mudanza (Figura 3.2).



Fig. 3.2. Muebles y lista de pistas.

El diseño de los entornos, personajes y modelos se hizo pensando en que es un juego enfocado a niños por lo tanto tiene un toque muy infantil, amistoso y divertido (Figura 3.3). El cambio de concepto también sirvió para agregar más entretenimiento al juego, le dio una conclusión al proyecto y un nombre: Fantasmagia. Posterior a eso se hicieron pruebas con un nuevo grupo de niños los cuales respondieron bien y se pudo terminar el juego adecuadamente para su nueva presentación en el 3er Curso **“Manejo Interdisciplinario de los Trastornos de Comunicación Humana”** del INR en la conferencia llamada **“El Uso de la Tecnología, Informática y Comunicación en la Terapia de Lenguaje”**. Fantasmagia se entregó como un juego ejecutable a los terapeutas para su uso y prueba con pacientes en varios grupos.

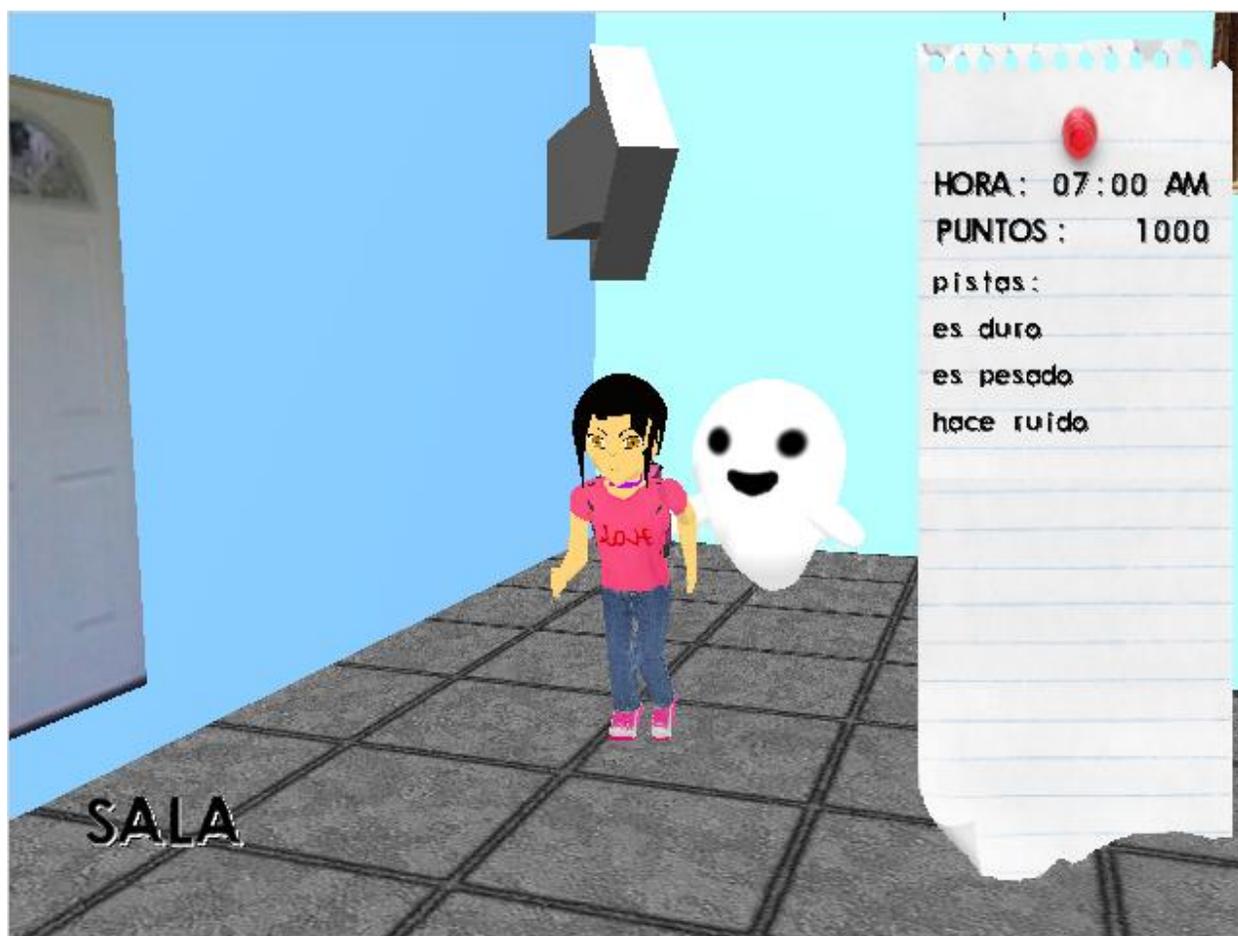


Fig. 3.3. Rediseño de personajes.

3.3 Planeación

En el principio se realizaron reuniones para conocer qué tipo de software deseaban. Se les mostraron ejemplos de juegos que teníamos a la mano, para que se dieran una idea del tipo de programa que se podría elaborar. En la primera reunión, las terapistas querían saber si las ideas que ellas tenían en mente podían ser realizadas por nosotros, y nosotros queríamos saber qué tipo de software requerían. Desde un principio se tenía las ganas de que se creara alguna clase de videojuego, lo que querían era un juego serio para facilitar sus sesiones con los niños.

Durante la segunda reunión conocimos su forma de trabajo con los niños, varios ejercicios: su finalidad y mecánica; todo con el objetivo de generar el concepto del juego que podríamos realizar. Se generó una idea al final de ver su forma de trabajo el cual se pulió y se presentó para una tercera reunión la cual se aceptó.

Inmediatamente se comenzó un diseño rápido y programación de los primeros prototipos. El 1er Curso Internacional y 2do Curso Nacional “**Manejo Interdisciplinario de los Trastornos de Comunicación Humana**” en el Instituto Nacional de Rehabilitación estaba cercano a los días en los cuales se hicieron las primeras reuniones y accedimos a preparar una presentación rápida para mostrarla en una conferencia. Nuestra planeación inicial se basó en el tiempo límite que teníamos para generar un prototipo para la conferencia que mostrara las principales características del juego. Este tiempo inicial también nos serviría para afinar los detalles y requerimientos con los terapeutas a los cuales les entregaríamos avances periódicamente. Posterior a la conferencia seguiríamos el desarrollo de la aplicación haciendo reuniones con terapeutas y niños hasta entregar un producto final.

De manera general, el desarrollo se dividió en dos áreas importantes: el desarrollo de programación y el desarrollo gráfico. Ambas áreas demandan mucho tiempo y es posible hasta cierto grado ir desarrollando cada una por separado. El desarrollo programático consta del desarrollo lógico del programa, es decir escribir las acciones que puede realizar el jugador, los límites del mundo en el que se desarrolla, definir cuándo se reproducen sonidos, aplicar cambios de gráficos, el movimiento de los objetos, personajes secundarios y enemigos, etc. El desarrollo gráfico consta del modelado de los objetos 3D, diseño de sus texturas, el dibujo de fondos, botones y todo lo relacionado con la interfaz gráfica.

3.4 Diseño

Desde el inicio eran claros algunos detalles del diseño en general del proyecto. Como se trata de un videojuego enfocado a niños entre 8 y 10 años el contenido tenía que ser apropiado, con ilustraciones infantiles, lenguaje amigable y con poca complejidad. Posteriormente, los requisitos y cambios durante el desarrollo fueron forjando el juego que los terapeutas querían. A continuación se enlistan diferentes áreas de diseño con sus especificaciones.

3.4.1 Diseño del Gameplay

El *gameplay* de un juego está conformado por el conjunto de mecánicas de juego. Las mecánicas son las pequeñas acciones que se pueden realizar dentro del mundo virtual por parte del jugador. La adición de un número alto de mecánicas puede subir la complejidad del juego, tanto como para el jugador como para los desarrolladores. Al referirse al nivel de complejidad de un juego según sus mecánicas, quiere decir la cantidad de reglas existen en el juego y que tan difícil es comprenderlas, existen juego con alta complejidad pero con mecánicas simples.

La primera mecánica de Fantasmagia es el poder trasladarse dentro del mundo virtual. El jugador se encuentra dentro de una casa, es capaz de moverse entre sus diferentes habitaciones y su patio frontal. El movimiento se introduce mediante el uso de las teclas de direcciones en el teclado, donde la tecla de arriba se usa para avanzar, abajo para retroceder y la derecha e izquierda para girar el personaje a la dirección respectiva.

En el ejercicio de terapia en el que se basó Fantasmagia, el terapeuta preguntaba por un objeto que cumpliera con las dos o más características especiales que pidiera y la tarea del niño era seleccionar un objeto que cumpliera con dichas características. En el videojuego, la mecánica que responde a esta regla es que el avatar del jugador en el mundo de Fantasmagia pueda acercarse a un objeto y pueda “seleccionarlo”. El acto de “selección” de un objeto es que

el personaje del juego tome el objeto como si lo estuviera cargando. La forma en la que el programa responde a la acción de selección es con un “Correcto” o “Incorrecto” dependiendo de si las características del objeto seleccionado concuerdan con la lista pedida.

