

# Artífice: Generador de ciudades virtuales mediante autómatas celulares. Un enfoque desde los sistemas complejos y auto-organizantes

Alexzander Arriaga Martinez

**Asesores:**

Dr. Abraham Sánchez López  
(FCC-BUAP)

Dr. Carlos Gershenson  
(IIMAS-UNAM)

Dr. Isaac J. Rudomín Goldberg  
(ITESM-CEM)

15 de julio de 2011

## Agenda

Presentación  
Marco teórico  
Planteamiento del problema  
Estado del Arte  
Análisis del problema  
Propuesta de solución  
Implementación  
Resultados obtenidos  
Conclusiones

# Agenda

- 1 **Presentación**
- 2 Marco teórico
- 3 Planteamiento del problema
- 4 Estado del arte
- 5 Análisis del problema
- 6 Propuesta de solución
- 7 Implementacion
- 8 Resultados obtenidos
- 9 Conclusiones

## Agenda

Presentación  
Marco teórico  
Planteamiento del problema  
Estado del Arte  
Análisis del problema  
Propuesta de solución  
Implementación  
Resultados obtenidos  
Conclusiones

# Agenda

- 1 Presentación
- 2 Marco teórico
- 3 Planteamiento del problema
- 4 Estado del arte
- 5 Análisis del problema
- 6 Propuesta de solución
- 7 Implementacion
- 8 Resultados obtenidos
- 9 Conclusiones



## Agenda

Presentación  
Marco teórico  
Planteamiento del problema  
Estado del Arte  
Análisis del problema  
Propuesta de solución  
Implementación  
Resultados obtenidos  
Conclusiones

# Agenda

- 1 Presentación
- 2 Marco teórico
- 3 Planteamiento del problema
- 4 Estado del arte
- 5 Análisis del problema
- 6 Propuesta de solución
- 7 Implementacion
- 8 Resultados obtenidos
- 9 Conclusiones



## Agenda

Presentación  
Marco teórico  
Planteamiento del problema  
Estado del Arte  
Análisis del problema  
Propuesta de solución  
Implementación  
Resultados obtenidos  
Conclusiones

# Agenda

- 1 Presentación
- 2 Marco teórico
- 3 Planteamiento del problema
- 4 Estado del arte
- 5 Análisis del problema
- 6 Propuesta de solución
- 7 Implementacion
- 8 Resultados obtenidos
- 9 Conclusiones



## Agenda

Presentación  
Marco teórico  
Planteamiento del problema  
Estado del Arte  
Análisis del problema  
Propuesta de solución  
Implementación  
Resultados obtenidos  
Conclusiones

# Agenda

- 1 Presentación
- 2 Marco teórico
- 3 Planteamiento del problema
- 4 Estado del arte
- 5 Análisis del problema
- 6 Propuesta de solución
- 7 Implementacion
- 8 Resultados obtenidos
- 9 Conclusiones

## Agenda

Presentación  
Marco teórico  
Planteamiento del problema  
Estado del Arte  
Análisis del problema  
Propuesta de solución  
Implementación  
Resultados obtenidos  
Conclusiones

# Agenda

- 1 Presentación
- 2 Marco teórico
- 3 Planteamiento del problema
- 4 Estado del arte
- 5 Análisis del problema
- 6 Propuesta de solución
- 7 Implementacion
- 8 Resultados obtenidos
- 9 Conclusiones

## Agenda

Presentación  
Marco teórico  
Planteamiento del problema  
Estado del Arte  
Análisis del problema  
Propuesta de solución  
Implementación  
Resultados obtenidos  
Conclusiones

# Agenda

- 1 Presentación
- 2 Marco teórico
- 3 Planteamiento del problema
- 4 Estado del arte
- 5 Análisis del problema
- 6 Propuesta de solución
- 7 Implementacion
- 8 Resultados obtenidos
- 9 Conclusiones



## Agenda

Presentación  
Marco teórico  
Planteamiento del problema  
Estado del Arte  
Análisis del problema  
Propuesta de solución  
Implementación  
Resultados obtenidos  
Conclusiones

# Agenda

- 1 Presentación
- 2 Marco teórico
- 3 Planteamiento del problema
- 4 Estado del arte
- 5 Análisis del problema
- 6 Propuesta de solución
- 7 Implementacion
- 8 Resultados obtenidos
- 9 Conclusiones



## Agenda

Presentación  
Marco teórico  
Planteamiento del problema  
Estado del Arte  
Análisis del problema  
Propuesta de solución  
Implementación  
Resultados obtenidos  
Conclusiones

# Agenda

- 1 Presentación
- 2 Marco teórico
- 3 Planteamiento del problema
- 4 Estado del arte
- 5 Análisis del problema
- 6 Propuesta de solución
- 7 Implementacion
- 8 Resultados obtenidos
- 9 Conclusiones



Agenda

**Presentación**

Marco teórico

Planteamiento del problema

Estado del Arte

Análisis del problema

Propuesta de solución

Implementación

Resultados obtenidos

Conclusiones

Objetivo general

Objetivos específicos

# Presentación



FCC | BUAP

## Objetivo general

Construir un software, llamado *Artífice* para simular el crecimiento de ciudades virtuales en términos de la ocupación de su terreno y con base en ello estudiar el crecimiento poblacional.



## Objetivos específicos

- Utilizar un método sistémico para determinar los componentes, relaciones y estructura de una ciudad.
- Construir un modelo que represente una ciudad, este modelo deberá ser capaz de evolucionar (mediante el uso de Autómatas Celulares).
- Simular el crecimiento de una ciudad en términos de la ocupación de su terreno.
- Implementar un visor 3D de las ciudades generadas.



## Objetivos específicos

- Utilizar un método sistémico para determinar los componentes, relaciones y estructura de una ciudad.
- Construir un modelo que represente una ciudad, este modelo deberá ser capaz de evolucionar (mediante el uso de Autómatas Celulares).
- Simular el crecimiento de una ciudad en términos de la ocupación de su terreno.
- Implementar un visor 3D de las ciudades generadas.

## Objetivos específicos

- Utilizar un método sistémico para determinar los componentes, relaciones y estructura de una ciudad.
- Construir un modelo que represente una ciudad, este modelo deberá ser capaz de evolucionar (mediante el uso de Autómatas Celulares).
- Simular el crecimiento de una ciudad en términos de la ocupación de su terreno.
- Implementar un visor 3D de las ciudades generadas.

## Objetivos específicos

- Utilizar un método sistémico para determinar los componentes, relaciones y estructura de una ciudad.
- Construir un modelo que represente una ciudad, este modelo deberá ser capaz de evolucionar (mediante el uso de Autómatas Celulares).
- Simular el crecimiento de una ciudad en términos de la ocupación de su terreno.
- Implementar un visor 3D de las ciudades generadas.

## Objetivos específicos

- Utilizar un método sistémico para determinar los componentes, relaciones y estructura de una ciudad.
- Construir un modelo que represente una ciudad, este modelo deberá ser capaz de evolucionar (mediante el uso de Autómatas Celulares).
- Simular el crecimiento de una ciudad en términos de la ocupación de su terreno.
- Implementar un visor 3D de las ciudades generadas.

Agenda

Presentación

**Marco teórico**

Planteamiento del problema

Estado del Arte

Análisis del problema

Propuesta de solución

Implementación

Resultados obtenidos

Conclusiones

Comprendiendo a la Complejidad

Autómatas celulares

El juego de la vida

# Marco teórico



FCC | BUAP

## Definición: Complejidad I

Raíz latina *plectere*  $\implies$  “entrelazado” o “entretejido” (Cilliers; 2002). Cualidad de aquello que está formado por múltiples partes, el comportamiento del **todo** no puede determinarse estudiando las **partes** por separado.

### Importante

“La complejidad de un sistema se incrementa con el número de componentes distintos, la cantidad de conexiones entre ellos, las complejidades de los elementos y las complejidades de las conexiones” (Gershenson; 2004).

## Sistemas complejos

Sistemas con múltiples partes; interesa el estudio de las interacciones de éstas y no la particularidad de las mismas.

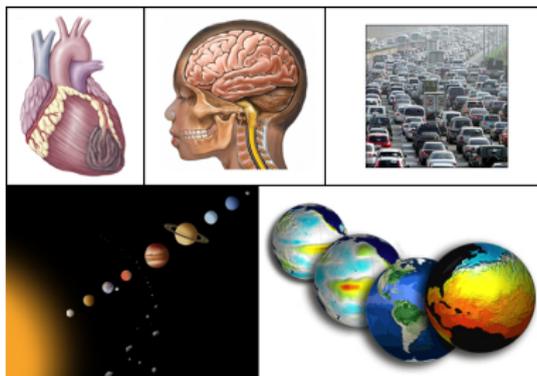


Figura: Ejemplos de sistemas complejos

- Agenda
- Presentación
- Marco teórico
- Planteamiento del problema
- Estado del Arte
- Análisis del problema
- Propuesta de solución
- Implementación
- Resultados obtenidos
- Conclusiones

Comprendiendo a la Complejidad  
**Autómatas celulares**  
El juego de la vida

# Autómatas celulares como motores conceptuales

## Autómata celular

**Modelo** para un sistema dinámico no continuo. Evoluciona a pasos discretos en un *espacio regular infinito*.



FCC | BUAP

## Autómata celular: definición

### Definición

Un Autómata Celular es una 4-tupla:

$$\langle \Sigma, v, \varphi, s_0 \rangle \quad (1)$$

Evolucionando en  $d \in \mathbb{Z}$  donde:

- $\Sigma$  es un conjunto finito de estados.
- $v$  una conexión local tal que  $v = \{X_{i,j,\dots,n;d}\}$ . Llamada también vecindad. Donde  $\{X_{i,j,\dots,n;d}\}$  representa el espacio de evoluciones.
- $\varphi : v \rightarrow \Sigma$  representa la función de transición.
- $s_0$  es la configuración inicial del Autómata Celular.

## Autómata celular: espacio de evoluciones

Un AC puede evolucionar en alguna dimensión, hasta el momento las más utilizadas han sido la 1D, 2D y 3D. Cada uno de estos espacios de evoluciones, se aplican a diferentes problemas.

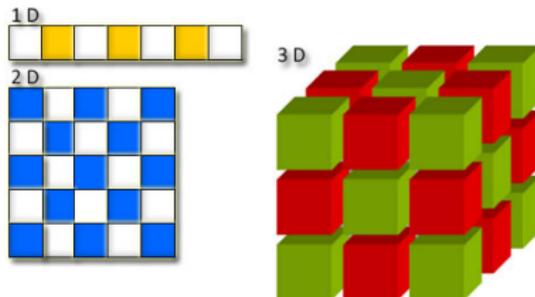


Figura: Espacios de evoluciones más utilizados

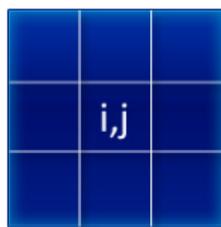


## Autómata celular: estado

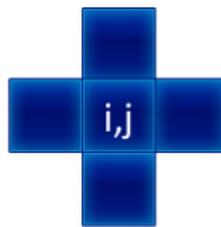
Un **estado**  $s$  es formalmente un elemento  $\sum, (s \in \sum)$  que puede ser asignado a una célula mediante la función de transición. El estado equivale a una configuración única y no ambigua del conjunto de estados de algún autómata celular n-dimensional.

