

РАЗРАБОТКА ЯЗЫКА ДОПОЛНЕННОЙ ВИРТУАЛЬНОСТИ ДЛЯ ИГРОВЫХ ПРИЛОЖЕНИЙ НА ОСНОВЕ ТЕОРИИ ФОРМАЛЬНЫХ ЯЗЫКОВ И ГРАММАТИК

DEVELOPMENT OF LANGUAGE OF AUGMENTED VIRTUALITY FOR GAME APPLICATIONS BASED ON FORMAL LANGUAGE AND GRAMMAR THEORY

A. Mikitchuk

Summary. This article is devoted to the study and development of the language of augmented virtuality on the basis of methods and approaches of the theory of formal languages and grammars. The author draws attention to the definition of augmented virtuality in the model of the information system for gaming applications with augmented virtuality and puts the task of developing the lexical and semantic analyzers for formal language. Considerable attention is paid to the definition of the alphabet and the grammar of the language, the finite automaton of the recognizer, regular expression and the syntax diagram. Describes software implementation of lexical and semantic procedures generation virtual game levels using the built-in Game Maker Studio programming language. The conclusion is made about the applicability of the theory of formal languages and grammars to the development of mechanisms for the interaction of components of information systems of augmented virtuality.

Keywords: mixed reality, augmented virtuality, lexical analyzer, syntax diagram, semantic procedure, Game Maker.

Микитчук Андрей Алексеевич

Аспирант, Саратовский социально-экономический институт (филиал) ФГБОУ ВО «Российский экономический университет имени Г.В. Плеханова»
andrew1709@mail.ru

Аннотация. Настоящая статья посвящена изучению применения методов и подходов теории формальных языков и грамматик для разработки языка дополненной виртуальности. Автор останавливается на определении термина дополненная виртуальность в рамках модели информационной системы для игровых приложений с дополненной виртуальностью и ставит задачу разработки лексического и семантического анализаторов для формального языка. Значительное внимание уделяется определению алфавита и грамматики языка, приводятся конечный автомат распознаватель, а так же регулярное выражение и синтаксическая диаграмма. Описывается программная реализация анализаторов и семантической процедуры генерации виртуальных игровых уровней с помощью встроенного в Game Maker Studio языка программирования. В заключении делается вывод о применимости теории формальных языков и грамматик для разработки механизмов взаимодействия компонент информационных систем дополненной виртуальности.

Ключевые слова: смешанная реальность, дополненная виртуальность, лексический анализатор, синтаксическая диаграмма, семантическая процедура, Game Maker.

В современном мире наибольшую популярность в массовом сегменте развлекательных программных продуктов имеют компьютерные игры и мобильные игровые приложения. В то же время задачи, возникающие при создании и наблюдении за поведением математических, физических, экономических, имитационных и других моделей, могут быть решены при помощи методов и подходов, используемых при разработке игровых приложений, а также путем взаимодействия с исследуемыми моделями в рамках них. Поэтому реализация игр является актуальной задачей как сама по себе, так и с прикладной точки зрения.

С развитием информационных технологий появились инструменты, которые не требуют профессиональных навыков для создания компьютерных игр. Поэтому на популярных площадках распространения программ публикуется все большее число игровых приложений со схожим геймплеем. Проблема уникальности является основной для разработчиков компьютерных игр.

Принимая во внимание факт разнообразия игровых жанров и наибольшее количество однотипных приложений в казуальных играх, мы будем рассматривать 2d платформеры. Это один из старейших жанров, основные принципы которого унаследовали многие жанры. Игровой процесс большинства двухмерных платформеров заключается в управлении неким персонажем и многократном выполнении несложных действий на однотипных игровых уровнях для достижения заданного результата.

Одним из решений проблемы уникальности в рамках игрового жанра платформер служит предоставление разработчикам готового инструмента на базе технологий смешивания реального и виртуального миров. Среди таких технологий, объединенных термином смешанная реальность и использующих алгоритмы компьютерного зрения, можно выделить дополненную виртуальность. Внедрение дополненной виртуальности позволит игрокам самостоятельно создавать игровые уровни.



Рис. 1. Схема модели информационной системы Photorun.

Целью данной работы является разработка языка дополненной виртуальности, правил генерации его предложений и построения по ним игровых уровней на примере реализованной автором информационной системы игрового приложения в жанре 2d платформер.