Cerca de la casa en donde se realiza el juego, hay un camión de mudanzas. Este camión de mudanzas es el lugar desde el cual los objetos que se presentan al jugador se originan; estos se acercan poco a poco a la casa hasta que vuelven a ingresar a ella si es que el jugador no logra interceptar el objeto que cumple con las características que se piden. Dentro de la casa hay un fantasma que persigue al jugador para sacarlo y evitar que tome los muebles de vuelta al camión de mudanzas. En caso de ser atrapado, pueden suceder dos cosas: que el fantasma se ría y desaparezca como si estuviera jugando con el personaje, o que atrape al jugador. Cuando el jugador es atrapado por el fantasma y éste no se encuentra cargando un mueble, el fantasma lo cargará hasta la salida de la casa, pero si lleva cargando un mueble, empezarán a forcejear por él, es decir que el fantasma tomará también el mueble y el jugador no se podrá mover hasta que logre liberarse. En ambos casos cuando el fantasma alcanza al jugador, éste puede escapar del agarre si presiona repetidamente la barra espaciadora. En caso de que el niño sea sacado 3 veces de la casa el juego avanzará al siguiente nivel.

Para ir presentando un mayor reto, la velocidad con la que avanzan los muebles hacia la casa aumenta conforme el niño va completando niveles. Los diferentes niveles se diferencian con el avanzar del día dentro del juego: mañana, tarde y noche. Intermedio entre los niveles, hay un nivel bonus en el cual una fila de diferentes objetos salen del camión a la casa; el objetivo no es el de recolectar todos si no que el niño debe tomar los que tengan una característica en común, los redondos, planos, de color azul, etc.; y llevarlos a una caja que deberá meter posteriormente al camión.

Cada objeto que es correctamente regresado al camión vale puntos que se van acumulando. Después de haber cumplido con todos los niveles, el jugador es presentado con una pantalla final que muestra su puntuación total.

3.4.2 Diseño de la interfaz

Desde un inicio, la interfaz estaba pensada a ser de texto ligero sin tantas opciones y con colores llamativos; la información mostrada durante el juego es simple y clara, para que el niño entienda lo que debe realizar y lo que se le está pidiendo; la fuente de los textos se estandarizó a una muy similar a la que las terapistas usan con los niños para facilitar su lectura y entendimiento.

Durante el juego, es necesario informar al jugador de las características del objeto que se le está pidiendo, el nivel en el que se encuentra, su puntuación y si el objeto que eligió cumple o no con lo requerido; todos estos datos se proporcionan en una barra de información en la parte derecha de la pantalla. Las características de los objetos se muestran en una lista para que el jugador las lea y pueda entonces ver las diferentes opciones que se acercan y evaluar cuál de ellos cumple con lo pedido (Fig. 3.4). Cuando el jugador toma alguno de los objetos, en la barra de información aparecen 3 cosas: una imagen del objeto seleccionado, el nombre del mismo y un texto informando si lo seleccionado cumple o no con todas las características de la lista. El texto en la barra de información es de color negro con un fondo blanco para aumentar el contraste y facilitar la lectura. Los objetos se muestran de nuevo en la barra de información junto con su nombre para que se aprecien los detalles o se identifique mejor lo que acaba de seleccionar.

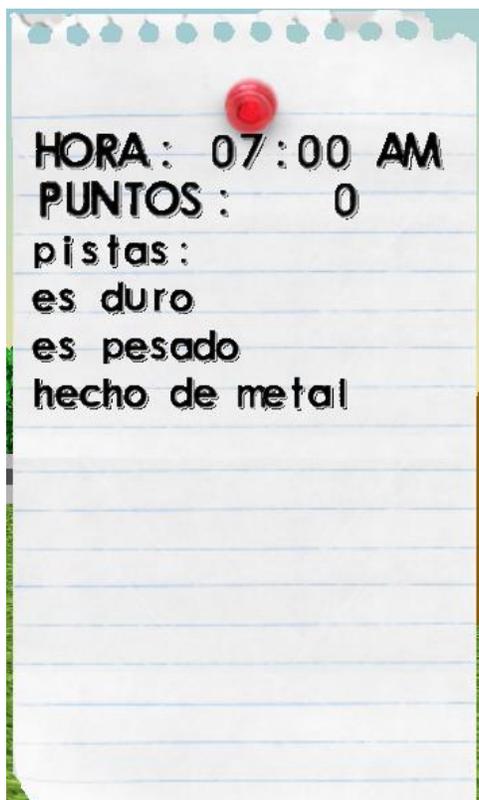


Fig. 3.4. Barra de información

Otra parte de la interfaz es una flecha que señala la ubicación del objeto cuando este logró meterse a la casa (Fig. 3.5). Los objetos pasan del camión de mudanzas a la casa, y todo el tiempo que tardan en recorrer esa distancia el jugador debe identificar la opción correcta para seleccionarlo y llevarlo de vuelta al camión, sin embargo cuando estos objetos llegan a la casa una flecha en la parte superior de la pantalla señalará en qué parte de la casa se encuentra el objeto que debe sacar de regreso a su punto de origen. De igual manera, una vez recuperado el mueble de la casa una flecha indica el camino de regreso al camión de mudanza (Fig. 3.6).



Fig. 3.5. Flecha apuntando hacia el mueble que debe regresar al camión.

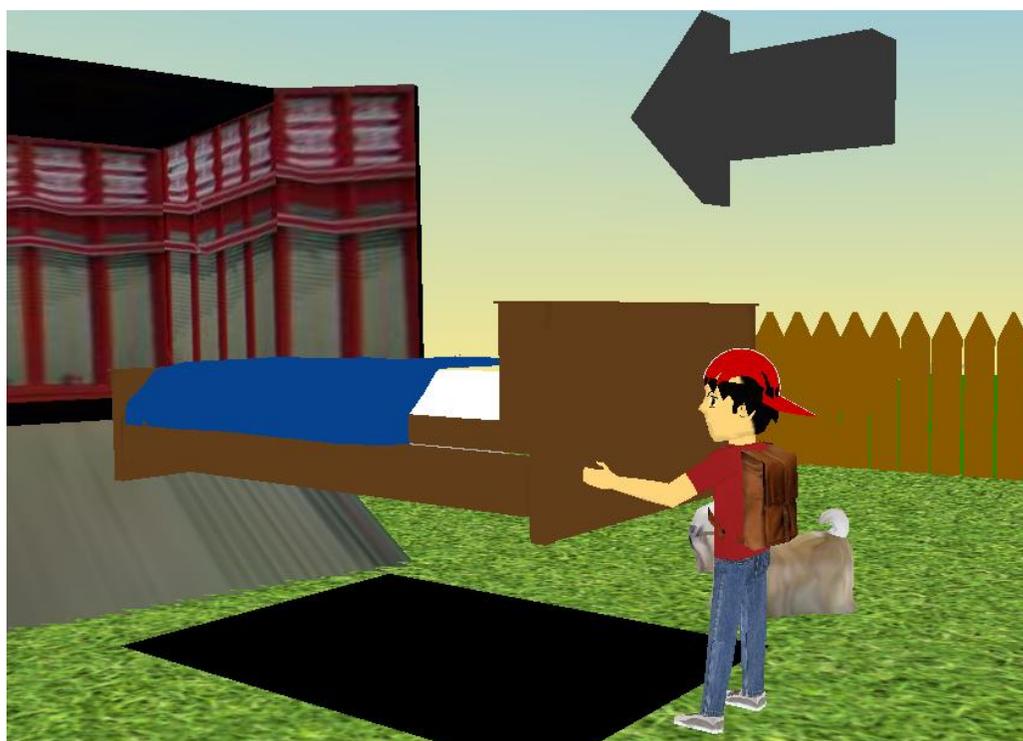


Fig. 3.6. Flecha apuntando hacia el camión.

En la barra de información lateral también se encuentran dos datos sobre el juego: La puntuación y una hora del día, que sirve como indicador del nivel (Fig. 3.7). En el juego existen tres niveles: el día, la tarde y la noche; cada uno indicado por una hora diferente del día además del cambio de iluminación del cielo. Mientras se avanza de niveles, la velocidad con la que los objetos se mueven del camión a la casa aumenta. La puntuación que lleva el jugador se muestra siempre y se actualiza cuando un objeto correcto es ingresado al camión de mudanza. Por último, al concluir el juego, una pantalla final muestra el puntaje total del jugador y anuncia el fin del juego.

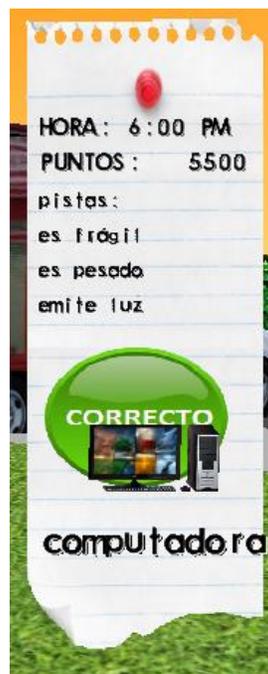


Fig. 3.7. Barra de información completa.