## Autómata celular: vecindad

Vecindades más comunes para AC's bidimensionales: **vecindad de Moore** y **vecindad de Von-Neumann**; llamamos **células vecinas** a aquellas que son inmediatamente adyacentes a una **célula central** (Ilachinski; 2001).



a)



b)

Figura: Vecindades de Moore y Von-Neumann



## Autómata celular: función de transición

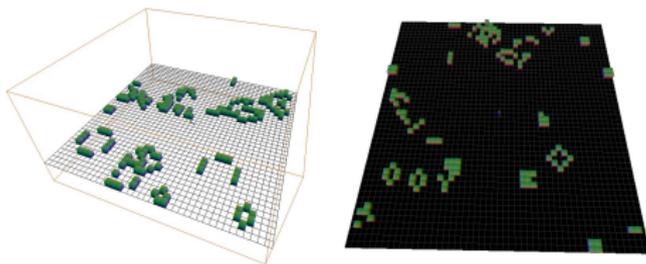
La función de transición  $\varphi : v \rightarrow \Sigma$  se aplica a todas las células del espacio de evoluciones.

La función  $\varphi$  toma como argumentos los valores de la célula en cuestión y los valores de sus vecinos, y regresa el nuevo valor que la célula tendrá en la siguiente generación.

Esta función  $\varphi$  se aplica de forma homogénea a todas las células, por cada paso discreto de tiempo.

## El juego de la vida

*Life*, el mejor ejemplo de autómatas celular bidimensional. Creado por John H. Conway, publicado en 1970 en la columna *Mathematical Recreations* en *Scientific American*. (McIntosh; 1988).



**Figura:** Implementación propia del juego de la vida



Agenda

Presentación

Marco teórico

Planteamiento del problema

Estado del Arte

Análisis del problema

Propuesta de solución

Implementación

Resultados obtenidos

Conclusiones

Comprendiendo a la Complejidad

Autómatas celulares

El juego de la vida

# Reglas del juego de la vida

- **Regla de supervivencia**

Agenda

Presentación

Marco teórico

Planteamiento del problema

Estado del Arte

Análisis del problema

Propuesta de solución

Implementación

Resultados obtenidos

Conclusiones

Comprendiendo a la Complejidad

Autómatas celulares

El juego de la vida

## Reglas del juego de la vida

- **Regla de supervivencia**
- **Regla de nacimiento**



FCC | BUAP

Agenda

Presentación

Marco teórico

Planteamiento del problema

Estado del Arte

Análisis del problema

Propuesta de solución

Implementación

Resultados obtenidos

Conclusiones

Comprendiendo a la Complejidad

Autómatas celulares

El juego de la vida

# Reglas del juego de la vida

- **Regla de supervivencia**
- **Regla de nacimiento**
- **Regla de muerte**



FCC | BUAP

## Reglas del juego de la vida

- **Regla de supervivencia**
- **Regla de nacimiento**
- **Regla de muerte**

Veamos un ejemplo...



- Agenda
- Presentación
- Marco teórico
- Planteamiento del problema**
- Estado del Arte
- Análisis del problema
- Propuesta de solución
- Implementación
- Resultados obtenidos
- Conclusiones

Motivación  
Caso de estudio: planeación en urbanismo

# Planteamiento del problema

# Motivación

- 1 El estudio de las ciudades en términos de crecimiento poblacional resulta un tema relevante en nuestros días.

## Motivación

- 1 El estudio de las ciudades en términos de crecimiento poblacional resulta un tema relevante en nuestros días.
- 2 Los seres humanos hemos llegado casi ya a los 6900 millones de personas.

## Motivación

- 1 El estudio de las ciudades en términos de crecimiento poblacional resulta un tema relevante en nuestros días.
- 2 Los seres humanos hemos llegado casi ya a los 6900 millones de personas.
- 3 Los simuladores para el crecimiento poblacional y generación de ciudades son privados y su código fuente no está disponible para mejorarlos.



## Motivación

- 1 El estudio de las ciudades en términos de crecimiento poblacional resulta un tema relevante en nuestros días.
- 2 Los seres humanos hemos llegado casi ya a los 6900 millones de personas.
- 3 Los simuladores para el crecimiento poblacional y generación de ciudades son privados y su código fuente no está disponible para mejorarlos.
- 4 Existen trabajos realizados con autómatas celulares, sin embargo en México existe poco desarrollo en esta área.



## Motivación

- 1 El estudio de las ciudades en términos de crecimiento poblacional resulta un tema relevante en nuestros días.
- 2 Los seres humanos hemos llegado casi ya a los 6900 millones de personas.
- 3 Los simuladores para el crecimiento poblacional y generación de ciudades son privados y su código fuente no está disponible para mejorarlos.
- 4 Existen trabajos realizados con autómatas celulares, sin embargo en México existe poco desarrollo en esta área.
- 5 Los proyectos que usan autómatas celulares no brindan herramientas de visualización 3D.



## Motivación

- 1 El estudio de las ciudades en términos de crecimiento poblacional resulta un tema relevante en nuestros días.
- 2 Los seres humanos hemos llegado casi ya a los 6900 millones de personas.
- 3 Los simuladores para el crecimiento poblacional y generación de ciudades son privados y su código fuente no está disponible para mejorarlos.
- 4 Existen trabajos realizados con autómatas celulares, sin embargo en México existe poco desarrollo en esta área.
- 5 Los proyectos que usan autómatas celulares no brindan herramientas de visualización 3D.

## Ejemplo de la importancia de la planeación en urbanismo

**CASO:** Nuevo San Juan del Grijalva, Chiapas.

En 2007 las lluvias destruyeron 11 comunidades indígenas el gobierno mexicano crea la **primera ciudad rural sustentable**



**Figura:** Nuevo San Juan del Grijalva



# Estado del Arte

## Grid layout y primitivas geométricas I

Genera una ciudad en tiempo real, p. ej. *UndiscoveredCity* que crea una red de caminos basada en un *grid*. Los edificios se construyen con primitivas geométricas simples.

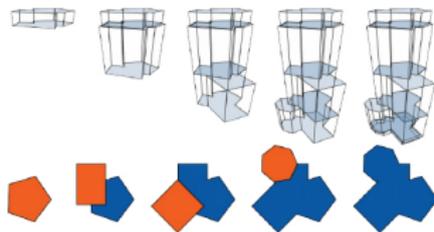


Figura: Construcción de piso y caras de los edificios de *UndiscoveredCity*

## Grid layout y primitivas geométricas II

Las coordenadas de localización de los edificios en el *grid* se usan como semilla para la generación. El texturizado se realiza con 10 imágenes diferentes.



Figura: Ciudad generada con tiempo real con *UndiscoveredCity*



## Sistemas Lindenmayer (*L-Systems*) I

Gramáticas formales utilizadas originalmente por Biólogos para modelar el proceso de crecimiento de plantas. Creadas por **Aristid Lindenmayer**.



**Figura:** Un ejemplo de generación de plantas



## Sistemas Lindenmayer (*L-Systems*) II

Idea central  $\implies$  **sobreescritura**. Define objetos por reemplazo sucesivo de sus partes. Se comienza con un objeto simple y se modifica mediante producciones. Formalmente un *L-System* se define como (Kelly & McCabe; 2003):

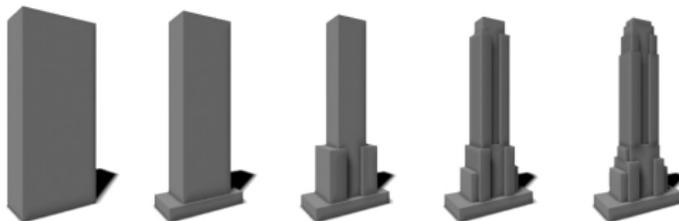


Figura: Edificios generados con *L-Systems*



## Simulación basada agentes

Enfoque propuesto en 2004 (Lechner, Watson, et. al; 2004). *CityBuilder* construido sobre *NetLogo*<sup>TM</sup>. Esta simulación describe comportamientos y semántica (local) con agentes que interactúan con su entorno.

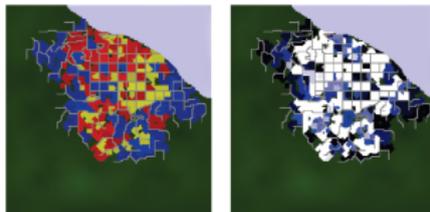
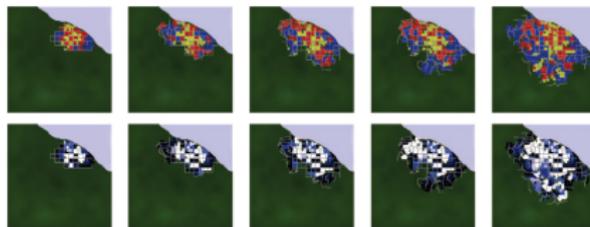


Figura: Simulación de crecimiento con *CityBuilder*

Los agentes base son los **agentes desarrolladores** quienes buscan extensiones de tierra urbanizables, cuando la encuentran “proponen” a la ciudad su construcción y si ésta la autoriza (con base en sus restricciones), se compra la tierra y se construye o mejora la estructura previa.



**Figura:** Una secuencia de fases de desarrollo. Arriba: uso de la tierra. Abajo: densidad de población

## AC's en urbanismo

“La estructura espacial (que además es discreta) de ciudades reales hace que su representación como autómatas celulares sea natural” (Mavroudi; 2007). Las propiedades de una unidad espacial (lote de terreno) se pueden determinar mediante el análisis de sus vecindades.

## Sistemas basados en reglas de urbanismo puro

- Simuladores que generan ciudades hipotéticas.
- Los resultados que generan pueden explicarse mediante reglas de urbanismo.
- Usados para probar nuevas ideas sobre urbanismo.
- Destacan los trabajos de Webster & Wu, 1998 y 2002.



## Sistemas basados en reglas de urbanismo puro

- Simuladores que generan ciudades hipotéticas.
- Los resultados que generan pueden explicarse mediante reglas de urbanismo.
- Usados para probar nuevas ideas sobre urbanismo.
- Destacan los trabajos de Webster & Wu, 1998 y 2002.