Термин дополненная виртуальность имеет прикладной характер и его определение может меняться в зависимости от области исследования. Считается, что понятие дополненной виртуальности впервые дано в работе П. Милграма и Ф. Кишино в 1994 в работе «A taxonomy of mixed reality visual displays» [1]. В рамках континуума «реальность-виртуальность» оно занимает место, более близкое к виртуальной реальности. Впервые данная теория была воплощена в проекте «Окна на мир» («Windows on the World») [2]. Его суть заключается в построении перемещающимся роботом виртуальной модели реальной местности. В свою очередь более поздний проект «3D виртуальная среда для архитектурной коллаборации» («A Layer-based 3D Virtual Environment for Architectural Collaboration») [3] показал, что рассматриваемая технология может быть применена в сфере строительства и архитектуры.

Рассматривая дополненную виртуальность с точки зрения игровых приложений и их моделей, можно дать следующее определение. Дополненная виртуальность — это совокупность технических средств, алгоритмов компьютерного зрения и обработки растровых изображений, направленная на построение двухмерных или трехмерных виртуальных игровых уровней со своим строгим описанием физики, на основе фотографий или видео пользователя [4]. Основная цель дополненной виртуальности в широком смысле — создание игрового уровня путем расположения игровых компонен-

тов на фоне объектов реального мира, изображенных на распознаваемой фотографии. Как правило, в теории распознавания образов широкий круг задач, целью которого, на первый взгляд, является распознавание изображения, на самом деле сводится к задаче выявления примитивов заданного типа на бинарных изображениях: прямых отрезков, окружностей, треугольников и прямоугольников. Так, цель дополненной виртуальности может быть уточнена: создание игрового уровня на основе бинарного представления исходного изображения путем выявления на нем геометрических объектов заданного типа и расположение игровых компонентов в точках, совпадающих с координатами их начала. Данное определение и цель были сформулированы и использованы в ходе разработки клиент-серверной информационной системы игрового приложения Photorun для Windows Phone и Windows

Рассматривая реализацию данной технологии в контексте проекта Photorun можно построить универсальную игровую модель — клиент-серверную информационную систему с дополненной виртуальностью, в которой входными данными для сервера, в роли которого выступает удаленная служба Windows Communication Foundation, является массив байт, полученный из фотографии пользователя, а выходными данными является строка, описывающая сгенерированный игровой уровень. Схема игровой модели, её основные компоненты и их взаимосвязь показаны на рисунке 1.

Для конечного пользователя видимыми компонентами дополненной виртуальности являются генерация уровней и игровой процесс, а не удаленное распознавание и описание виртуальных миров. Поэтому геймплей использует не только основные принципы любо-



Рис. 2. Фрагмент игрового уровня

го платформера, но имеет отличительную особенность, которой является динамический вектор силы тяжести на игровом уровне. При помощи простых действий пользователь может моментально менять направление вектора силы тяжести на 90 градусов. Такой сценарий позволяет игрокам исследовать труднодоступные части игровых локаций и позволяет творчески подойти к прохождению каждого уровня. Построение же виртуального мира по формальному описанию является ключевым этапом для достижения упомянутой выше цели дополненной виртуальности. Рассмотрим процесс построения игрового уровня подробнее.

Игровые миры создаются на основе строки, которую возвращает удаленная служба. Рассматривая содержимое данной строки с точки зрения теории формальных языков можно сказать, что она содержит набор слов авторского языка дополненной виртуальности L над алфавитом Σ . Каждое слово языка $L(\Sigma)$ описывает правила создания игровых объектов в рамках игрового мира. Фактически каждая фотография представляет собой уникальное графическое описание конечного набора цепочек над этим языком. Для того чтобы определить алфавит и язык дополненной виртуальности сперва опишем общие шаги выявления примитивов заданного типа и описания игрового уровня:

1. Выделение отрезков на бинарном представлении исходного изображения,
2. Поиск координат отрезков, длина которых кратна ширине платформы в пикселях,
3. Добавление в описание игрового уровня информации о координатах и цвете платформ на основе 2 шага,

4. Расстановка монет над платформами,
5. Добавление в описание игрового уровня информации о координатах монет на основе 4 шага.

Информация об игровых объектах платформа и монета, обозначаемых английскими буквами p и s соответственно, содержит не только их координаты на игровом уровне, но и данные о спрайте для придания игре особой атмосферы. Спрайт каждой платформы выбирается в зависимости от преобладающего «под» платформой цвета (по каналам RGB), для чего используются буквы r , g или b . Параметры объектов разделяются запятой, а описание каждого объекта отделяется точкой с запятой. То есть, запись «17,26, p , b » означает, что на игровом уровне в точке с координатами (17,26) будет добавлена платформа со спрайтом, который соответствует синему цвету. Таким образом, для описания игровых уровней используется алфавит $\Sigma = \{\epsilon, '0', '1', '2', '3', '4', '5', '6', '7', '8', '9', ',', 'p', 'c', 'r', 'g', 'b', '\}$. Уточним язык: $L = \{w \mid \text{слова, начинающиеся с двух разделенных запятой целых неотрицательных чисел, содержащие через запятую букву s или разделенные запятой буквы p и r , g или b , и заканчивающиеся точкой с запятой}\}$. На рисунке 2 показан фрагмент игрового уровня, который создан в соответствии с полученным описанием, на основе символов алфавита Σ .