Existen otros elementos gráficos de la interfaz que aparecen de vez en cuando durante el transcurso del juego. Uno es una representación de la tecla de la barra espaciadora; dicha imagen aparece cuando, dentro de la casa, el jugador es atrapado por el fantasma (Fig. 3.8). Lo que debe hacer es apretar la barra espaciadora cinco veces para liberarse, indicado por la interfaz descrita. Por último, durante el nivel bonus, una leyenda aparece en pantalla para explicar el objetivo del nivel bonus (Fig. 3.9).



Fig. 3.8. Interfaz indicando al jugador como escapar del fantasma



Fig. 3.9. Leyenda durante el nivel bonus.

3.4.3 Diseño del nivel

El diseño de niveles de un juego involucra no sólo la creación de los mundos que el jugador debe cruzar sino también todos los objetivos, los obstáculos, la colocación de enemigos o acertijos, el orden en el que van ocurriendo los diferentes eventos, la ubicación de artículos y objetos que el jugador puede recolectar, etc. Cada elemento de un nivel se encuentra especialmente diseñado para darle el reto suficiente al jugador sin que se vuelva demasiado difícil o aburrido; también debe estar en sintonía con la misma experiencia que los demás elementos del videojuego, como la música o la historia, quieren transmitir al jugador para ayudar a su inmersión al mundo virtual; como cualquier otro aspecto de un juego, un nivel bien diseñado tiene como objetivo la experiencia final que el jugador tendrá.

Fantasmagia se compone de un sólo tipo de nivel, la casa con su patio y camión de mudanza es el único mapa que existe, por lo tanto para generar niveles diferentes se hace cambio a los objetivos en cuestión. En los primeros prototipos del juego, la casa era una especie de pasillo largo con divisiones que hacían los diferentes cuartos; la arquitectura era así porque en un inicio Fantasmagia iba a ser un juego en dos dimensiones, sin embargo cuando se hizo el paso a tres dimensiones pronto el diseño del plano de la casa llegó a ser incongruente con el

movimiento y se cambió la proporción y distribución de los cuartos a una forma más común para una vivienda.

Los objetivos de los niveles ya se han descrito algunas veces, se trata de seleccionar un mueble dentro de un conjunto de ellos que salen volando de un camión de mudanza, cruzan el patio frontal de la casa para adentrarse a ella. Para seleccionar el objeto correcto del conjunto se presenta una lista de características del mismo para que a partir de ellas el jugador pueda identificarlo. En caso de que los muebles recorran todo el patio sin que se identifique a tiempo el objeto pedido, estos entrarán a la casa, los muebles falsos desaparecerán mientras que real se traslada a la habitación en donde se puede encontrar generalmente, por ejemplo, la televisión aparecería en la sala, la cama en un cuarto, y el lavamanos en el baño. El obstáculo que se presenta dentro de la casa es un solo fantasma que perseguirá al jugador, si llegara a atraparlo, será sacado de la casa si no trae el mueble consigo o el fantasma simplemente desaparecerá y reaparecerá en un nuevo lugar de la casa para repetir la persecución; en caso de que el jugador sea alcanzado por el fantasma pero lleve cargando un mueble, iniciará un forcejeo con el fantasma donde el niño debe apretar la tecla espaciadora cinco veces para liberarse y poder salir de la casa.

Después de completar tres niveles normales se presenta un nivel bonus donde los objetivos cambian un poco. Durante el nivel bonus, varios objetos salen en fila hacia la casa. Todos son objetos más pequeños en comparación con los muebles y van desde comida hasta objetos de limpieza. También en el nivel bonus aparece una caja a la mitad del patio, y un texto le hace saber al jugador que el nuevo objetivo es recolectar los objetos que cumplan con una característica en común la cual aparece en la barra de información. Los objetos recolectados deben ser llevados a la caja y cuando hayan pasado todos los objetos, el jugador debe llevar la caja hacia el camión para completar el nivel bonus. La razón de la existencia de la caja es para que sea más fácil recolectar los objetos sin tener que ir de vuelta al camión por cada objeto seleccionado.

Un aspecto que tomó mucho tiempo en que terminara bien fueron las colisiones del personaje con las paredes de la casa, principalmente en la entrada principal. Durante las pruebas, algunos niños tenían problemas en la navegación y por lo mismo no podían ingresar a la casa porque no lograban alinearse bien con la puerta, por lo tanto los límites de las colisiones de las paredes son más indulgentes para que el paso sea más simple aceptando un poco de error con el roce de las mismas.

3.5 Arte

El arte dentro del desarrollo de juegos se refiere a los gráficos en dos o tres dimensiones como el Arte de Concepto, Modelos 3D de personajes, Sprites, storyboards, etc. En este trabajo se habla sobre estos elementos desde el punto de vista de computación gráfica y la relación que existe entre el arte y la programación.

Se realizó arte para las imágenes de los menús y la historia, más los de la interfaz gráfica. Las texturas de los objetos fue otro lugar en donde se requirió crear imágenes para “colorear” los muebles, personajes y elementos en general. Y por último el modelado 3D para generar todos los objetos.

3.5.1 Arte de la interfaz

El dibujo de las interfaces, en casi todos los casos, se tratan de imágenes en dos dimensiones que se encuentran en un plano que se puede considerar como el plano que se encuentra sobre la pantalla del monitor. La interfaz puede ser los botones que aparecen en el menú principal, el logo del juego e incluso las imágenes que sirven para mostrar información durante el juego como la barra de información. Normalmente existe al menos un artista dedicado a generar estas imágenes y debe existir un director de arte que tiene como tarea verificar que las imágenes de la interfaz cumplan con lo requerido y se encuentren en un formato que el equipo de desarrolladores pueda usar; en pocas palabras, se encarga de que todo se encuentre bajo el estándar acordado.

Las imágenes en computación gráfica se piden con dimensiones que sean múltiplos de 2^n , como 64, 128, 256, 512, etc. Dependiendo de la librería empleada, también se puede requerir que las imágenes sean cuadradas, es decir imágenes de 128x128 ó 1024x1024, etc. Esto es porque las imágenes que tienen estas dimensiones se guardan de una forma más óptima en memoria y por lo tanto no se gastan recursos y se pueden dibujar en pantalla de forma más eficiente.



Fig. 3.10. Imagen de la historia introductoria al juego.

Los gráficos de interfaz que requirió Fantasmagia fueron las necesarias para la pantalla del menú principal; botones y la imagen de fondo; imágenes para una historia introductoria que aparece al iniciar el juego (Fig. 3.10), los pequeños gráficos que aparecen dentro del juego para mostrar información sobre el estado en el que se encuentra el jugador y por último, la imagen al terminar el juego que muestra el fin y la puntuación final (Fig. 3.11).



Fig. 3.11. Imagen del fin del juego

3.5.2 Texturas

Las texturas requieren las mismas condiciones que las imágenes de las interfaces: deben ser de dimensiones múltiplos de 2^n y si se requiere, que sean cuadradas. Además de estos requerimientos, la organización de las texturas en imágenes es algo especial. Se puede ver a las texturas como un papel que envolverá a los modelos 3D para darles color y se pueden generar de dos maneras: a partir del modelo 3D ya generado, en donde el artista "pinta" sobre la figura ya terminada como lo haría un pintor de juguetes o figuras de cerámica, y posteriormente se pasa por un programa que "desdobra" la pintura para obtener una imagen cuadrada; una segunda forma es generar la imagen cuadrada y llenarla de las diferentes caras del objeto al cual va a "colorear", y el encargado de ajustar la imagen al objeto es trabajo del artista de modelado 3D (Fig. 3.12). El proceso de colocar y ajustar una imagen a un objeto tridimensional se le llama *mapping* o mapeado de texturas.



Fig. 3.12. Ejemplo de una textura para el modelo de la guitarra.

Dentro de Fantasmagia, todas las texturas se crearon en el programa de GIMP (*GNU Image Manipulation Program*), un programa de manipulación de imágenes de código abierto; las imágenes generadas cumplen con las reglas descritas en el párrafo anterior en diferentes tamaños dependiendo de a qué objeto iba a texturizar. Los objetos más grandes requieren texturas de mayores escalas para que se vean con mayor calidad mientras que los objetos pequeños pueden usar imágenes de menor escala sin perder calidad y para reducir el uso de memoria.

3.5.3 Modelos 3D

La creación de modelos 3D para software con gráficos en tiempo real se usan técnicas de optimización ya que el dibujo de estos objetos pueden exigir mucho poder de procesamiento para generar una única imagen. El modelado de objetos 3D se puede hacer usando programas como Maya 3D o 3D Max Studio. En este proyecto se usó una herramienta de distribución libre llamada Blender, la cual llega a ser tan buena y eficiente en varios aspectos como programas privados.

La principal técnica de optimización es la reducción de polígonos; los modelos 3D se componen de una colección de vértices, aristas y caras que se relacionan entre sí, para darle forma a los objetos que queremos representar, mientras más de estos elementos existan, más tiempo le tomará a una computadora dibujar el modelo en pantalla. Sin embargo, la reducción de las caras puede resultar en objetos de menor calidad, es por eso que existe un conjunto de técnicas para aumentar el realismo de los modelos mientras se mantiene la velocidad en el render de los mismos.