## Sistemas basados en reglas de urbanismo puro

- Simuladores que generan ciudades hipotéticas.
- Los resultados que generan pueden explicarse mediante reglas de urbanismo.
- Usados para probar nuevas ideas sobre urbanismo.
- Destacan los trabajos de Webster & Wu, 1998 y 2002.

## Sistemas basados en reglas de urbanismo puro

- Simuladores que generan ciudades hipotéticas.
- Los resultados que generan pueden explicarse mediante reglas de urbanismo.
- Usados para probar nuevas ideas sobre urbanismo.
- Destacan los trabajos de Webster & Wu, 1998 y 2002.

## Sistemas basados en reglas de urbanismo puro

- Simuladores que generan ciudades hipotéticas.
- Los resultados que generan pueden explicarse mediante reglas de urbanismo.
- Usados para probar nuevas ideas sobre urbanismo.
- Destacan los trabajos de Webster & Wu, 1998 y 2002.

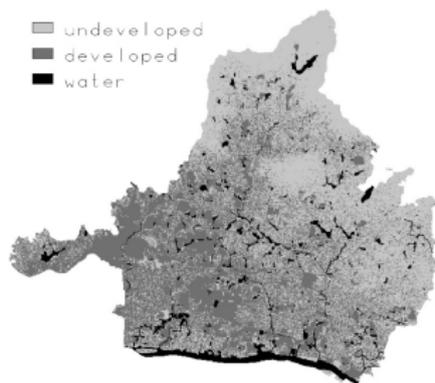


Figura: Ejemplo de generación con el método de Webster & Wu

## Modelos de ciudades reales

- Destacan los trabajos de Clarke & Gaydos 1997 y 1998. Modelos para la Bahía de San Francisco y el corredor de Washington/Baltimore (E.U.A).

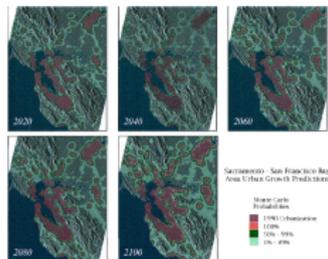


Figura: Ejemplo de generación con el método de Clarke & Gaydos



## Modelos altamente restringidos

- Modelos de planeación normativa.
- Simulan patrones basados en objetivos.
- Pueden ser evaluados para reunir ciertos criterios para desarrollo sustentable (Mavroudi; 2007).
- Yeh & Li 2002 y 2002, usan este tipo de modelos para generar formas urbanas monocéntricas y policéntricas.

## Modelos altamente restringidos

- Modelos de planeación normativa.
- Simulan patrones basados en objetivos.
- Pueden ser evaluados para reunir ciertos criterios para desarrollo sustentable (Mavroudi; 2007).
- Yeh & Li 2002 y 2002, usan este tipo de modelos para generar formas urbanas monocéntricas y policéntricas.

## Modelos altamente restringidos

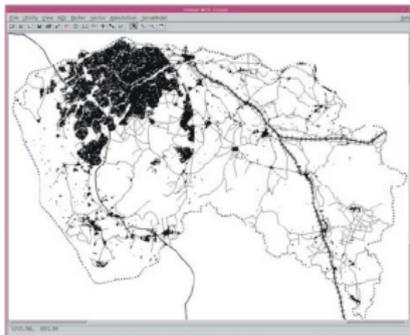
- Modelos de planeación normativa.
- Simulan patrones basados en objetivos.
- Pueden ser evaluados para reunir ciertos criterios para desarrollo sustentable (Mavroudi; 2007).
- Yeh & Li 2002 y 2002, usan este tipo de modelos para generar formas urbanas monocéntricas y policéntricas.

## Modelos altamente restringidos

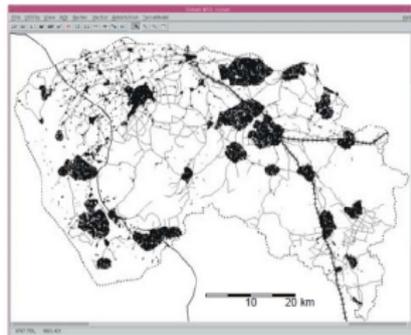
- Modelos de planeación normativa.
- Simulan patrones basados en objetivos.
- Pueden ser evaluados para reunir ciertos criterios para desarrollo sustentable (Mavroudi; 2007).
- Yeh & Li 2002 y 2002, usan este tipo de modelos para generar formas urbanas monocéntricas y policéntricas.

## Modelos altamente restringidos

- Modelos de planeación normativa.
- Simulan patrones basados en objetivos.
- Pueden ser evaluados para reunir ciertos criterios para desarrollo sustentable (Mavroudi; 2007).
- Yeh & Li 2002 y 2002, usan este tipo de modelos para generar formas urbanas monocéntricas y policéntricas.



a)



b)

**Figura:** Ejemplo de generación con el método de Yeh & Li

# Análisis del problema

- Agenda
- Presentación
- Marco teórico
- Planteamiento del problema
- Estado del Arte
- Análisis del problema**
- Propuesta de solución
- Implementación
- Resultados obtenidos
- Conclusiones

## Modelo sistémico

La ciudad y sus componentes

# Modelo sistémico

## Idea general

Modelar al objeto mediante la determinación de sus **componentes**, así como las **relaciones** entre ellos. Esas relaciones determinan la **estructura del objeto**.



## La ciudad y sus componentes



Modelo



- Agenda
  - Presentación
  - Marco teórico
  - Planteamiento del problema
  - Estado del Arte
  - Análisis del problema
  - Propuesta de solución**
  - Implementación
  - Resultados obtenidos
  - Conclusiones

Modelo basado en autómata celular  
Representación de la ciudad como sistema complejo  
Evolución y emergencia en el modelo

# Propuesta de solución

## Modelo basado en autómatas celulares

- **Espacio de evoluciones.** 2D por similitud con las extensiones de tierra.
- **Conjunto de estados.** Cada lote del terreno -célula- puede tener 1 de 7 estados que corresponden con el tipo de construcción.
  - 1 No urbanizado
  - 2 Residencial
  - 3 Empresarial
  - 4 Comercial
  - 5 Zona verde
  - 6 Camino
  - 7 Vacío

## Modelo basado en autómatas celulares

- **Espacio de evoluciones.** 2D por similitud con las extensiones de tierra.
- **Conjunto de estados.** Cada lote del terreno -célula- puede tener 1 de 7 estados que corresponden con el tipo de construcción.
  - 1 No urbanizado
  - 2 Residencial
  - 3 Empresarial
  - 4 Comercial
  - 5 Zona verde
  - 6 Camino
  - 7 Vacío



## Modelo basado en autómatas celulares

- **Espacio de evoluciones.** 2D por similitud con las extensiones de tierra.
- **Conjunto de estados.** Cada lote del terreno -célula- puede tener 1 de 7 estados que corresponden con el tipo de construcción.
  - 1 No urbanizado
  - 2 Residencial
  - 3 Empresarial
  - 4 Comercial
  - 5 Zona verde
  - 6 Camino
  - 7 Vacío



# Modelo basado en autómatas celulares

- **Vecindad.** De Moore.
- **Configuración inicial.** Imagen, texto plano o entrada directa.
- **Función de transición.** Toma la vecindad de Moore y se aplica iterativamente para que la ciudad **evolucione** por  $n$  generaciones. El estado de una célula se determina mediante un **proceso de umbralización** tomando las restricciones dadas por el usuario (restricciones de población).

# Modelo basado en autómatas celulares

- **Vecindad.** De Moore.
- **Configuración inicial.** Imagen, texto plano o entrada directa.
- **Función de transición.** Toma la vecindad de Moore y se aplica iterativamente para que la ciudad **evolucione** por  $n$  generaciones. El estado de una célula se determina mediante un **proceso de umbralización** tomando las restricciones dadas por el usuario (restricciones de población).

## Modelo basado en autómatas celulares

- **Vecindad.** De Moore.
- **Configuración inicial.** Imagen, texto plano o entrada directa.
- **Función de transición.** Toma la vecindad de Moore y se aplica iterativamente para que la ciudad **evolucione** por  $n$  generaciones. El estado de una célula se determina mediante un **proceso de umbralización** tomando las restricciones dadas por el usuario (restricciones de población).

## Modelo basado en autómatas celulares

- **Vecindad.** De Moore.
- **Configuración inicial.** Imagen, texto plano o entrada directa.
- **Función de transición.** Toma la vecindad de Moore y se aplica iterativamente para que la ciudad **evolucione** por  $n$  generaciones. El estado de una célula se determina mediante un **proceso de umbralización** tomando las restricciones dadas por el usuario (restricciones de población).

- Agenda
- Presentación
- Marco teórico
- Planteamiento del problema
- Estado del Arte
- Análisis del problema
- Propuesta de solución**
- Implementación
- Resultados obtenidos
- Conclusiones

Modelo basado en autómatas celulares  
**Representación de la ciudad como sistema complejo**  
Evolución y emergencia en el modelo

## Representación de la ciudad como sistema complejo

Artífice ve a la ciudad como un conjunto de células porciones de tierra. Como en todo sistema complejo **no interesa** estudiar cada lote por separado sino más bien **qué hacen en conjunto** (vecindades).

## Evolución y emergencia en el modelo

Artífice provee **evolución** dado que una ciudad **evoluciona** desde una configuración inicial ( $t = 0$ ) hasta una configuración de **no evolución** ( $t = n$ ) con  $n$  el número de generaciones. La evolución puede detenerse por dos factores:

- 1 Las restricciones impuestas ya no lo permiten (al evaluar la función de transición).
- 2 Se termina el terreno disponible.

## Evolución y emergencia en el modelo

Artífice provee **evolución** dado que una ciudad **evoluciona** desde una configuración inicial ( $t = 0$ ) hasta una configuración de **no evolución** ( $t = n$ ) con  $n$  el número de generaciones. La evolución puede detenerse por dos factores:

- 1 Las restricciones impuestas ya no lo permiten (al evaluar la función de transición).
- 2 Se termina el terreno disponible.



## Evolución y emergencia en el modelo

Artífice provee **evolución** dado que una ciudad **evoluciona** desde una configuración inicial ( $t = 0$ ) hasta una configuración de **no evolución** ( $t = n$ ) con  $n$  el número de generaciones. La evolución puede detenerse por dos factores:

- 1 Las restricciones impuestas ya no lo permiten (al evaluar la función de transición).
- 2 Se termina el terreno disponible.