Исходя из перечисленных правил, очевидно, что не любая цепочка символов алфавита Σ принадлежит искомому языку $L(\Sigma)$. Для формализации процесса получения и верификации слов языка сначала необходимо задать грамматику (по Хомскому), затем описать конеч-

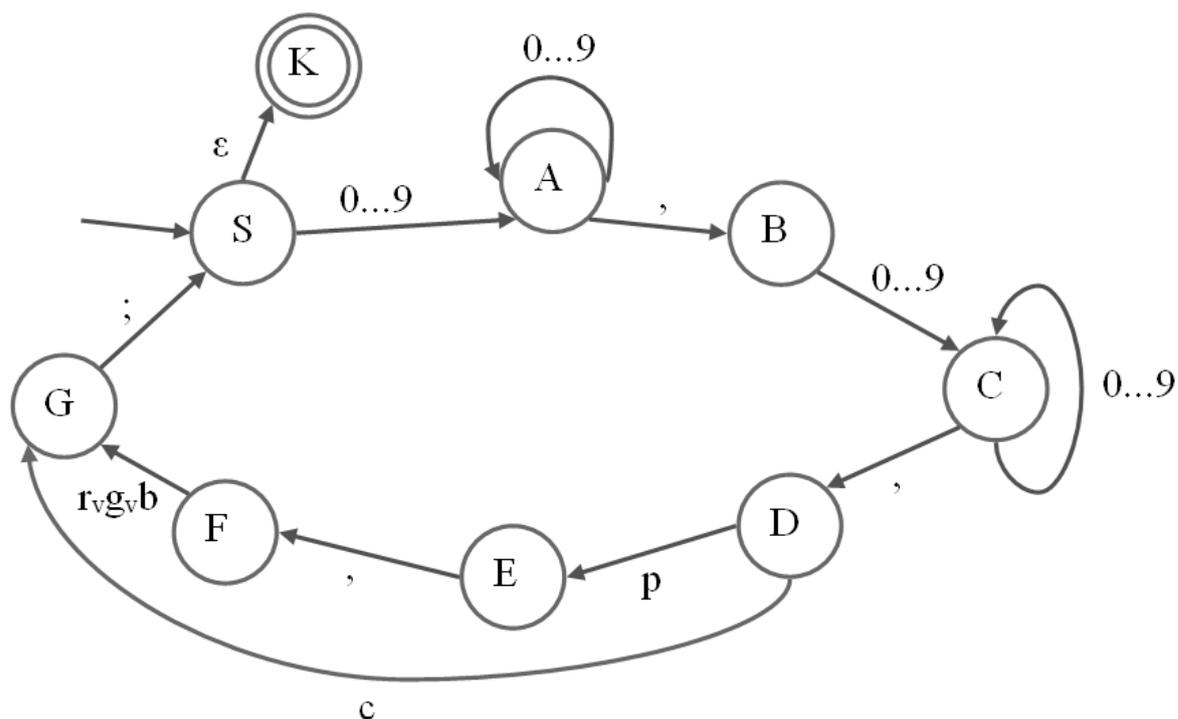


Рис. 3. Граф автомата А

ный автомат или регулярное выражение и синтаксическую диаграмму. Это позволит построить семантическую процедуру, для которой необходимы распознаватель языка или лексический и синтаксический анализаторы[5]. Главной задачей семантической процедуры является генерации игрового мира на устройстве конечного пользователя. В силу того, что обрабатываемые изображения в общем случае могут иметь любой размер, нельзя описать все возможные слова данного языка, т.к. он содержит бесконечное число цепочек разной длины. Поэтому задача построения всех возможных сочетаний координат и типов объектов определенных цветов не будет рассматриваться.