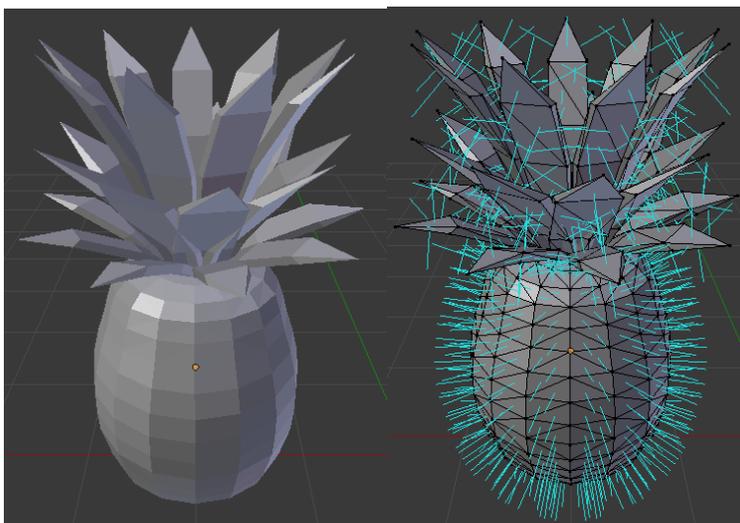


Fig. 3.13. Modelo en 3D (izquierda), Modelo 3D con Normales a sus caras (derecha)

Una técnica muy usada en este proyecto fue el uso de normales en los vértices. El cálculo de la iluminación de los modelos 3D emplea la ubicación de la fuente de luz en la escena 3D y los vectores normales a las caras, los cuales son vectores perpendiculares a la misma cara. Sin las normales en los vértices, la iluminación delata los límites de cada cara y les da un aspecto poco natural (Fig. 3.13). Al habilitar las normales de los vértices, la iluminación ayuda a ver al modelo con una superficie curva más suave; es más rápido valerse del cálculo de la luz sobre cada superficie a tener que dibujar múltiples caras para alcanzar el mismo efecto (Fig. 3.14).

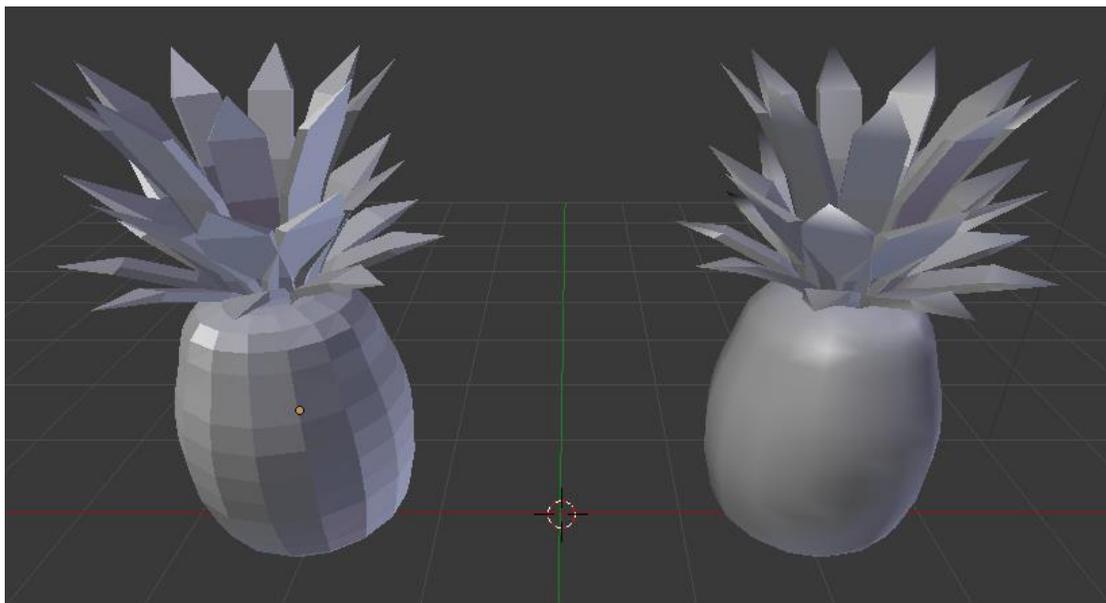


Fig. 3.14. Modelo sin normales a los vértices (izquierda), Modelo con normales a los vértices (derecha)

Toda la creación de modelos fue hecha especialmente para Fantasmagia porque muchos de los objetos 3D que se pueden encontrar en varias fuentes no son creados pensando que serán *rendereados* en tiempo real, como sucede en un videojuego. También, como se iba a requerir mostrar varios objetos al mismo tiempo, la optimización de modelos se volvió algo más imperativo ya que la convivencia de tantos modelos con una enorme cantidad de polígonos pudo afectar el rendimiento del juego.

Otra técnica de optimización empleada fue la división de áreas del juego para discriminar secciones del juego que no son necesarias que sean procesadas. El procesamiento implica usar recursos del sistema para dibujar las diferentes caras que forman objetos de la escena, calcular su iluminación, reflexiones, texturas, etc. Mientras más caras existan, mayor tiempo tardará la máquina en terminar de dibujar un solo cuadro y es probable que muchas de esas caras no aparezcan en pantalla, sin embargo requirieron tiempo de procesamiento. Dependiendo de cuántos cuadros por segundo se quiera alcanzar, se pueden emplear diferentes técnicas de selección de polígonos a dibujar, unas más complejas que otras y se distinguen por la cantidad de caras que pueden llegar a ignorar durante el proceso de *rendering*.

La técnica empleada para este proyecto no requirió de un algoritmo complejo ya que no existen varios objetos en escena y los escenarios son sencillos. La división en Fantasmagia se hizo aprovechando la idea de las diferentes áreas existentes en el juego: El patio y la casa; y la sección de la casa fue dividida en zonas aún más pequeñas. Esto quiere decir, que cuando el jugador se encuentra en el área del patio, todos los elementos de la casa, habitaciones, muebles y los fantasmas, no son tomados en cuenta durante el *render* o en la actualización de sus objetos.

Esta técnica, aunque simple, ayuda mucho durante el render de objetos de la escena en la que se encuentra el jugador y no desperdicia tanto tiempo en objetos irrelevantes. Mientras el jugador se encuentre en el patio, los objetos importantes son: el frente de la casa junto con la puerta principal, el pasto, las bardas a ambos lados del patio, el cielo, la calle, el camión de mudanzas y los objetos que vuelan del camión a la casa. Dentro de la casa, todo lo que pasa a dibujarse depende del cuarto en el que se encuentre el jugador; para el baño, la cocina y la alcoba, sólo se dibuja el mismo cuarto más el mueble que probablemente esté buscando el jugador en el momento. Para el caso de la sala, además de dibujar el mismo cuarto y los muebles, se agrega un fantasma que persigue al niño para sacarlo o forcejear lo que lleve cargando. El fantasma sólo se mueve cuando el jugador avanza o retrocede, desaparece cuando abandona la sala y aparece en una posición aleatoria cuando reingresa a la habitación.

3.6 Programación

La programación es la encargada de conectar todos los assets generados por las demás áreas; las imágenes, modelos, efectos especiales, textos, música, etc. Es la que brinda vida y responsividad por parte del software al usuario con el fin de generar una experiencia interactiva y lo más inmersiva posible. Las áreas que requieren de programación en un videojuego son variados, desde el comportamiento de botones en un menú hasta la inteligencia artificial. Otra clasificación puede ser en tres áreas: Programación de contenido, Programación de componentes y Programación lógica.

3.6.1 Programación de contenido

El contenido se refiere a los sonidos, texturas, efectos especiales, animación; todo lo que sea audiovisual. La programación de contenido es la integración de todos estos materiales al juego; es la encargada de establecer puntos de inicio de reproducción de una animación o sonido, o de mezclar sonidos dependiendo de las acciones del jugador, o de controlar la intensidad y canales de audio de la música, etc.

El contenido en nuestro proyecto se conforma de efectos de sonido, animaciones, música y pequeños efectos visuales. El control de este material audiovisual está en función de las acciones ingresadas por el jugador y de eventos del mismo juego. Los efectos de sonido son pisadas, el ladrido del perro, la risa del fantasma cuando te atrapa, sonidos que indican si el mueble tomado es correcto o incorrecto y un efecto adicional que indica que se ha dejado el elemento correcto dentro del camión.

Todas las animaciones en el juego se hicieron programáticamente, sin necesidad de un programa externo que auto-genera archivos de animación que el juego lea e interprete. Aunque el uso de animaciones programáticas lleva más tiempo de desarrollo, reduce el tamaño del producto final y mejora su ejecución. Las animaciones dentro de Fantasmagia son las del movimiento de caminata del jugador, el andar del perro, el giro y desaparición de los muebles falsos y del mismo fantasma cuando alcanza al jugador y por último las animaciones de textos informativos.

El comportamiento de la música está en función de en qué parte del juego se encuentre el jugador. Hay una canción que se reproduce durante la historia inicial y en el menú principal, y la canción cambia dependiendo del nivel: una para la mañana, la tarde, la noche y el nivel bonus.

Por último, el juego tiene un par de efectos visuales, todos hechos programáticamente. El primero es la transición de color que se puede ver en el cielo cuando el jugador avanza de nivel. El segundo se trata de partículas que se encuentran debajo del fantasma para simular una especie de aura que iba dejando mientras flotaba; el uso de shaders fue innecesario.