## Evolución y emergencia en el modelo

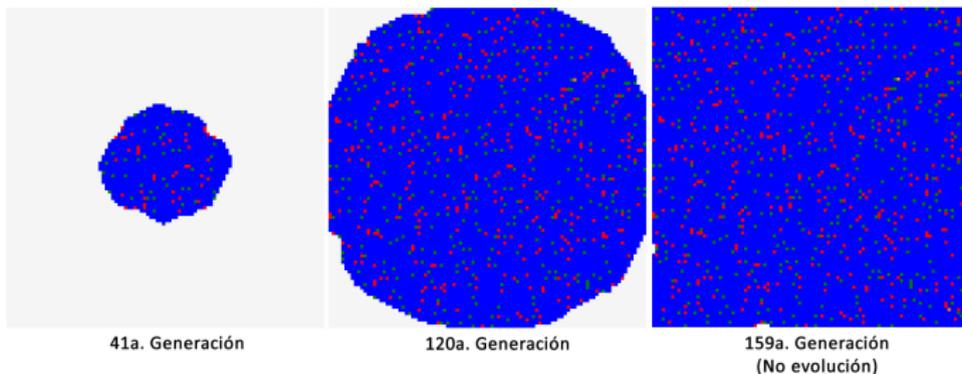


Figura: Ejemplos de no evolución

Agenda  
Presentación  
Marco teórico  
Planteamiento del problema  
Estado del Arte  
Análisis del problema  
Propuesta de solución  
**Implementación**  
Resultados obtenidos  
Conclusiones

Análisis Orientado a Objetos  
Diseño Orientado a Objetos  
Visor 3D  
MEF para renderizado de ciudades  
Renderizado por estado  
Técnicas avanzadas de graficado por computadora  
Artífice como un rompecabezas

# Implementación

Agenda  
Presentación  
Marco teórico  
Planteamiento del problema  
Estado del Arte  
Análisis del problema  
Propuesta de solución  
**Implementación**  
Resultados obtenidos  
Conclusiones

## Análisis Orientado a Objetos

Diseño Orientado a Objetos

Visor 3D

MEF para renderizado de ciudades

Renderizado por estado

Técnicas avanzadas de graficado por computadora

Artífice como un rompecabezas

# Análisis Orientado a Objetos

- 1 Objetos
- 2 Relaciones
- 3 Comportamiento



FCC | BUAP

Agenda  
Presentación  
Marco teórico  
Planteamiento del problema  
Estado del Arte  
Análisis del problema  
Propuesta de solución  
**Implementación**  
Resultados obtenidos  
Conclusiones

## Análisis Orientado a Objetos

Diseño Orientado a Objetos

Visor 3D

MEF para renderizado de ciudades

Renderizado por estado

Técnicas avanzadas de graficado por computadora

Artífice como un rompecabezas

# Análisis Orientado a Objetos

- 1 Objetos
- 2 Relaciones
- 3 Comportamiento



FCC | BUAP

Agenda  
Presentación  
Marco teórico  
Planteamiento del problema  
Estado del Arte  
Análisis del problema  
Propuesta de solución  
**Implementación**  
Resultados obtenidos  
Conclusiones

## Análisis Orientado a Objetos

Diseño Orientado a Objetos

Visor 3D

MEF para renderizado de ciudades

Renderizado por estado

Técnicas avanzadas de graficado por computadora

Artífice como un rompecabezas

# Análisis Orientado a Objetos

- 1 Objetos
- 2 Relaciones
- 3 Comportamiento

Agenda  
Presentación  
Marco teórico  
Planteamiento del problema  
Estado del Arte  
Análisis del problema  
Propuesta de solución  
**Implementación**  
Resultados obtenidos  
Conclusiones

## Análisis Orientado a Objetos

Diseño Orientado a Objetos

Visor 3D

MEF para renderizado de ciudades

Renderizado por estado

Técnicas avanzadas de graficado por computadora

Artífice como un rompecabezas

# Análisis Orientado a Objetos

- 1 Objetos
- 2 Relaciones
- 3 Comportamiento

Agenda  
Presentación  
Marco teórico  
Planteamiento del problema  
Estado del Arte  
Análisis del problema  
Propuesta de solución  
**Implementación**  
Resultados obtenidos  
Conclusiones

Análisis Orientado a Objetos

Diseño Orientado a Objetos

Visor 3D

MEF para renderizado de ciudades

Renderizado por estado

Técnicas avanzadas de graficado por computadora

Artífice como un rompecabezas

## Diseño Orientado a Objetos

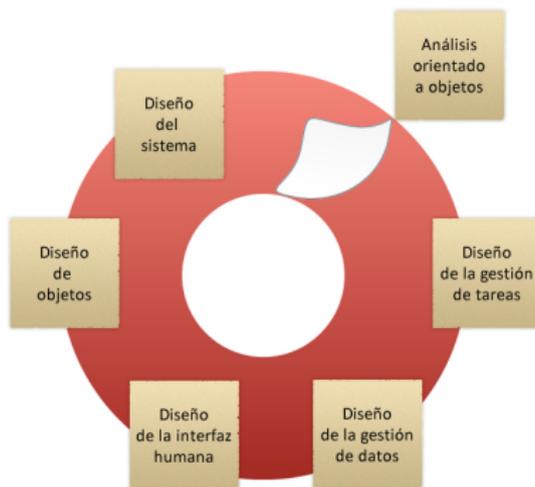


Figura: Flujo del proceso de diseño *Orientado a Objetos*



FCC | BUAP

# Diseño Orientado a Objetos

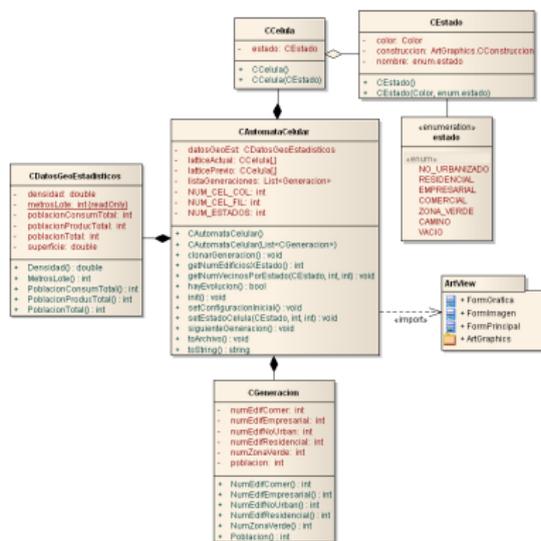


Figura: Diagrama de clases del paquete ArtModel



# Diseño Orientado a Objetos

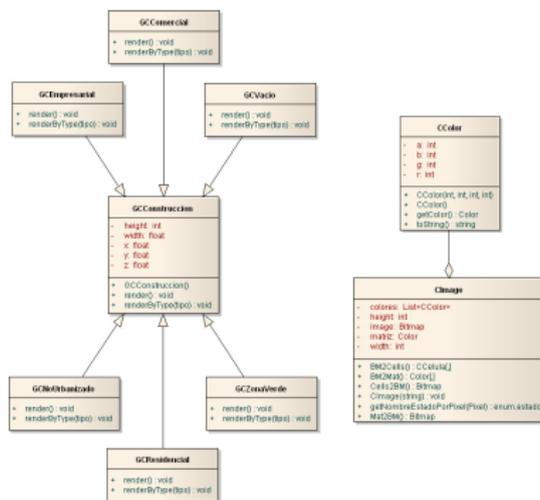


Figura: Diagrama de clases del paquete *ArtGraphics*

# Diseño Orientado a Objetos

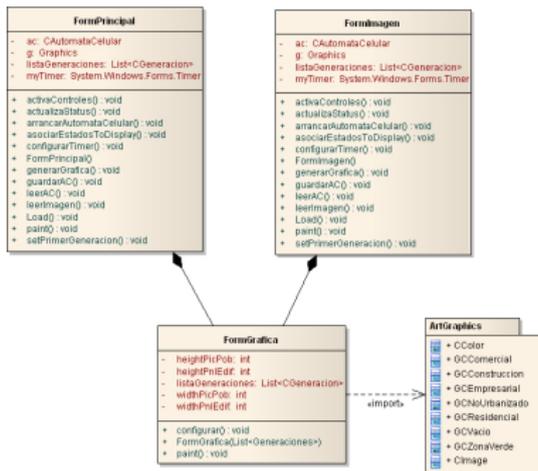


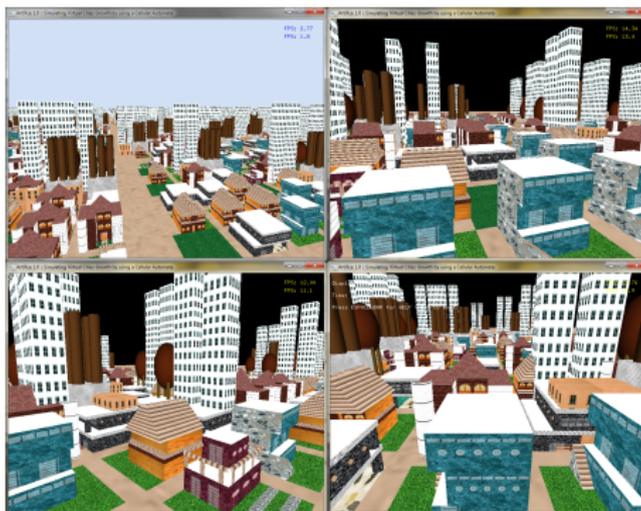
Figura: Diagrama de clases del paquete ArtView



Agenda  
Presentación  
Marco teórico  
Planteamiento del problema  
Estado del Arte  
Análisis del problema  
Propuesta de solución  
**Implementación**  
Resultados obtenidos  
Conclusiones

Análisis Orientado a Objetos  
Diseño Orientado a Objetos  
**Visor 3D**  
MEF para renderizado de ciudades  
Renderizado por estado  
Técnicas avanzadas de graficado por computadora  
Artífice como un rompecabezas

## Visor 3D



**Figura:** Ejemplo de ciudades en el visor 3D



FCC | BUAP

## MEF para renderizado de ciudades

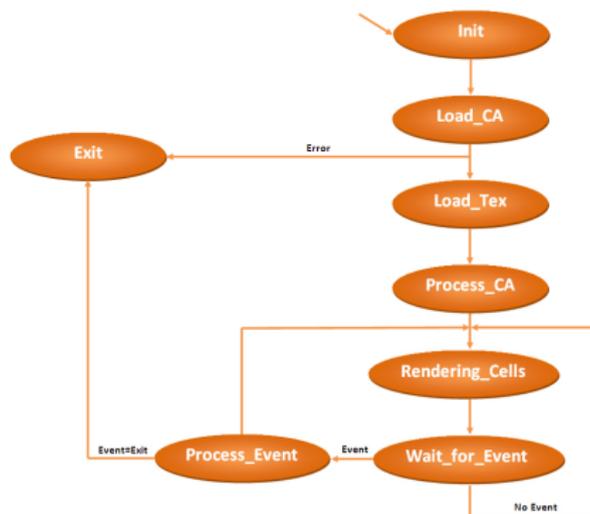
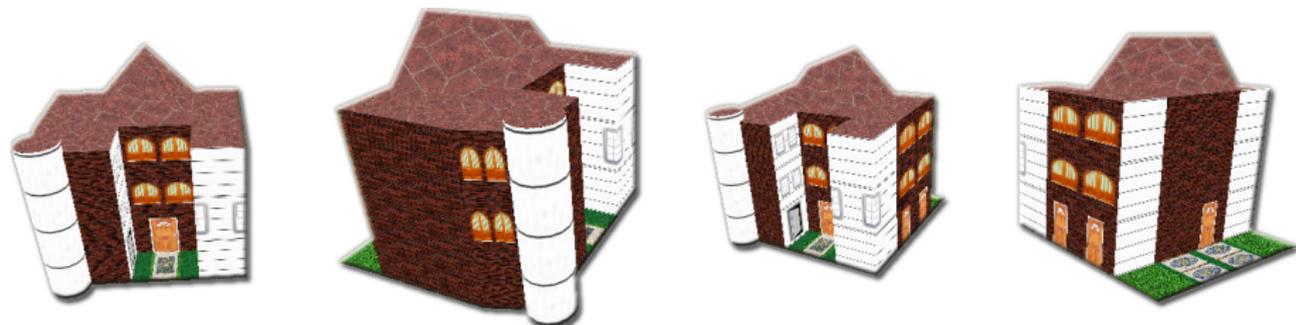


Figura: Máquina de estados finitos para renderizado



## Renderizado por estado

Se tiene un edificio para cada estado (a excepción del estado residencial con cinco posibles edificios).