Рассмотрим элементы порождающей грамматики $G(T, N, P, S)$. С точки зрения порождающих грамматик элементы алфавита Σ являются терминальными символами, т.е. $T = \Sigma$. В таком случае множество нетерминальных символов $N = \{S, A, B, C, D, E, F, G\}$, где S , очевидно, является целевым символом. Элементы множества правил вывода P перечислены ниже:

- ◆ $S \rightarrow \epsilon | 0A | 1A | 2A | 3A | 4A | 5A | 6A | 7A | 8A | 9A$
- ◆ $A \rightarrow , B | 0A | 1A | 2A | 3A | 4A | 5A | 6A | 7A | 8A | 9A$
- ◆ $B \rightarrow 0C | 1C | 2C | 3C | 4C | 5C | 6C | 7C | 8C | 9C$
- ◆ $C \rightarrow , D | 0C | 1C | 2C | 3C | 4C | 5C | 6C | 7C | 8C | 9C$
- ◆ $D \rightarrow pE | cG$
- ◆ $E \rightarrow , F$
- ◆ $F \rightarrow rG | gG | bG$
- ◆ $G \rightarrow ; S$

Очевидно, что грамматика G отвечает условиям 3 типа грамматик по Хомскому, причем она является праволинейной. Соответственно язык $L(\Sigma)$ является автоматным. Для программной реализации синтаксического анализатора удобнее использовать конечный автомат[6] или синтаксическую диаграмму. На рисунке 3 изображен граф автомата A , построенного по грамматике G и допускающего язык $L(\Sigma)$.

Вершина K и дуга (S, K) отображают конечное состояние автомата A и переход в него под действием входного символа ϵ . Для наглядности переходов на графе не отображено состояние ошибки — состояние, специально вводимое при моделировании автомата-распознавателя, в которое осуществляется переход при подаче непредвиденного входного символа. Очевидно, что требования к генерируемому на удаленном сервисе описанию и продукции грамматики G составлены таким образом, что построенный автомат является детерминированным. Это освобождает нас от применения теоремы Клини для построения конечного детерминированного автомата по недетерминированному и упрощает задачу построения распознавателя языка. Очевидно, что для создания распознавателя можно воспользоваться универсальным алгоритмом распознавателя автоматных языков, который реализуем на современных популярных высокоуровневых языках программирования или уже доступен в виде расширений. В этом случае актуальным является вопрос о применимости такой реализации

диаграмма:



игровой объект:



координата:

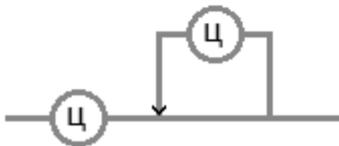


Рис. 4. Синтаксическая диаграмма языка дополненной виртуальности

широким кругом разработчиков, столкнувшихся с проблемой уникальности игр — решение задачи переноса этого алгоритма или применение автоматного стиля программирования и его интеграция с семантической процедурой генерации игрового уровня может иметь высокий порог входа при использовании специальных инструментов для создания игр. Поэтому в качестве альтернативного наглядного способа описания и распознавания автоматного языка $L(\Sigma)$ рассмотрим регулярное выражение и синтаксическую диаграмму. Регулярное выражение имеет вид: $R = ((0+1+2+3+4+5+6+7+8+9)^*(0+1+2+3+4+5+6+7+8+9)^*(p,(r+g+b)+c);)^*$. На рисунке 4 изображена синтаксическая диаграмма, эквивалентная описанному регулярному выражению.

Диаграмма, изображенная на рисунке 4, состоит из трех частей. Первая отражает возможность распознавания повторяющихся последовательностей игровых объектов — слов языка $L(\Sigma)$, разделенных, согласно правилам языка, точкой с запятой. Блок «игровой объект» равносителен правилам описанного языка с учетом вынесения распознавания координат в отдельную часть диаграммы.

Рассмотрим создание игрового уровня с использованием описанной синтаксической диаграммы на примере настольной версии игрового приложения Photorun. Данная версия реализована с помощью GameMaker.

Это среда разработки компьютерных игр, реализующая парадигму событийно — ориентированного и визуального программирования и рассчитанная на создание 2D игр. Игровые уровни в GameMaker являются одним из ключевых игровых ресурсов и называются комнатами. Над комнатами можно совершать ряд заложенных действий, например переход в комнату или выход из нее. А в рамках комнат можно обрабатывать событие создания, выполняя последовательность действий, которые задаются с помощью визуальных инструментов или описываются на встроенном языке программирования GML. Листинг 1 содержит фрагмент программы на языке GML, которая реализует синтаксическую процедуру и семантическую операцию генерации игрового уровня на основе описанного языка $L(\Sigma)$ при загрузке комнаты игрового процесса.