3.6.3 Programación de componentes

Los componentes son todos los objetos que existen dentro del mundo virtual, como los personajes, enemigos, accesorios, entre otros. Como ejemplo de este tipo de programación entra el movimiento del personaje de acuerdo a la entrada del jugador o las animaciones de personajes y objetos. Básicamente es la definición de las reglas individuales de cada objeto, cuánto se pueden mover, hacia dónde, cuáles son las acciones que realiza, etc.

En Fantasmagia, el personaje se mueve en dos sentidos, adelante y atrás, con las teclas de dirección arriba y abajo; mientras que las teclas derecha e izquierda lo hacen girar hacia esos lados correspondientes para poder cambiar su dirección. Los objetos que vienen del camión sólo se mueven del mismo hacia la casa, el perro y el fantasma siguen las mismas reglas de movimiento que el personaje, pero son independientes de la entrada en el teclado por el jugador.

Otros componentes son previos al juego, como por ejemplo la historia introductoria al inicio del programa. Hace uso de imágenes que se presentan con un texto en la parte inferior. El comportamiento de este componente es: mostrar una imagen en la parte superior y un texto en la parte inferior, esperar cinco segundos, cambiar el texto al siguiente en la cola, esperar cinco segundos, cambiar a la siguiente imagen y texto en la fila, repetir el comportamiento hasta que se termine con las imágenes y el texto en la fila. Por lo tanto, este componente necesita una lista de imágenes con otra de texto que sea el doble de largo que el de las imágenes para que

funcione correctamente. También tiene la funcionalidad de omitir; si el jugador presiona la tecla espaciadora, el componente pasará al final de las listas y terminará su funcionalidad.

El menú principal se considera otro componente, donde el jugador puede presionar las teclas de dirección arriba o abajo para cambiar la opción a seleccionar; lo cual puede hacer con la tecla espaciadora. Los créditos se muestran como texto en manera ascendente para mostrar nombres y el papel o actividad que desempeñaron en la creación de Fantasmagia. El comportamiento de la pantalla de los créditos es ir recorriendo el texto en la pantalla. Otro ejemplo de componente es la pantalla de selección de personaje, que es muy similar en cuanto a comportamiento a la del menú principal. El usuario usa las flecha de dirección derecha e izquierda para cambiar entre los dos posibles personajes con los que se puede jugar: el niño, Óscar y la niña, Eliza; y usa la tecla espaciadora para confirmar su selección.

3.6.4 Programación de lógica

Por último, la programación lógica es la parte que se encarga de conectar contenido y componentes para definir las leyes del mundo virtual. Si la programación de componentes es la definición de reglas independientes de cada objeto, la programación lógica son las reglas que rigen a todos esos componentes en conjunto. Estos límites y reglas de interacción que existen pueden ser que elementos son rígidos y no pueden ser atravesados por los personajes, qué ocurre cuando el jugador presiona la tecla espaciadora junto a un mueble, hacia dónde navega la aplicación cuando el texto de los créditos a terminado su recorrido, cuándo debe cambiar el nivel y el color del cielo, etc.

Aquí se programan las reglas para ganar, perder o avanzar en el juego; en nuestro caso, cuántos muebles se necesitan regresar al camión para subir de nivel, cuándo debe entrar al nivel bonus, cómo se comportan los objetos en el nivel bonus, qué debe cambiar, tanto gráfico como lógicamente en el juego si el jugador no logra interceptar el mueble antes de que entre de nuevo a la casa, qué pasa cuando el fantasma captura al jugador cuando éste lleva un mueble en las manos, qué ocurre cuando no, etc.

En general, no existe un orden en el que se deba completar cada parte de la programación. Un programador de lógica puede empezar a trabajar con el comportamiento del enemigo aunque no exista el contenido del mismo; puede usar un cubo o una esfera como un modelo provisional en lo se le entrega el producto final. Sin embargo, es importante que exista una buena comunicación entre los mismos programadores para que todo encaje correctamente y que la convivencia entre los elementos se vea natural.

3.7 Requisitos funcionales y no funcionales

A manera de resumen del comportamiento del juego, se nombran los requisitos funcionales (RF) y no funcionales (RNF) con los que se construyó Fantasmagia. Los RF son aquellos que describen qué debe hacer el sistema ante las diferentes entradas, ya sean del usuario o de algún otro sistema; ejemplo de ellas son la respuesta del sistema cuando se presiona cierta combinación de teclas. Por su parte, los RNF hablan sobre las propiedades del sistema, como la seguridad, rendimiento o disponibilidad.

Como todo videojuego, la respuesta del sistema es en tiempo real, es decir cuando el jugador presiona un botón el software da una retroalimentación inmediata de su acción en el mundo virtual. Tal es el caso durante el movimiento del personaje con las flechas direccionales o cuando se presiona la barra espaciadora. Durante el último evento, si hay un objeto junto al personaje del juego; éste debe cargarlo y también le tiene que informar si ha seleccionado un objeto que cumple con las características que el sistema le muestra en pantalla en una letra de molde. Al presionar la barra espaciadora nuevamente, el personaje debe soltar el objeto y éste debe desaparecer si se trataba de uno incorrecto y así evitar que lo tome nuevamente accidentalmente o seguir su camino a la casa si era el correcto. Dentro de la casa, como RF se tuvo que el fantasma se moviera sólo cuando se moviera el personaje, para facilitar el trayecto al niño y darle tiempo para pensar y planear su estrategia. En caso de que el fantasma atrapara al jugador ocurre una de tres cosas elegidas al azar. El primer escenario es que el fantasma sólo gire y desaparezca mientras se ríe en tono de juego y vuelve a perseguirlo desde un nuevo punto de la casa. Como segundo y tercer posible desenlace, si el personaje no está cargando un mueble, el fantasma lo llevaría hacia afuera de la casa, pero si tiene un objeto en sus manos, el fantasma iniciaría un forcejeo con el fantasma por el objeto. En los últimos dos escenarios, es posible evitarlos si el jugador presiona la barra espaciadora 5 veces.

El juego lleva un conteo de los puntos por cada mueble correcto que es llevado al camión, y por los objetos del nivel bonus depositados en la caja. Después de 3 niveles con 2 niveles bonus, el juego termina. No tarda más de 15 minutos, el terapeuta que lleve la sesión puede hacer avanzar el juego presionando la tecla R si le toma mucho tiempo al niño cada nivel.

Los RNF giran en torno a las especificaciones de hardware con las que contaban el laboratorio del instituto. Este videojuego tuvo la necesidad de tener un rápido rendimiento y de ser compacto para que pudiera ser usado por las terapistas en sus sesiones con los niños. Los equipos de cómputo contaban con muy poca memoria RAM (512 MB), procesadores anticuados y ningún tipo de tarjetas para ayudar con el procesamiento de video, por lo que Fantasmagia se desarrolló pensando en que debía ser simple, rápido y de óptimo manejo de los pocos recursos disponibles.

“Los jugadores son artistas que crean su propia realidad dentro del juego.”

-Shigeru Miyamoto

Capítulo 4 - Modificaciones y pruebas

Contamos con el apoyo para realizar pruebas de las diferentes etapas del desarrollo de Fantasmagia con niños que eran pacientes de las terapistas. Estos ejercicios nos ayudaron a encontrar fallas y realizar mejoras de acuerdo a las reacciones y comentarios de los niños y terapistas. También realizamos muestras del software a personal y autoridades del INR.

Previo a las pruebas con los niños, se les mostró a las terapistas los avances del software conforme se iban realizando. Cuando se tuvo una versión del juego con un ciclo casi completo de juego y niveles con objetivos definidos, se le consideró listo para las pruebas. A partir de entonces, todas las modificaciones, mejoras y correcciones se hicieron con un solo objetivo en mente: que el juego sirviera como herramienta en la terapia mientras mantenía la atención completa del niño.

Para cumplir con el objetivo dado, se necesitaba la aprobación de dos áreas. La primera era el área de las terapistas; durante la muestra de avances, ellos buscaban qué partes del juego servían al propósito de “apoyo a la terapia” y qué partes necesitaban ser cambiadas para cumplirlo. La segunda parte le corresponde al diseño del software, para mantener el interés del niño. Como diseñadores del juego, la forma de conseguirlo es enfocándose en crear la experiencia que queremos que el jugador sienta en cada sección del videojuego.

4.1 Pruebas con la versión alpha

Las primeras pruebas del juego se realizaron cuando se tenía una versión casi estable, con niveles funcionales pero sin un final concreto. La versión alpha del juego tenía un concepto diferente del juego terminado: El objetivo era evitar que muebles flotadores “escaparan” de la casa, el tiempo que estos objetos tardaban en salir de la casa hasta la calle a través del patio era el tiempo que el jugador tenía para identificar el mueble que cumpliera con las características enlistadas en pantalla. Una vez encontrado el objeto, éste debía ser colocado en el lugar de la casa al que correspondía.

La perspectiva del juego era desde un costado de la casa, siendo un pasillo largo que conectaba todas las habitaciones una al lado de otra. El movimiento del personaje podía ser en dos dimensiones: izquierda o derecha; arriba o abajo.