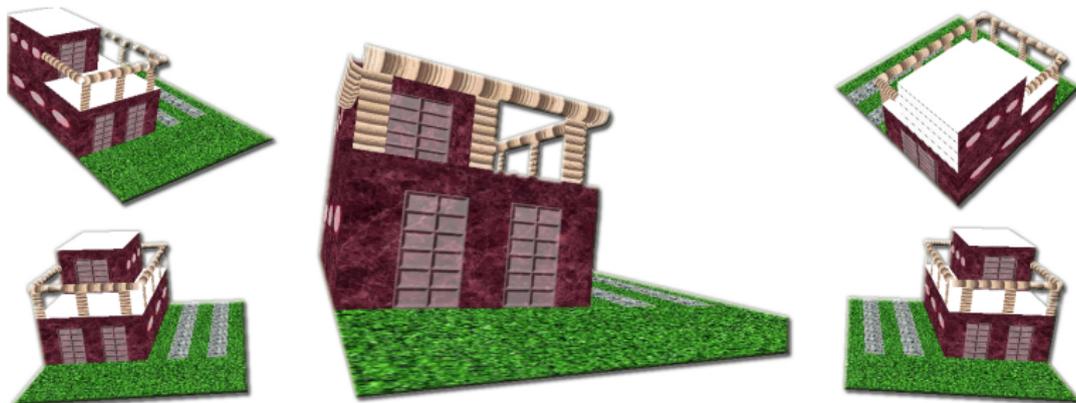
**Figura:** Primer modelo de edificio para estado residencial



Agenda  
Presentación  
Marco teórico  
Planteamiento del problema  
Estado del Arte  
Análisis del problema  
Propuesta de solución  
**Implementación**  
Resultados obtenidos  
Conclusiones

Análisis Orientado a Objetos  
Diseño Orientado a Objetos  
Visor 3D  
MEF para renderizado de ciudades  
**Renderizado por estado**  
Técnicas avanzadas de graficado por computadora  
Artífice como un rompecabezas

## Renderizado para estado residencial



**Figura:** Segundo modelo de edificio para estado residencial



FCC | BUAP

Agenda  
Presentación  
Marco teórico  
Planteamiento del problema  
Estado del Arte  
Análisis del problema  
Propuesta de solución  
**Implementación**  
Resultados obtenidos  
Conclusiones

Análisis Orientado a Objetos  
Diseño Orientado a Objetos  
Visor 3D  
MEF para renderizado de ciudades  
**Renderizado por estado**  
Técnicas avanzadas de graficado por computadora  
Artífice como un rompecabezas

## Renderizado para estado residencial



Figura: Tercer modelo de edificio para estado residencial



FCC | BUAP

Agenda  
Presentación  
Marco teórico  
Planteamiento del problema  
Estado del Arte  
Análisis del problema  
Propuesta de solución  
**Implementación**  
Resultados obtenidos  
Conclusiones

Análisis Orientado a Objetos  
Diseño Orientado a Objetos  
Visor 3D  
MEF para renderizado de ciudades  
**Renderizado por estado**  
Técnicas avanzadas de graficado por computadora  
Artífice como un rompecabezas

## Renderizado para estado residencial



**Figura:** Cuarto modelo de edificio para estado residencial



Agenda  
Presentación  
Marco teórico  
Planteamiento del problema  
Estado del Arte  
Análisis del problema  
Propuesta de solución  
**Implementación**  
Resultados obtenidos  
Conclusiones

Análisis Orientado a Objetos  
Diseño Orientado a Objetos  
Visor 3D  
MEF para renderizado de ciudades  
**Renderizado por estado**  
Técnicas avanzadas de graficado por computadora  
Artífice como un rompecabezas

## Renderizado para estado residencial

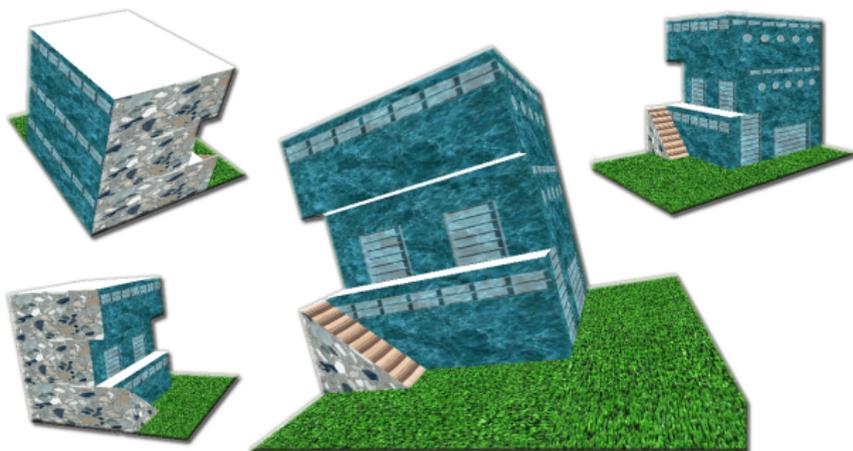


Figura: Quinto modelo de edificio para estado residencial

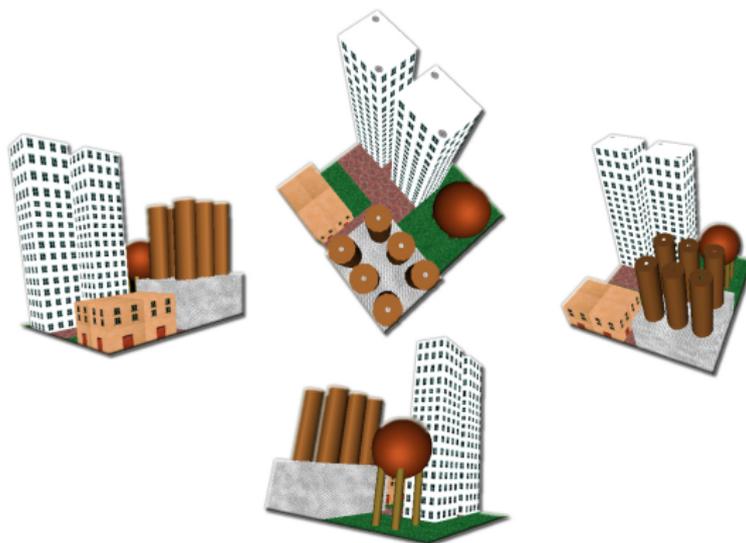


FCC | BUAP

Agenda  
Presentación  
Marco teórico  
Planteamiento del problema  
Estado del Arte  
Análisis del problema  
Propuesta de solución  
**Implementación**  
Resultados obtenidos  
Conclusiones

Análisis Orientado a Objetos  
Diseño Orientado a Objetos  
Visor 3D  
MEF para renderizado de ciudades  
**Renderizado por estado**  
Técnicas avanzadas de graficado por computadora  
Artífice como un rompecabezas

## Renderizado para estado empresarial



**Figura:** Modelo de edificio para estado empresarial

Alexander Arriaga Martinez

Artífice: Generador de ciudades virtuales mediante autómatas cel

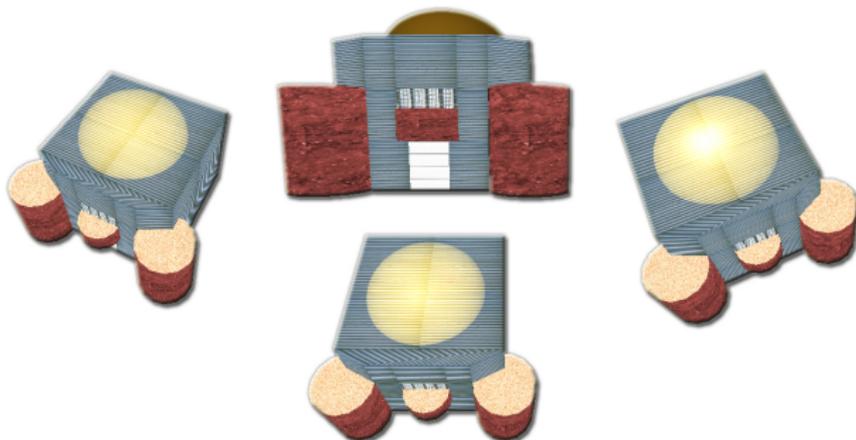


FCC | BUAP

Agenda  
Presentación  
Marco teórico  
Planteamiento del problema  
Estado del Arte  
Análisis del problema  
Propuesta de solución  
**Implementación**  
Resultados obtenidos  
Conclusiones

Análisis Orientado a Objetos  
Diseño Orientado a Objetos  
Visor 3D  
MEF para renderizado de ciudades  
**Renderizado por estado**  
Técnicas avanzadas de graficado por computadora  
Artífice como un rompecabezas

## Renderizado para estado comercial

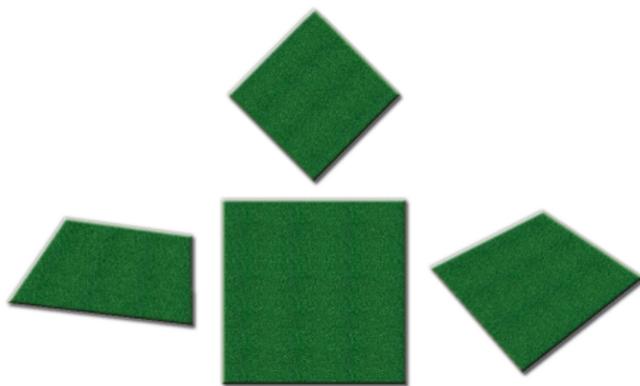


**Figura:** Modelo de edificio para estado comercial

Agenda  
Presentación  
Marco teórico  
Planteamiento del problema  
Estado del Arte  
Análisis del problema  
Propuesta de solución  
**Implementación**  
Resultados obtenidos  
Conclusiones

Análisis Orientado a Objetos  
Diseño Orientado a Objetos  
Visor 3D  
MEF para renderizado de ciudades  
**Renderizado por estado**  
Técnicas avanzadas de graficado por computadora  
Artífice como un rompecabezas

## Renderizado para estado zona verde



**Figura:** Modelo para estado zona verde

Agenda  
Presentación  
Marco teórico  
Planteamiento del problema  
Estado del Arte  
Análisis del problema  
Propuesta de solución  
**Implementación**  
Resultados obtenidos  
Conclusiones

Análisis Orientado a Objetos  
Diseño Orientado a Objetos  
Visor 3D  
MEF para renderizado de ciudades  
**Renderizado por estado**  
Técnicas avanzadas de graficado por computadora  
Artífice como un rompecabezas

## ¿Por qué OpenGL?

- 1 Interfaz directa con el hardware.
- 2 Programación “desde cero” implementación tan eficiente como se requiera.
- 3 Estándar a nivel mundial para generación de aplicaciones 3D.
- 4 Estándar abierto (se puede contribuir a su desarrollo).
- 5 Independiente del hardware.
- 6 Multiplataforma (Shreiner; 2010).