Листинг 1. Фрагмент лексического и синтаксического анализаторов и семантической процедуры:

```
var i = 0, xCoord = 0, yCoord = 0, coord = «»;
while(i++<strLength)
{
    coord = «»;
    while string_char_at(lvlDescription, i) != «,»
        coord += string_char_at(lvlDescription, i++);
    xCoord = real(coord);
    if string(xCoord) != coord break;
```

```

i++;
coord = «»;
while string_char_at(lvlDescription, i) != «,»
    coord += string_char_at(lvlDescription, i++);
yCoord = real(coord);
if string(yCoord) != coord break;
i++;
var objType = string_char_at(lvlDescription, i++);
if objType == «p»
{
    if string_char_at(lvlDescription, i++) != «,»
        break;
    var color = string_char_at(lvlDescription, i++);
    {
        if color == «r» instance_create_
            layer(xCoord*2, yCoord*2, «Instances»,
                obj_cloud_r);
        else if color == «g» instance_create_
            layer(xCoord*2, yCoord*2, «Instances»,
                obj_cloud_g);
        else if color == «b» instance_create_
            layer(xCoord*2, yCoord*2, «Instances»,
                obj_cloud_b);
        else break;
    }
}
else if objType == «c»
    instance_create_layer(xCoord*2, yCoord*2,
        «Instances», obj_coin);
else break;
if string_char_at(lvlDescription, i) != «,»
    break;
}
}

```

Дадим краткое пояснение представленному фрагменту кода, используемым в нем переменным и функциям Game Maker. В цикле с предусловием происходит посим-

вольная обработка строки lvlDescription, содержащей цепочку языка $L(\Sigma)$. Цикл завершит свое выполнение, когда обработана вся строка или встречается неожиданный символ. В переменной i хранится индекс обрабатываемого символа описания игрового уровня. Обращение к символу на позиции i осуществляется с помощью функции string_char_at(). Координаты игрового объекта записываются в переменные действительного типа xCoord и yCoord после преобразования из строковой переменной coord с помощью функций real() и string(). Первая функция используется для получения числа из строки, а вторая для проверки соответствия полученного числа исходной строке. Такая последовательного применения необходима из-за особенности метода real, которая заключается в игнорировании нецифровых символов во входном параметре при преобразовании его в число. Далее по полученным координатам с помощью функции instance_create_layer() в комнату добавляется игровой объект платформа или монета в соответствии со значениями символьной переменной objType, и, в случае обработки платформ, преобладающим цветом.

Таким образом, достигнута основная цель работы, заключающаяся в разработке и демонстрации языка дополненной виртуальности с помощью методов теории формальных языков и конечных автоматов на примере подсистемы генерации игровых уровней игрового приложения. Показана связь и применимость теории автоматов и формальных языков в общем контексте дополненной виртуальности для игровых приложений, а так же мультимедийных программ и их информационных систем. Очевидно, что реализация публичного сервиса, предоставляющего разработчикам казуальных игровых приложений возможность использовать технологию дополненной виртуальности, является актуальным решением существующей проблемы уникальности компьютерных игр.

ЛИТЕРАТУРА

1. Milgram P. Taxonomy of Mixed Reality Visual Displays. / P. Milgram, A. F. Kishino/IEICE Transactions on Information and Systems, Vol E77-D, No.12 December 1994. -P 1321–1329.
2. Simsarian K. T. Windows on the world: An example of augmented virtuality. / K. T. Simsarian, Kristian T. Akesson Karl-Petter// [Электронный ресурс]. — Режим доступа: https://www.researchgate.net/publication/2349157_Windows_on_the_World_An_example_of_Augmented. (дата обращения 05.04.2018)
3. Schmidt S. A Layer-based 3D Virtual Environment for Architectural Collaboration / S. Schmidt, G. Bruder, F. Steinicke // Proceedings of the EuroVR Conference — 2015 [Электронный ресурс]. -Режим доступа: <https://basilic.informatik.uni-hamburg.de/Publications/2015/SBS15/eurovr2015.pdf>. (дата обращения 5.04.2018)
4. Микитчук А. А. Определение и цель дополненной виртуальности в контексте игровых приложений в жанре 2d-платформер. // «Научный диалог: молодой ученый» сборник научных трудов по материалам VII международной научной конференции (г. Самара, 22 июня 2017 г.). Самара: ЦНК МНИФ «Общественная наука», 2017. — С. 6–10.
5. Серебряков В. А. Теория и реализация языков программирования / В. А. Серебряков, М. П. Галочкин, Д. Р. Гончар, М. Г. Фуругян. — М.: МЗ-Пресс, 2006.
6. Гилл А. Введение в теорию конечных автоматов. М.: Наука, 1966.

© Микитчук Андрей Алексеевич (andrew1709@mail.ru).

Журнал «Современная наука: актуальные проблемы теории и практики»