4.1.1 Modificaciones y errores encontrados

Durante las primeras pruebas se encontraron errores esperados (falsas características para algunos objetos, elementos gráficos no mostrados, colisiones de objetos mal acomodados), cambios necesarios en la interfaz (colores más contrastantes, objetos difíciles de identificar, texturas mal mapeadas), pero lo más relevante fue que identificamos puntos importantes en la interacción que dificultaban una sesión de juego fluida y quebrantaban la experiencia que se buscaba crear. Los controles del personaje resultaron problemáticos, algunos niños tenían dificultad al momento de navegarse por la casa, sobre todo por las colisiones de las paredes y portales de las puertas. Era común ver niños que se quedaban atorados al querer doblar una esquina o que no podían siquiera entrar a la casa. También había complicaciones al tratar de acercarse a los objetos y tomarlos ya que varias veces no se acercaban lo suficiente o pasaban de largo el mueble y les costaba regresar por él.

Algo que aprendimos rápidamente en las primeras pruebas fue la cantidad enorme de información que se puede recolectar en cada una de ellas por parte de los niños. Además de los errores y modificaciones antes mencionadas, empezaron a cuestionar el concepto del juego pidiendo explicaciones para lo que no tenía una razón conocida para ellos, lo cual nos hizo elaborar más en detalles que nunca pensamos. Así el concepto pasó a tratarse de un ladrón que intentaba robar los muebles de la casa, creando muebles falsos (que correspondían a los muebles que no cumplían con las características de la lista dada) e intentaba jalar todos los objetos, falsos y el verdadero, con un ventilador gigante.

4.2 Pruebas con la versión beta

Con el segundo concepto del juego aplicado, nuevamente tuvo que pasar por la revisión de las terapistas. Lo que tenían que aprobar era básicamente la historia, gráficos del nuevo concepto y objetos y muebles adicionales; por su parte, el diseño del juego siguió evolucionando, se preparó el sistema para agregar pequeñas misiones, como por ejemplo preparar comida, arreglar el cuarto o bañar a un perro, cuyo modelo 3D y comportamiento también se agregó.

Se ideó una historia que tuviera sentido con las mecánicas del juego. Como los muebles se movían solos y se alejaban de la casa, lo que inventamos fue que los muebles estaban siendo robados por un ladrón que con una aspiradora gigante succionaba las cosas de la casa. Lo que nos faltaba encontrar un "porque" se necesitaba identificar el mueble que se distinguía y cumplía con las opciones de la lista dada. La explicación dada es que el ladrón había creado muebles falsos para confundir a los niños y poder robar los objetos con mayor facilidad. La historia fue añadida en respuesta a las preguntas de los niños sobre la motivación, razón o motivo del por qué teníamos que recobrar los muebles correctos. Esta adición vino con nuevos objetos, en el mundo, como la aspiradora y los gráficos para contar la historia de manera amigable y simple.

Empezamos a trabajar en las bases de sistema para las misiones especiales dentro del juego, esto implicó agregar nuevos objetos, como una licuadora, frutas, leche, una tina, un jabón, etc. -objetos que tenían relación entre ellos o que se usan en conjunto para realizar una acción o crear un producto final. Por ejemplo, una misión podría involucrar juntar los ingredientes y materiales necesarios para hacer un licuado, reunirlos en la cocina y preparar el licuado. Rápidamente nos dimos cuenta que muchas de las misiones ideadas constaban de reunir

objetos específicos en un lugar y que la variedad de objetivos resultarían en decenas de objetos que se tendrían que modelar.

Se terminó de crear un sistema de niveles que se representa como las diferentes fases del día: mañana, tarde y noche. Cada uno de estos niveles se comprendía de 3 grupos de muebles, llamados oleadas, donde se muestra una lista de características semánticas y el jugador debe elegir el objeto del grupo que cumpla con las características. Sólo aparece una oleada a la vez y la lista y los objetos son diferentes por cada una. Entre niveles aparecían los objetivos especiales, que para la versión beta sólo eran unos pocos; no más de cinco. Por último, esta versión ya comenzó a tener música y sonidos que fue algo que pidieron los niños durante las pruebas de la versión alpha.

4.2.1 Modificaciones y errores encontrados

Los niños respondieron muy bien a los cambios realizados, inmediatamente fue claro que el incremento en contenido renovó su interés y su emoción. Nuevamente los bugs y errores aparecieron, como por ejemplo el perro que quedaba atorado en una esquina. El proyecto parecía estar casi completo, sólo restaba agregar nuevas misiones y agregar un final visible para indicar el fin del juego; por desgracia era necesario invertir mucho tiempo para crear nuevos objetivos y la idea era incluir bastantes de ellos con el fin de mantener un interés en el juego por mucho tiempo por parte de los niños. Esto nos obligó a reevaluar el proyecto e idear cambios mayores sin la necesidad de invertir tanto tiempo.

Así es como finalmente se llegó al concepto que formaría Fantasmagia y que ya se ha explicado en anteriores capítulos. La razón por la que se apostó por estos últimos cambios fue por la rapidez con la que se podían realizar y el nivel de interés que tendrían los niños.

Como se prometió, los cambios para crear Fantasmagia se realizaron de manera rápida. Se presentó la nueva idea a las terapistas para obtener su opinión previa a que se aplicaran los cambios. La idea de tener fantasmas dentro del juego tuvo una recepción de cautela, sin embargo como se pudieron realizar los cambios de manera rápida, pudimos realizar un demo para que ellas pudieran decidir si se aceptaba el nuevo concepto o si se mantenía mejor como estaba.

El diseño gráfico se hizo con la idea de que tenía que ser apto para niños. Por lo tanto los fantasmas tienen características más infantiles (cabeza, manos y rasgos faciales grandes, cara sonriente). No se realizaron cambios en la luz o en el diseño de la casa, aunque se nombraba como una casa embrujada para no crear un ambiente de terror o suspenso; se mantuvieron los colores llamativos.

El comportamiento de los fantasmas también tuvo que ser adecuado para el concepto del juego. Para facilitar el juego, en especial para niños que no están acostumbrados a jugar videojuegos 3D, el fantasma no se acercaba a los niños si ellos no se movían de su lugar, lo cual les permitía evaluar su situación, planear cuál sería su ruta de escape y hacía parecer más la interacción con los fantasmas como un juego, sin tantas presiones. En dado caso de que el fantasma atrapara al jugador ocurriría una de dos cosas: el fantasma se ríe y desaparece o él empieza a forcejear con el jugador por el mueble. En el primer caso cuando el fantasma sólo desaparece, se agregó para reforzar la idea de un ambiente amigable con fantasmas que sólo quieren jugar. En dado caso que se inicie un forcejeo con el jugador por el mueble, el niño tiene la posibilidad de mantener la posesión del objeto si presiona la barra espaciadora cinco veces, lo cual hace que el fantasma ría y desaparezca.

El nuevo demo se presentó a las terapistas las cuales aprobaron los nuevos cambios y se dio luz verde para entregar una versión final para prueba con niños.

4.3 Pruebas con la versión final

Cuando se presentaron los cambios a los niños, ellos respondieron con mayor emoción. La nueva idea de fantasmas amigables persiguiéndolos por la casa les acumulaba exaltación que se liberaba como risas y gritos al ser capturados o como alivio al escapar con el mueble de la casa. Se hicieron múltiples pruebas con grupos diferentes de niños para ver cómo responderían. Se llegó a dar el caso donde niños que ya tenían amplia experiencia con el juego, que habían jugado desde sus primeras versiones, empezaban a enseñarles ellos mismos a los que apenas iban conociéndolo al grado en el que se olvidaban de la presencia de los terapistas y la nuestra. Fue un agrado ver su reacción, lo que nos hizo ver que ya teníamos un producto final del proyecto.

No sólo se realizaron pruebas con niños, también se mostró Fantasmagia a personal del Instituto Nacional de Rehabilitación. Comenzamos a hacer reuniones especiales para presentar el juego y que otros pudieran ver el funcionamiento y propósito. Todos comentaron sobre lo único y especial que era la idea, y pronto se empezó a hablar sobre su uso como herramienta de terapia.

4.3.1 Modificaciones y errores encontrados

No hubo muchas modificaciones en cuanto a errores y apariencia del juego, más allá de pequeños errores en las características de ciertos objetos e imágenes y texturas. En este punto la mayoría de los cambios se realizaron en la presentación de los créditos y ajustes en la fuente del texto; últimos detalles que ya no afectaban a la jugabilidad.

Los únicos cambios mayores fueron un sistema de puntaje y una imagen final para indicar el término del juego mostrando los puntos obtenidos para fomentar la competencia. Se hicieron pequeños ajustes a la velocidad de los objetos y la duración del juego ya que la idea era trabajar con Fantasmagia en grupo de niños y todos deberían tener la posibilidad de jugar.

Entre pruebas se hizo un nuevo nivel bonus que presentaba una hilera de objetos que iban desde el camión hacia la casa. La idea de este nivel bonus era obtener todos los objetos que compartieran una característica en común, no había penalización en la pérdida de objetos, sólo era para obtener puntos extras. A las terapistas les agradó bastante esta pequeña adición ya que trabajaba de una manera un poco diferente la discriminación semántica de los objetos.