## ¿Por qué OpenGL?

- 1 Interfaz directa con el hardware.
- 2 Programación “desde cero” implementación tan eficiente como se requiera.
- 3 Estándar a nivel mundial para generación de aplicaciones 3D.
- 4 Estándar abierto (se puede contribuir a su desarrollo).
- 5 Independiente del hardware.
- 6 Multiplataforma (Shreiner; 2010).

Agenda  
Presentación  
Marco teórico  
Planteamiento del problema  
Estado del Arte  
Análisis del problema  
Propuesta de solución  
**Implementación**  
Resultados obtenidos  
Conclusiones

Análisis Orientado a Objetos  
Diseño Orientado a Objetos  
Visor 3D  
MEF para renderizado de ciudades  
**Renderizado por estado**  
Técnicas avanzadas de graficado por computadora  
Artífice como un rompecabezas

## ¿Por qué OpenGL?

- 1 Interfaz directa con el hardware.
- 2 Programación “desde cero” implementación tan eficiente como se requiera.
- 3 Estándar a nivel mundial para generación de aplicaciones 3D.
- 4 Estándar abierto (se puede contribuir a su desarrollo).
- 5 Independiente del hardware.
- 6 Multiplataforma (Shreiner; 2010).

Agenda  
Presentación  
Marco teórico  
Planteamiento del problema  
Estado del Arte  
Análisis del problema  
Propuesta de solución  
**Implementación**  
Resultados obtenidos  
Conclusiones

Análisis Orientado a Objetos  
Diseño Orientado a Objetos  
Visor 3D  
MEF para renderizado de ciudades  
**Renderizado por estado**  
Técnicas avanzadas de graficado por computadora  
Artífice como un rompecabezas

## ¿Por qué OpenGL?

- 1 Interfaz directa con el hardware.
- 2 Programación “desde cero” implementación tan eficiente como se requiera.
- 3 Estándar a nivel mundial para generación de aplicaciones 3D.
- 4 Estándar abierto (se puede contribuir a su desarrollo).
- 5 Independiente del hardware.
- 6 Multiplataforma (Shreiner; 2010).

## ¿Por qué OpenGL?

- 1 Interfaz directa con el hardware.
- 2 Programación “desde cero” implementación tan eficiente como se requiera.
- 3 Estándar a nivel mundial para generación de aplicaciones 3D.
- 4 Estándar abierto (se puede contribuir a su desarrollo).
- 5 Independiente del hardware.
- 6 Multiplataforma (Shreiner; 2010).

## ¿Por qué OpenGL?

- 1 Interfaz directa con el hardware.
- 2 Programación “desde cero” implementación tan eficiente como se requiera.
- 3 Estándar a nivel mundial para generación de aplicaciones 3D.
- 4 Estándar abierto (se puede contribuir a su desarrollo).
- 5 Independiente del hardware.
- 6 Multiplataforma (Shreiner; 2010).

## ¿Por qué OpenGL?

- 1 Interfaz directa con el hardware.
- 2 Programación “desde cero” implementación tan eficiente como se requiera.
- 3 Estándar a nivel mundial para generación de aplicaciones 3D.
- 4 Estándar abierto (se puede contribuir a su desarrollo).
- 5 Independiente del hardware.
- 6 Multiplataforma (Shreiner; 2010).

## Técnicas avanzadas de graficado por computadora

- Manejo de nivel de detalle.
- Uso de *display lists*.

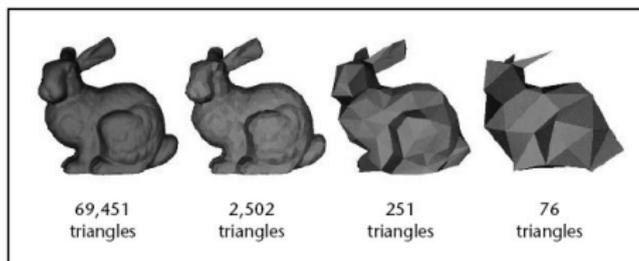


Figura: Ejemplo de manejo de nivel de detalle

## Técnicas avanzadas de graficado por computadora

- Manejo de nivel de detalle.
- Uso de *display lists*.

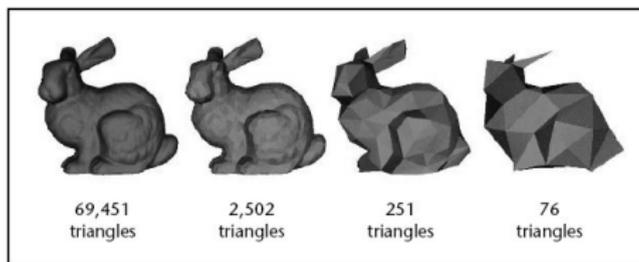


Figura: Ejemplo de manejo de nivel de detalle

## Técnicas avanzadas de graficado por computadora

- Manejo de nivel de detalle.
- Uso de *display lists*.

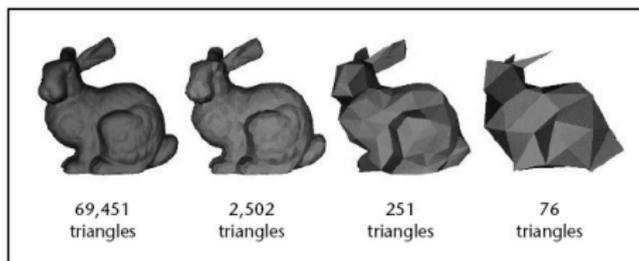


Figura: Ejemplo de manejo de nivel de detalle

# Habitantes

Dos clases de población:

- 1 Pob. productora
- 2 Pob. consumidora

Agenda  
Presentación  
Marco teórico  
Planteamiento del problema  
Estado del Arte  
Análisis del problema  
Propuesta de solución  
**Implementación**  
Resultados obtenidos  
Conclusiones

Análisis Orientado a Objetos  
Diseño Orientado a Objetos  
Visor 3D  
MEF para renderizado de ciudades  
Renderizado por estado  
Técnicas avanzadas de graficado por computadora  
**Artífice como un rompecabezas**

## Construcciones

Diversos tipos dependiendo del estado de la célula actual. En visor 2D se representan con colores.

Color	Estado	RGB	Hexadecimal
	No urbanizado	(255, 255, 255)	#FFFFFF
	Residencial	(0, 0, 255)	#0000FF
	Empresarial	(255, 0, 0)	#FF0000
	Comercial	(255, 255, 0)	#FFFF00
	Zona Verde	(0, 255, 0)	#00FF00
	Camino	(0, 0, 0)	#000000
	Vacio	(204, 204, 204)	#CCCCCC

Figura: Colores propuestos para la versión 2D de *Artífice*



## Vialidades

La red de caminos se toma como una cuadrícula estática (puede darse con una imagen).

## Terreno

La versión inicial es de 100 lotes por lado en un *grid*. Cada lote tiene una extensión de 100m por lado. Cuando se carga una imagen el terreno se toma del tamaño de ésta (w,h).

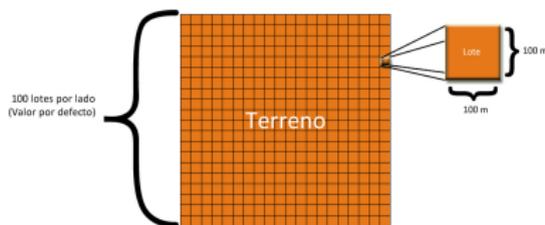


Figura: Terreno de *Artífice*

## Entrada I

- Archivos de texto.
- Imágenes (PNG, JPEG, BMP,...).

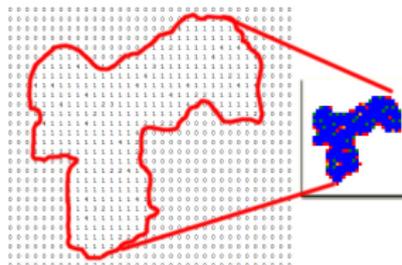
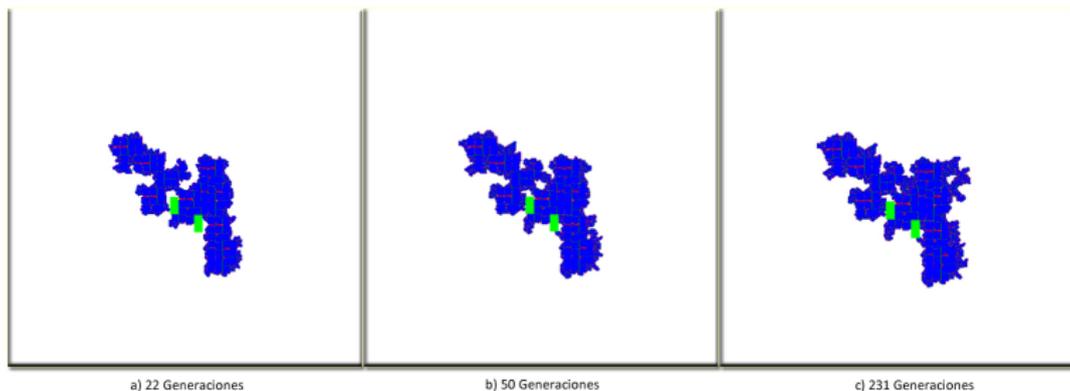


