جغرافیا و روابط انسانی، پاییز ۱٤۰۲، دوره٦، شماره۲، صص۲۰۷-۱۷۰

تحلیلی بر مدل‌سازی مبتنی بر قانون و قواعد؛ نرم‌افزار سیتی اینجین

علی اسکوئی‌ارس[1*]، فریدون بابائی اقدم[2]، ایرج تیموری

۱-دانشجوی دکتری جغرافیا و برنامه‌ریزی شهری، گروه جغرافیا و برنامه‌ریزی شهری، دانشکده برنامه‌ریزی و علوم محیطی، دانشگاه تبریز، تبریز، ایران (ali.oskouee.aras@tabrizu.ac.ir)

۲-دانشیار جغرافیا و برنامه‌ریزی شهری، دانشگاه تبریز، دانشکدهٔ برنامه‌ریزی و علوم محیطی، ، گروه جغرافیا و برنامه‌ریزی شهری، تبریز، ایران

۳-دانشیار جغرافیا و برنامه‌ریزی شهری، دانشگاه تبریز، دانشکدهٔ برنامه‌ریزی و علوم محیطی، ، گروه جغرافیا و برنامه‌ریزی شهری، تبریز، ایران

تاریخ دریافت:۱٤۰۱/۱۲/۰۲ تاریخ پذیرش: ۱٤۰۱/۱۲/۲۲

چکیده

سیتی اینجین، یک برنامه نرم‌افزاری در جهت مدل‌سازی شهری بر اساس قوانین مشخص می‌باشد. این برنامه به عنوان یک کانال ارتباطی انعطاف‌پذیر به جهت تبدیل داده‌های دو بعدی به مدل‌های شهری سه بعدی ارائه می‌شود. کاربردهای ویژهٔ این برنامهٔ نرم‌افزاری شامل پردازش داده‌های سیستم اطلاعات جغرافیایی، نقشه‌برداری و همچنین سیستم اطلاعات شهریِ دوبعدی (GIS) با سه هدف مشخص می‌باشد، این اهداف عبارتند از؛ ایجاد یک مدل شهری سه‌بعدی، ایجاد تصویری دقیق از روند شهرک سازی پیشنهاد شده، یا حتی کاوش در فضای طراحی یک پروژه بالقوه است. بخش اصلی مدل‌سازی با نرم‌افزار Esri's CityEngine بر اساس دستور و قوانین مشخص مزایای منحصر به فردی دارد؛ امکان ساخت شهرهای بزرگ به آسانی ساخت و ساز شهرهای کوچک با همان کیفیت حتی در مقیاس بزرگ‌تر، یکی از مزیت‌های بی‌نظیر این رویکرد است. علاوه بر این، رویکرد مبتنی بر قانون به این معنی است که فضاهای طراحی بزرگ را می‌توان به سرعت، به صورت تعاملی و تحلیلی بررسی و مقایسه کرد. چنین مزایایی باید با دقت در برابر گذر بیش از حد زمان برای ایجاد و پارامترسازی قوانین و گاهاً ایجاد مدل‌های سَبکی یا مدل‌های تقریبی ایجاد شده متعادل شوند. (مدل‌های تقریبی با هدف آسانی انجام محاسبات، و مدل‌های سبکی با هدف تفسیر عوامل به کار برده شده مورد استفاده قرار می‌گیرند.) کانال ارتباطی سیتی اینجین که از جریان‌های کاری ستی‌تر ناشی می‌شود، ممکن است در ابتدا بسیار اثرگذار باشد. ما جریان‌های کاری اصلی و تغییر پذیری این رویکرد، ترسیم زبان برنامه‌نویسی رویه‌ای مورد استفاده را معرفی می‌کنیم و مسیرهای تغییر فرمت فایل‌های موجود را مورد بحث و بررسی قرار می‌دهیم.

واژگان کلیدی: سیتی اینجین، سه بعدی، مدل‌سازی شهری، سیستم اطلاعات جغرافیایی.

شاپا الکترونیکی:3851-645 doi 10.22034/GAHR.2023.386685.1820

مقدمه

فناوری‌های سه بعدی نحوه‌ای که ما برنامه‌ریزی انجام می‌دهیم، موضوعات را درک می‌کنیم، ارتباط برقرار می‌کنیم و محیط‌های شهری خود را مستند می‌کنیم کاملا تغییر می‌دهد. با این حال، تغییرات مذکور به ندرت به آسانی انجام می‌گیرند، چراکه مشکلات و چالش‌های متعددی پیرامون ایجاد تغییر از مدل‌سازی دو بعدی به سه بعدی حاصل از زنجیرهٔ ابزار(مجموعه‌ای از نرم افزارهای برنامه‌نویسی) وجود دارد. خواندن طرح‌ها و نقشه‌های دوبعدی اغلب چالش برانگیز است زیرا آنها به اندازهٔ یک بعد از دنیای سه بعدی ما فاصله دارند. داده‌های سه‌بعدی باید با استفاده از ترفندها و قراردادهای مختلف، مانند خطوط کانتور(به نقاطی با ارتفاع برابر و بالاتر از یک سطح معین)، نمودارهای ارتفاع، نمادها و سایه‌گذاری، رمزگذاری شوند. این موضوع به این دلیل است که طراحی‌های سه بعدی شامل اطلاعات بیشتری نسبت به طراحی‌های دو بعدی است.