Una vez completados los cambios se preparó para presentar el proyecto final en el 3er Curso **“Manejo Interdisciplinario de los Trastornos de Comunicación Humana”** que se realizó en el mismo Instituto Nacional de Rehabilitación. Durante el curso pudimos mostrar el proyecto final y hablar sobre las sesiones que se tuvieron con niños.

4.4 Entrega final

En muchas ocasiones la pregunta obligada durante las reuniones y pruebas era cuándo tendríamos una versión para la entrega final. Tardamos un poco en tener todo listo para considerar el proyecto como concluido por los pequeños detalles que tuvimos que atender, en su mayoría en la parte de los créditos. Una vez que todos aprobaron los últimos cambios y no hubo más agregados, pudimos entregar un disco con el ejecutable del juego listo para ser usado.

Por último, se realizó otra pequeña prueba y muestra con otras terapistas para enseñarles cómo funcionaba, aprendieran a manejarlo y pudieran empezar a usarlo con sus grupos de niños y generar datos sobre el verdadero apoyo que brindaría Fantasmagia como herramienta en la rehabilitación del lenguaje. Una manera simple para medir la eficacia del programa, ideada por el Lic. Manuel Contreras Ordaz, es creando un cuestionario que sería llenado por las mismas terapistas de acuerdo con la experiencia que tengan trabajando con su grupo de niños. Las preguntas serían relacionadas con lo funcional que fue el programa durante las sesiones de terapia y su opinión personal.

“La mente que se abre a nuevas ideas, jamás volverá a su tamaño original”

-Albert Einstein

Capítulo 5 - Conclusiones

El objetivo de este proyecto era generar una herramienta que apoyara en la complejidad que es la rehabilitación del lenguaje en niños. Muchas dificultades que se tienen que enfrentar los terapeutas son la inquietud de los niños, problemas con familiares, poco material de trabajo y escasez en la renovación de técnicas de terapia.

Logré crear un juego completo basado en ejercicios empleados en la terapia actual y probarlo con niños en proceso de rehabilitación. Me apoyé en el uso de nuevas tecnologías más la idea del Gamification para transformar el proceso por el que pasan pacientes con problemas de lenguaje en uno más placentero y divertido.

En el capítulo uno se planteó los diferentes problemas con los que trabajan los terapeutas, su origen, clasificaciones, desarrollo humano, varios problemas y formas de trabajo en cada una de ellos. Esto me sirvió para crear un proyecto que realmente trabajara sobre el conocimiento médico y lograra su objetivo. Como la comunicación es un proceso bastante complejo y amplio, nos enfocamos en un sólo área de problemas del lenguaje ya que sería muy difícil y caro crear un juego que fuera una herramienta en la rehabilitación de todas las áreas de la comunicación humana.

Una vez identificada el área de trabajo se eligió un tipo de software que ayudó a realizar el objetivo planteado: ayudar en la terapia de la rehabilitación del lenguaje de niños entre los 8 y 10 años de edad. Es por eso que en el capítulo dos se habla sobre todos los tipos de software y la forma de desarrollarlos. Aunque rápidamente la decisión de realizar un software de entretenimiento por ser sus usuarios finales niños en proceso de rehabilitación, se habló a detalle los tipos de software de entretenimiento, los videojuegos, su creación, clasificación y uso fuera del campo de entretenimiento.

En el capítulo tres se empezó a hablar sobre el proceso de desarrollo que tomó Fantasmagia, desde concepto de computación gráfica, técnicas que usa el juego para correr en máquinas sin tanto poder de procesamiento, el desarrollo de imágenes, pruebas, mantenimiento y presentaciones. Se divide la descripción del desarrollo en diferentes áreas, ya que los videojuegos hacen uso de muchas tecnologías para su creación.

Por último, el capítulo cuatro contiene las pruebas, transformaciones y detallado de las versiones por las que pasó Fantasmagia. Lo que considero más importante es como se tuvo que ir modificando por los resultados de las pruebas con los niños hasta llegar al producto final, ya sea por influencias externas u observaciones del uso del programa. Se resaltan parte de las

dificultades que tuvo el programa y las soluciones encontradas hasta entregar el juego en su versión final. Aunque hubo varias pruebas con niños, éstas sirvieron únicamente para la creación y no como estudio formal de los posibles beneficios que pueden existir por el uso del juego como herramienta en la terapia de problemas del lenguaje en niños. Hubo pláticas con autoridades del Instituto Nacional de Rehabilitación para realizar estudios con diversos grupos de niños usando Fantasmagia como parte de su terapia, sin embargo por diversas dificultades ya no se pudieron completar.

5.1 Conocimiento adquirido

Durante el largo proceso de creación aprendí bastante sobre la planeación de proyectos, la importancia de establecer un plan, analizar el problema, tomarse el tiempo de ingeniar un diseño antes de empezar el desarrollo. Es mejor encontrar y corregir problemas durante las primeras etapas del desarrollo de software antes de que se tengan incontables líneas de código y cueste más trabajo y tiempo componerlo. El tener un plan trazado también sirve para medir y controlar el avance del proyecto; se puede comparar el progreso real contra el progreso estimado y tomar medidas conforme se necesite.

La presencia de las pruebas con cada avance del proyecto, me hizo ver la importancia de las mismas al enterarme de varios problemas y cambios necesarios del gameplay o la interfaz gráfica. En cada prueba, resaltaron errores que difícilmente pudimos haber encontrado nosotros, ya que nuestros *testers* expresaron las posibilidades del software.

Dentro de la parte de programación, aprendí sobre una buena estructuración del proyecto siguiendo el paradigma de programación orientada a objetos. Se facilita el trabajo; es más fácil y fluido el mantenimiento y corrección de errores. Otra valiosa lección, es sobre la complejidad de un software de entretenimiento gráfico, o videojuego. Se vuelve importante la optimización de recursos ya que los gráficos por computadora son muy demandantes, es bastante lo que tiene que computar la máquina como para que tomemos a la ligera la complejidad del código que escribimos.

Por último, pero igual de importante, fue la interacción con las personas que solicitan el desarrollo de software, los usuarios finales y terceros que mostraron interés por el proyecto. Tuve la oportunidad de tratar directamente con la gente que evaluó el programa ya que son las que esperaban recibir parte del beneficio que pueden brindar el apoyo de nuevas tecnologías. Me reuní con ellos, hablamos sobre el camino que tomaría el desarrollo y evaluamos el progreso. Tratamos con los usuarios finales, los niños en terapia, para realizar las pruebas y obtener retroalimentación con sus comentarios y preguntas. El tercer grupo de personas involucradas fueron las presentes en las conferencias y personal del INR a los que se les presentó nuestra propuesta y también nos llenaron con sus opiniones y comentarios. Todo resultó ser una experiencia positiva que se redirigió hacia el mejoramiento de Fantasmagia y la solidificación del concepto del desarrollo de software.

Nos enfrentamos a reuniones con los *clientes*, donde experimentamos la toma de requerimientos, entender la necesidad que se tenía y explicaciones mutuas: tanto ellos intentando que entendiéramos exactamente lo que necesitaban y hacían actualmente, y nosotros enseñándoles nuestros trabajos y nuestras propuestas para cumplir sus requerimientos. Realicé

prototipos, reuniones para mostrar avances, presentaciones, pruebas; realmente siento que trabajamos en un proyecto verdadero que cubre necesidades reales usando por primera vez muchos de los conceptos, procesos, planes, técnicas y experiencias que aprendimos en la Facultad de Ingeniería de la UNAM.

5.2 Resultados obtenidos

Este trabajo tiene tres objetivos los cuales fueron presentados en la Introducción. Aquí se presenta un análisis de los resultados del producto final.

Para comenzar, se entregó un videojuego completo con el cual se puede trabajar en una sesión de terapia, que era el primer objetivo. El empleo de un videojuego como herramienta en la rehabilitación del lenguaje fue recibido con enorme entusiasmo por parte del Licenciado Manuel Contreras Ordaz y su equipo de terapeutas. En el año 2018, el juego aún es usado en terapias con niños y por parte del mismo Lic. Manuel en su vida personal.

Sin duda, estoy convencido que el segundo objetivo, que era capturar la atención de los niños en sus ejercicios de rehabilitación, se ha cumplido desde las primeras versiones. Después de aprender cómo jugar Fantasmagia, los niños se adentran al mundo y las terapeutas no tuvieron problemas para usar esta atención para ejercitar el lenguaje mediante comentarios y reflexiones que el mismo videojuego facilita (Fig. 5.1).



Fig. 5.1. Paciente probando el juego.

La experiencia general de los infantes fue de curiosidad, asombro y felicidad. Se comunicaban y se apoyaban entre ellos, realizaron el cambio de sus ejercicios al medio digital de una manera fluida (Fig. 5.2).

El tercer objetivo es un poco más complejo de medir. En un escenario ideal, se busca lograr que el empleo de un videojuego serio en la rehabilitación del lenguaje ayude a agilizar la

terapia e incrementar su eficiencia. Para tener una correcta conclusión sobre este asunto se requiere de la participación del área de Rehabilitación para que realice pruebas con varios grupos de niños y examinar su progreso por un tiempo determinado. Por desgracia no pudimos contar con los recursos para un estudio formal.