Figura: Ejemplo de archivo de texto de entrada



## Entrada II



**Figura:** Ejemplo de mapa de entrada y su evolución



Agenda  
Presentación  
Marco teórico  
Planteamiento del problema  
Estado del Arte  
Análisis del problema  
Propuesta de solución  
Implementación  
**Resultados obtenidos**  
Conclusiones

Ejemplos de ejecución  
Alcances y limitaciones  
Limitaciones del software  
Limitaciones del visor 2D y 3D

# Resultados obtenidos

Agenda  
Presentación  
Marco teórico  
Planteamiento del problema  
Estado del Arte  
Análisis del problema  
Propuesta de solución  
Implementación  
**Resultados obtenidos**  
Conclusiones

## Ejemplos de ejecución

Alcances y limitaciones  
Limitaciones del software  
Limitaciones del visor 2D y 3D

# Ejemplos de ejecución I

Agenda  
Presentación  
Marco teórico  
Planteamiento del problema  
Estado del Arte  
Análisis del problema  
Propuesta de solución  
Implementación  
Resultados obtenidos  
Conclusiones

Ejemplos de ejecución  
Alcances y limitaciones  
Limitaciones del software  
Limitaciones del visor 2D y 3D

## Ejemplos de ejecución II

RC	RE	RZV	No. Gen.
100	18	10	57
330	20	15	296
495	20	20	298
625	16	16	656
692	18	20	210
725	25	25	301
825	20	16	151
856	20	25	397
1054	22	35	273
1054	30	40	221

Cuadro: Estadísticas de diez ciudades con restricciones variables



## Ejemplos de ejecución

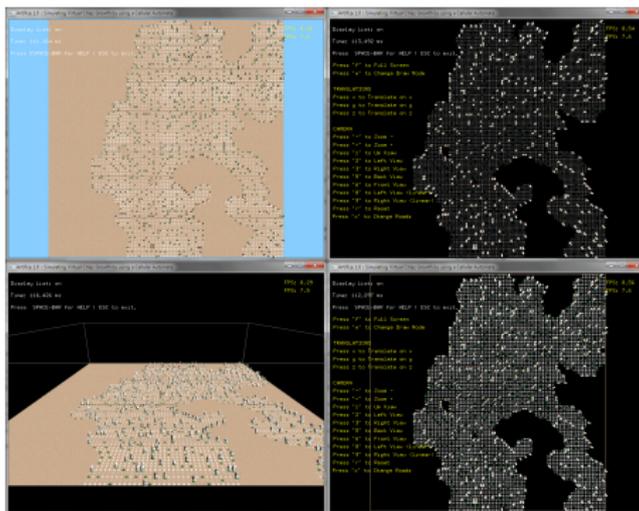


Figura: Primera ciudad arbitraria en *Artifice*

Agenda  
Presentación  
Marco teórico  
Planteamiento del problema  
Estado del Arte  
Análisis del problema  
Propuesta de solución  
Implementación  
Resultados obtenidos  
Conclusiones

Ejemplos de ejecución  
Alcances y limitaciones  
Limitaciones del software  
Limitaciones del visor 2D y 3D

## Ejemplos de ejecución

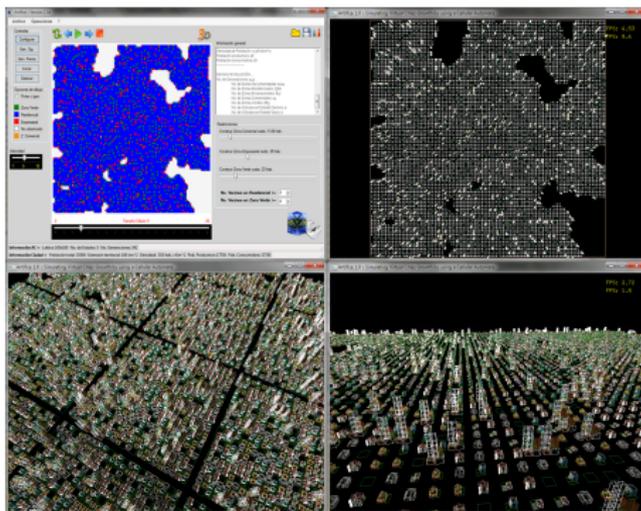


Figura: Segunda ciudad arbitraria en *Artifice*

Agenda  
Presentación  
Marco teórico  
Planteamiento del problema  
Estado del Arte  
Análisis del problema  
Propuesta de solución  
Implementación  
Resultados obtenidos  
Conclusiones

Ejemplos de ejecución  
Alcances y limitaciones  
Limitaciones del software  
Limitaciones del visor 2D y 3D

## Ejemplos de ejecución

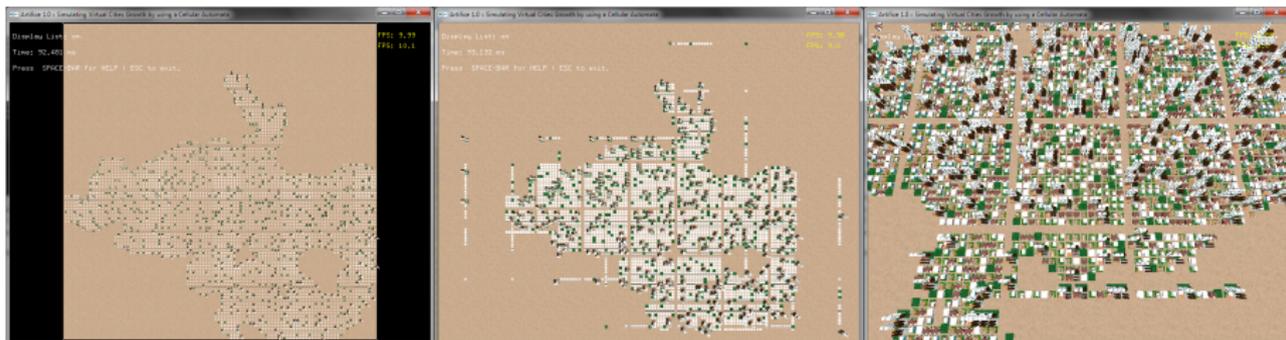


Figura: Tercera ciudad arbitraria en *Artifice*

# Alcances

- 1 Modelo de autómata celular para generación de ciudades hipotéticas
- 2 Implementación sistémica
- 3 Simulación de ocupación del terreno
- 4 Incluye un visor 3D
- 5 Posibilidad de imágenes y archivos de texto plano como entrada
- 6 Siete posibles estados por célula (lote)
- 7 Control paramétrico de reglas de transición
- 8 Manejo de 2 niveles de detalle para estado residencial
- 9 Permite analizar crecimiento poblacional por generación
- 10 Implementación *Orientada a Objetos*



# Alcances

- 1 Modelo de autómata celular para generación de ciudades hipotéticas
- 2 Implementación sistémica
- 3 Simulación de ocupación del terreno
- 4 Incluye un visor 3D
- 5 Posibilidad de imágenes y archivos de texto plano como entrada
- 6 Siete posibles estados por célula (lote)
- 7 Control paramétrico de reglas de transición
- 8 Manejo de 2 niveles de detalle para estado residencial
- 9 Permite analizar crecimiento poblacional por generación
- 10 Implementación *Orientada a Objetos*



# Alcances

- 1 Modelo de autómata celular para generación de ciudades hipotéticas
- 2 Implementación sistémica
- 3 Simulación de ocupación del terreno
- 4 Incluye un visor 3D
- 5 Posibilidad de imágenes y archivos de texto plano como entrada
- 6 Siete posibles estados por célula (lote)
- 7 Control paramétrico de reglas de transición
- 8 Manejo de 2 niveles de detalle para estado residencial
- 9 Permite analizar crecimiento poblacional por generación
- 10 Implementación *Orientada a Objetos*

# Alcances

- 1 Modelo de autómata celular para generación de ciudades hipotéticas
- 2 Implementación sistémica
- 3 Simulación de ocupación del terreno
- 4 Incluye un visor 3D
- 5 Posibilidad de imágenes y archivos de texto plano como entrada
- 6 Siete posibles estados por célula (lote)
- 7 Control paramétrico de reglas de transición
- 8 Manejo de 2 niveles de detalle para estado residencial
- 9 Permite analizar crecimiento poblacional por generación
- 10 Implementación *Orientada a Objetos*

# Alcances

- 1 Modelo de autómata celular para generación de ciudades hipotéticas
- 2 Implementación sistémica
- 3 Simulación de ocupación del terreno
- 4 Incluye un visor 3D
- 5 Posibilidad de imágenes y archivos de texto plano como entrada
- 6 Siete posibles estados por célula (lote)
- 7 Control paramétrico de reglas de transición
- 8 Manejo de 2 niveles de detalle para estado residencial
- 9 Permite analizar crecimiento poblacional por generación
- 10 Implementación *Orientada a Objetos*

# Alcances

- 1 Modelo de autómata celular para generación de ciudades hipotéticas
- 2 Implementación sistémica
- 3 Simulación de ocupación del terreno
- 4 Incluye un visor 3D
- 5 Posibilidad de imágenes y archivos de texto plano como entrada
- 6 Siete posibles estados por célula (lote)
- 7 Control paramétrico de reglas de transición
- 8 Manejo de 2 niveles de detalle para estado residencial
- 9 Permite analizar crecimiento poblacional por generación
- 10 Implementación *Orientada a Objetos*

## Alcances

- 1 Modelo de autómata celular para generación de ciudades hipotéticas
- 2 Implementación sistémica
- 3 Simulación de ocupación del terreno
- 4 Incluye un visor 3D
- 5 Posibilidad de imágenes y archivos de texto plano como entrada
- 6 Siete posibles estados por célula (lote)
- 7 Control paramétrico de reglas de transición
- 8 Manejo de 2 niveles de detalle para estado residencial
- 9 Permite analizar crecimiento poblacional por generación
- 10 Implementación *Orientada a Objetos*

## Alcances

- 1 Modelo de autómata celular para generación de ciudades hipotéticas
- 2 Implementación sistémica
- 3 Simulación de ocupación del terreno
- 4 Incluye un visor 3D
- 5 Posibilidad de imágenes y archivos de texto plano como entrada
- 6 Siete posibles estados por célula (lote)
- 7 Control paramétrico de reglas de transición
- 8 Manejo de 2 niveles de detalle para estado residencial
- 9 Permite analizar crecimiento poblacional por generación
- 10 Implementación *Orientada a Objetos*



## Alcances

- 1 Modelo de autómata celular para generación de ciudades hipotéticas
- 2 Implementación sistémica
- 3 Simulación de ocupación del terreno
- 4 Incluye un visor 3D
- 5 Posibilidad de imágenes y archivos de texto plano como entrada
- 6 Siete posibles estados por célula (lote)
- 7 Control paramétrico de reglas de transición
- 8 Manejo de 2 niveles de detalle para estado residencial
- 9 Permite analizar crecimiento poblacional por generación
- 10 Implementación *Orientada a Objetos*

## Alcances

- 1 Modelo de autómata celular para generación de ciudades hipotéticas
- 2 Implementación sistémica
- 3 Simulación de ocupación del terreno
- 4 Incluye un visor 3D
- 5 Posibilidad de imágenes y archivos de texto plano como entrada
- 6 Siete posibles estados por célula (lote)
- 7 Control paramétrico de reglas de transición
- 8 Manejo de 2 niveles de detalle para estado residencial
- 9 Permite analizar crecimiento poblacional por generación
- 10 Implementación *Orientada a Objetos*



## Limitaciones del software

- 1 **Particionado del terreno**
- 2 Número de estados
- 3 Modificación de estados ya establecidos
- 4 No se utilizan agentes que interactúen con su entorno
- 5 Plataforma de desarrollo
- 6 Sólo se toman siete estados

## Limitaciones del software

- 1 Particionado del terreno
- 2 Número de estados
- 3 Modificación de estados ya establecidos
- 4 No se utilizan agentes que interactúen con su entorno
- 5 Plataforma de desarrollo
- 6 Sólo se toman siete estados

## Limitaciones del software

- 1 Particionado del terreno
- 2 Número de estados
- 3 Modificación de estados ya establecidos
- 4 No se utilizan agentes que interactúen con su entorno
- 5 Plataforma de desarrollo
- 6 Sólo se toman siete estados

## Limitaciones del software

- 1 Particionado del terreno
- 2 Número de estados
- 3 Modificación de estados ya establecidos
- 4 No se utilizan agentes que interactúen con su entorno
- 5 Plataforma de desarrollo
- 6 Sólo se toman siete estados

## Limitaciones del software

- 1 Particionado del terreno
- 2 Número de estados
- 3 Modificación de estados ya establecidos
- 4 No se utilizan agentes que interactúen con su entorno
- 5 Plataforma de desarrollo
- 6 Sólo se toman siete estados



## Limitaciones del software

- 1 Particionado del terreno
- 2 Número de estados
- 3 Modificación de estados ya establecidos
- 4 No se utilizan agentes que interactúen con su entorno
- 5 Plataforma de desarrollo
- 6 Sólo se toman siete estados

## Limitaciones del visor 2D y 3D

- 1 Colores
- 2 Número de estados
- 3 Agregar soporte para *CUDA*
- 4 Uso de *shaders GLSL ó HLSL*
- 5 Generación de multivariada de construcciones por estado
- 6 Texturizado de vialidades
- 7 Objetos móviles: autos, camiones, aviones, etc.
- 8 Renderizado de los habitantes

## Limitaciones del visor 2D y 3D

- 1 Colores
- 2 Número de estados
- 3 Agregar soporte para *CUDA*
- 4 Uso de *shaders GLSL ó HLSL*
- 5 Generación de multivariada de construcciones por estado
- 6 Texturizado de vialidades
- 7 Objetos móviles: autos, camiones, aviones, etc.
- 8 Renderizado de los habitantes

## Limitaciones del visor 2D y 3D

- 1 Colores
- 2 Número de estados
- 3 Agregar soporte para *CUDA*
- 4 Uso de *shaders GLSL ó HLSL*
- 5 Generación de multivariada de construcciones por estado
- 6 Texturizado de vialidades
- 7 Objetos móviles: autos, camiones, aviones, etc.
- 8 Renderizado de los habitantes

## Limitaciones del visor 2D y 3D

- 1 Colores
- 2 Número de estados
- 3 Agregar soporte para *CUDA*
- 4 Uso de *shaders GLSL ó HLSL*
- 5 Generación de multivariada de construcciones por estado
- 6 Texturizado de vialidades
- 7 Objetos móviles: autos, camiones, aviones, etc.
- 8 Renderizado de los habitantes

## Limitaciones del visor 2D y 3D

- 1 Colores
- 2 Número de estados
- 3 Agregar soporte para *CUDA*
- 4 Uso de *shaders GLSL ó HLSL*
- 5 Generación de multivariada de construcciones por estado
- 6 Texturizado de vialidades
- 7 Objetos móviles: autos, camiones, aviones, etc.
- 8 Renderizado de los habitantes

## Limitaciones del visor 2D y 3D

- 1 Colores
- 2 Número de estados
- 3 Agregar soporte para *CUDA*
- 4 Uso de *shaders GLSL ó HLSL*
- 5 Generación de multivariada de construcciones por estado
- 6 Texturizado de vialidades
- 7 Objetos móviles: autos, camiones, aviones, etc.
- 8 Renderizado de los habitantes

## Limitaciones del visor 2D y 3D

- 1 Colores
- 2 Número de estados
- 3 Agregar soporte para *CUDA*
- 4 Uso de *shaders GLSL ó HLSL*
- 5 Generación de multivariada de construcciones por estado
- 6 Texturizado de vialidades
- 7 Objetos móviles: autos, camiones, aviones, etc.
- 8 Renderizado de los habitantes

## Limitaciones del visor 2D y 3D

- 1 Colores
- 2 Número de estados
- 3 Agregar soporte para *CUDA*
- 4 Uso de *shaders GLSL ó HLSL*
- 5 Generación de multivariada de construcciones por estado
- 6 Texturizado de vialidades
- 7 Objetos móviles: autos, camiones, aviones, etc.
- 8 Renderizado de los habitantes

# Conclusiones

## Conclusiones y perspectivas

- *Artífice* proporciona una forma alternativa de estudiar el crecimiento de ciudades.
- La generación se encuentra basada en autómatas celulares bidimensionales cuya implementación no es muy compleja.
- La función de transición aplica reglas simples que producen patrones complejos.
- *Artífice* brinda la posibilidad de ver la evolución de una ciudad en términos de la ocupación de su terreno y su número de habitantes.

## Conclusiones y perspectivas

- *Artífice* proporciona una forma alternativa de estudiar el crecimiento de ciudades.
- La generación se encuentra basada en autómatas celulares bidimensionales cuya implementación no es muy compleja.
- La función de transición aplica reglas simples que producen patrones complejos.
- *Artífice* brinda la posibilidad de ver la evolución de una ciudad en términos de la ocupación de su terreno y su número de habitantes.

## Conclusiones y perspectivas

- *Artífice* proporciona una forma alternativa de estudiar el crecimiento de ciudades.
- La generación se encuentra basada en autómatas celulares bidimensionales cuya implementación no es muy compleja.
- La función de transición aplica reglas simples que producen patrones complejos.
- *Artífice* brinda la posibilidad de ver la evolución de una ciudad en términos de la ocupación de su terreno y su número de habitantes.

## Conclusiones y perspectivas

- *Artífice* proporciona una forma alternativa de estudiar el crecimiento de ciudades.
- La generación se encuentra basada en autómatas celulares bidimensionales cuya implementación no es muy compleja.
- La función de transición aplica reglas simples que producen patrones complejos.
- *Artífice* brinda la posibilidad de ver la evolución de una ciudad en términos de la ocupación de su terreno y su número de habitantes.

## Conclusiones y perspectivas

- Se ha implementado un visor 3D sobre C++, utilizando OpenGL. El visor ha sido optimizado con manejo de nivel de detalle y uso de *display lists*.
- La metodología de desarrollo se basa en Ingeniería *Orientada a Objetos* lo que asegura la reutilización de componentes.
- Faltan aspectos por mejorar, pero con tiempo y esfuerzo *Artífice* será una buena herramienta para los propósitos que fue construido.

## Conclusiones y perspectivas

- Se ha implementado un visor 3D sobre C++, utilizando OpenGL. El visor ha sido optimizado con manejo de nivel de detalle y uso de *display lists*.
- La metodología de desarrollo se basa en Ingeniería *Orientada a Objetos* lo que asegura la reutilización de componentes.
- Faltan aspectos por mejorar, pero con tiempo y esfuerzo *Artífice* será una buena herramienta para los propósitos que fue construido.

## Conclusiones y perspectivas

- Se ha implementado un visor 3D sobre C++, utilizando OpenGL. El visor ha sido optimizado con manejo de nivel de detalle y uso de *display lists*.
- La metodología de desarrollo se basa en Ingeniería *Orientada a Objetos* lo que asegura la reutilización de componentes.
- Faltan aspectos por mejorar, pero con tiempo y esfuerzo *Artífice* será una buena herramienta para los propósitos que fue construido.

## Referencias I

- 1 **Peter M. Allen**, *Cities and Regions as Self-Organizing Systems Models of Complexity*. Taylor & Francis e-Library. 1997.
- 2 **Michael Batty**, *Fractal Cities. A Geometry of Form and Function*. Academic Press. Harcourt Brace & Company, Publishers. 1994.
- 3 **Andrew Ilachinski**, *Cellular Automata, A Discrete Universe*. World Scientific Publishing. 2001.
- 4 **Luke Benstead, Dave Astle And Kevin Hawkins**, *Beginning OpenGL. Game Programming-Second Edition*. Course Technology. CENGAGE Learning. 2009.



## Referencias II

- 5 **Roger S. Pressman**, *Ingeniería del Software. Un enfoque práctico*. Mc Graw Hill. 2001.
- 6 **Carlos Gershenson**, *Enfrentando a la Complejidad: Predecir vs. Adaptar*. Cornell University Library. 1991.
- 7 **P. V. Coveney**, *Self-Organization and Complexity: A New Age for Theory, Computation and Experiment*. Proceedings of the Nobel Symposium on Self-Organization. 2003.
- 8 **George Kelly & Hugh McCabe**, *A survey of Procedural Techniques for City Generation*. ITB Journal. 2003.
- 9 **Yoav Parish & Pascal Müller**, *Procedural Modeling of Cities*. ACM SIGGRAPH, Los Angeles, CA, USA. 2001.

## Referencias III

- 10 **Michael Batty & Paul A. Longley**, *Urban shapes as fractals*. Area, volumen 19.3, 1987.
- 11 **Stefan Greuter, Jeremy Parker, Nigel Stewart & Geoff Leach**, *Real-time Procedural Generation of 'Pseudo Infinite' Cities*. RMIT University, Melbourne, Victoria, Australia.
- 12 **Carlos Gershenson and Francis Heylighen**, *How can we think the complex?*. Richardson, Kurt (ed.) *Managing the Complex: Philosophy, Theory and Application*. 2004.
- 13 **Paul Cilliers**, *Complexity & Postmodernism Understanding Complex Systems*. Taylor & Francis e-Library. 2002.
- 14 **Kate Ascher**, *Anatomy of a City*. The Penguin Press. 2005.



Agenda  
Presentación  
Marco teórico  
Planteamiento del problema  
Estado del Arte  
Análisis del problema  
Propuesta de solución  
Implementación  
Resultados obtenidos  
Conclusiones

Conclusiones y perspectivas  
Referencias

¿Mejor ya el Age no?

¡Gracias!