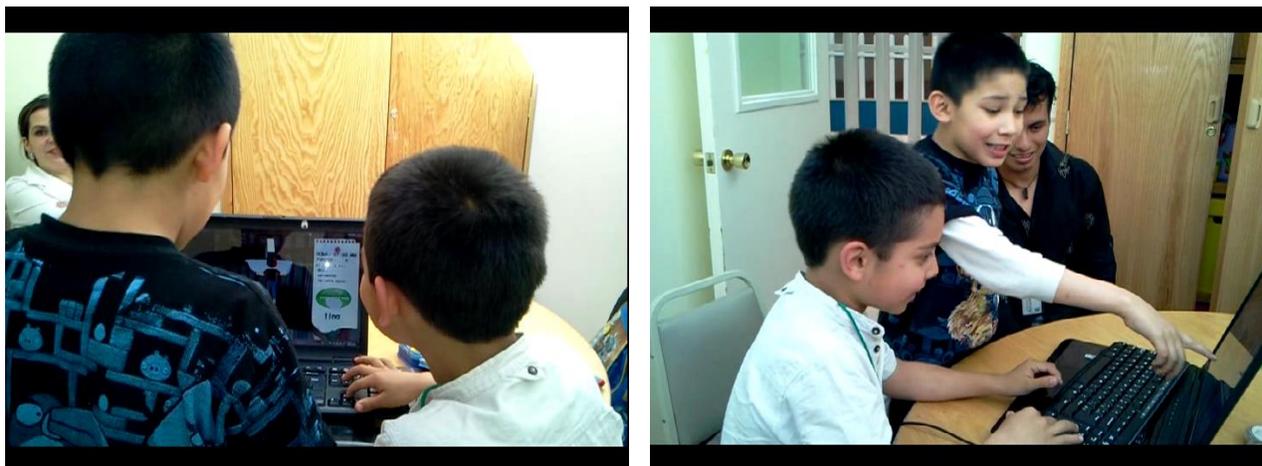


Fig. 5.2. Dos niños jugando mientras las terapistas los supervisan.

Aun así, tomé los comentarios y observaciones de los y las terapistas del lenguaje que usan el juego y creo que en base a su amplia experiencia en el campo son capaces de juzgar la efectividad de una herramienta de apoyo. Fantasmagia recibió atención y elogios por parte de los médicos del INR, las terapistas del lenguaje decidieron agregarlo a su material de trabajo y otras áreas de rehabilitación (de lenguaje y números) comentaron lo interesados que estaban en obtener un juego para su campo.

5.3 Trabajo a futuro

Como lo indiqué anteriormente, el proyecto de Fantasmagia fue un completo desarrollo en el cual puse en práctica mucho de lo que se nos enseñó en la Facultad. En definitiva, a mí me hizo cimentar muchos conceptos y comprender mejor los consejos que muchas veces nos brindaban los académicos.

Para empezar, la forma de trabajo y organización del desarrollo de software. La organización del código del programa, en específico la estructura de los programas que usan el paradigma de programación orientada a objetos. Entendí mejor el por qué y cómo se deben organizar todos los archivos de código fuente y recursos (como las imágenes o sonidos), ya que de no tener un orden, es fácil perderse entre tantos archivos y dificultar las correcciones y mantenimiento. Al ser un videojuego, el programa depende de que pueda ser dibujado en tiempo real sin retrasos, por lo que pude poner a prueba distintas técnicas de optimización de modelos 3D.

Con esta experiencia pude darme una gran probada sobre el desarrollo de proyectos de computación gráfica, los problemas que tienen, como la consideración del poder computacional

del equipo en el que va a correr el producto final, las técnicas para mejorar el rendimiento en dicha máquina. Pero lo que consideraría lo mejor que obtuve fue la administración del proyecto; ya que pude tomar las riendas del avance, la forma y ejecución de cada aspecto del software y no sólo el desarrollo. Ser el responsable de las decisiones que afectaron el producto final me hizo comprender mejor el por qué están definidos los diferentes pasos en las metodologías de desarrollo de software. Ahora soy capaz de juzgar dichas metodologías con una mejor idea de qué esperar de cada una y cómo adaptarlas para el producto final.

Durante el desarrollo del videojuego, un consejo que siempre veía de varios diseñadores de videojuegos era el pensar siempre en la experiencia final de usuario. Cada luz, cada color, cada sonido, cada texto, cada decisión; debe tener como base para su existencia las sensaciones y emociones que se desean por parte del jugador. Si se decidió hacer un juego de terror, entonces cada elemento debe buscar inducir el miedo como experiencia principal al jugador, o si se hace un juego de *football*, queremos que se sienta como si estuviera viendo un partido en el estadio. Esta ideología la he adoptado no sólo para otros videojuegos que he desarrollado, también para cualquier software en los que tome parte. El enfoque que le doy a mis trabajos ahora es enfocado al usuario final del mismo donde espero que tenga la mejor experiencia con comodidad, facilidad y sin problemas u obstáculos innecesarios.

Por último, creo que hay un gran futuro en los "videojuego serios" y me encantaría poder indagar más en ese campo para usar las nuevas tecnologías para realizar cambios importantes en el mundo mientras se tiene mucha diversión.

Bibliografía

Abt, C. C. (1970). *Serious Games*. Boston: University Press of America.

Acosta, V. y Moreno, A, (2005). *Dificultades del lenguaje en ambientes educativos*. Barcelona: Masson.

Barrett, M. D. (1986). *The development of word meaning*. New York: Springer-Verlag.

Barrett, M. D. (1995). *The handbook of child language*. Oxford: Blackwell Publishing.

Bronckart, J. (1980). *Teorías del lenguaje*. Barcelona: Herder.

Crawford, C. (1997). *The art of computer game design* [on line]. Washington State University Vancouver. Disponible en:
http://www.digitpress.com/library/books/book_art_of_computer_game_design.pdf

Dioses, A. (2004). *Clasificación y Semiología de los Trastornos del Lenguaje en el Niño*. Manuscrito no publicado, Universidad Nacional Mayor de San Marcos en Lima, Perú.

Froemling, K., Grice, G. y Skinner, J. (2011) *Communication: The Handbook*. Boston, MA: Allyn & Bacon.

Granic, I. Lobel, A. y Engels, R. (2014). The benefits of playing video games. *American Psychological Association*. Radboud University Nijmegen. Vol. 69, No 1, p. 66-78.

INEGI (2016, 10 de agosto). *Estadísticas a Propósito del... Día Internacional de la Juventud (15 a 29 años)*. *Datos Nacionales*. Recuperado de
http://www.inegi.org.mx/saladeprensa/aproposito/2016/juventud2016_0.pdf

Lemley, L. (2004). Discovering Computers [on line]. Disponible en:
<http://uwf.edu/clemley/cgs1570w/notes/Concepts-3.htm>

Margaret, R. (2006, abril). Definition: software [on line]. Disponible en:
<http://searchmicroservices.techtarget.com/definition/software>

Martínez, J. (2008, 10 de diciembre). Neurolingüística: patologías y trastornos del lenguaje. *Revista Digital Universitaria*. Recuperado de
<http://www.revista.unam.mx/vol.9/num12/art103/art103.pdf>

McGonigal, J. (2010, febrero). Gaming can make a better world. TED 2010.

McGonigal, J. (2011, 22 de enero). Be a gamer, save the world. *The Wall Street Journal*.

Narbona, J. y Chevríe, C. (2003). El lenguaje del niño. Desarrollo normal, evaluación y trastornos. Barcelona: Masson.

Peña, J. (1988). Manual de Logopedia. Barcelona: Masson.

Piaget, J. (1972). El juicio y el razonamiento en el niño. Buenos Aires: Guadalupe. (Trabajo original publicado en 1924)

Pressman, R. S. (2010). Ingeniería del software. México: McGraw-Hill.

Puyuelo, M. y Rondal, J. A. (2005). Manual de desarrollo y alteraciones del lenguaje. Barcelona: Masson.

Rodríguez, R. (2009). Advierte de problemas de lenguaje infantil. *El Universal*. Recuperado de <http://archivo.eluniversal.com.mx/nacion/165805.html>.

Saussure, F. (2011). Course in General Linguistics. Nueva York: Columbia University Press. (Trabajo original publicado en 1916)

Schrage, M. (2000). Serious Play: How the world's best companies simulate to innovate. Boston: Harvard Business School Press.

Schwaber, K. (2004) Agile Project Management with Scrum. Microsoft Press.

Shethna, J. (2015, diciembre 21). What is Application Software & Its Types [on line]. Disponible en: <https://www.educba.com/what-is-application-software-its-types/>

Skinner, B. (1981). Conducta Verbal. México: Trillas.

Sommerville, I. (2011). Software Engineering. Boston: Pearson.

Ward, J. (2008, abril 29). What is a Game Engine? [on line]. Disponible en: https://www.gamecareerguide.com/features/529/what_is_a_game_.php

Whitten, J. L. y Bentley, L. D. (2007). System Analysis & Design Methods. New York: McGraw-Hill/Irwin.

Whorf, B. (1971). Lenguaje, pensamiento y realidad. Barcelona: Barral.