جمع‌آوری و به اشتراک‌گذاری این اطلاعات سه بعدی، تا همین اواخر، دشوار و ممنوع بوده است. از آنجایی که فناوری‌های مختلف مانند CAD سه بعدی کالا و بازسازی فتوگرامتری توسعه یافته‌اند، ما می‌توانیم مدل‌های سه بعدی و مجازی دنیای سه‌بعدی خود را با دقت بسازیم. همزمان با افزایش دقت در داده‌ها، مدل‌های سه بعدی، منجر به دسترسی راحت‌تر به داده‌ها می‌شود. در حالی که همیشه امکان ایجاد مدل‌های مقیاس فیزیکی از محیط‌ها وجود داشته است، اما از معایب این مدل می‌توان به گران و هزینه‌بر بودن، دشواری در حمل و نقل یا اشتراک‌گذاری آن‌ها و در نهایت ذخیره‌سازی حجیم است. فناوری‌هایی همه‌گیر همانند واقعیت مجازی و واقعیت افزوده (VR، AR، که اغلب به صورت XR خلاصه می‌شود) به همه افراد، از کودکان گرفته تا برنامه‌ریزان شهری اجازه می‌دهد تا طرح‌های پیچیده را با کاوش در مقیاس‌های دنیای واقعی درک کنند. ابزارهای سه‌بعدی مانند شبیه‌سازی فیزیکی (پتانسیل خورشیدی، مدل‌سازی پنجره) و نظریه‌ها و دیدگاه‌های مختلف به مهندسان کمک می‌کنند تا با سرعت بخشیدن در یافتن فضاهای طراحی و درک آن، تولید طرح‌های بهتر و با درک هرچه بهتر هر موضوعی محیط‌های تجربی بهتری طراحی کنند. با این حال، مدل‌سازی سه بعدی دشوار است. نمایش سه بعدی عملا یک شبکه است. این شبکه مجموعه‌ای از رئوس در فضای سه بعدی می‌باشند که ما سعی بر ایجاد مثلث بین آنها را داریم. و با ایجاد مثلث‌های بسیار، می‌توانیم محیط‌های پیچیده سه بعدی را نمایش دهیم. ابزارهای زیادی برای ایجاد این شبکه‌های چند ضلعی وجود دارد. ابزارهای مدل‌سازی سه بعدی سنتی و دَستی راهی را برای ایجاد همزمان مثلث‌های متعدد همراه با عناصر اولیه پیچیده‌تر (همچون کره‌ها، مکعب‌ها، منحنی‌ها، سطوح، اکستروژن‌ها و غیره) ارائه می‌دهند.

چنین ابزارهای دستی عبارتند از (Autodesk Maya (2019، Trimble SketchUp (2019)، یا Blender (2019). با وجود اینکه این ابزارهای دستی بی‌نهایت پیچیده و عمومی شده‌اند، همچنان درخواست تمرکز بیشتر و بهتر در ارتباط با موقعیت‌یابی، ویرایش مثلث‌ها و عناصر و موارد اولیه از کاربران مطرح می‌شود. در منطقهٔ شهری که ما مدل می‌کنیم، ممکن است فردی را به عنوان یک متخصص با صبر و حوصله بسیار، برای طراحی و قرار دادن یک دستگیره در کروی شکل بر روی هر در ورودی مستطیل مانند، برای هر ساختمان، به کار گرفته شود. کاری که ما ترجیح می‌دهیم انجام شود این است که با ایجاد قانونی عبارت "یک کره را به هر در ورودی وصل کنید"، را کدگذاری کند. خوشبختانه، کامپیوترها در انجام این نوع کارهای تکراری عملکرد نسبتاً خوبی را از خود نشان می‌دهند – به شرطی که بتوان راهی پیدا کنیم که به آنها توضیح دهیم که دقیقا چه کاری باید انجام دهند. مدل‌سازی مبتنی بر قانون و قواعد راهی است که به کمک آن می‌توانیم آموزش به کامپیوترها را عملی کنیم. و به طور ویژه درگیر با یک سیستم مدل‌سازی به نام Esri's CityEngine خواهیم شد. چنین سیستم‌های مدل‌سازی ابزارهایی را برای تولید شبکه‌های سه‌بعدیِ رویه‌ای بر اساس سیستم‌های مبتنی بر قوانین ارائه می‌دهند – آنها قادرند مدل‌هایی با میلیون‌ها رئوس در ثانیه ایجاد کنند. اینجاست که ما شاهد مزیت دیگری از کار با مدل‌های سه بعدی مجازی به جای فیزیکی هستیم. برنامه‌های رایانه‌ای می‌توانند با پیروی از قاعده و قانون مدل‌های شبکه‌ای چند ضلعی و مجازی را به گونه‌ای ابرانسانی با سرعت و دقت هرچه بیشتر ایجاد کنند. می‌توانیم مکرراً قوانین را تغییر دهیم و محیط‌های حاصل را روی صفحه نمایش، در واقعیت مجازی مشاهده و کاوش کنیم یا با استفاده از چاپگر سه‌بعدی آنها را به صورت فیزیکی تولید کنیم. انجام همان تغییرات در یک مدل فیزیکی سه بعدی نیازمند زمان بسیاری است.

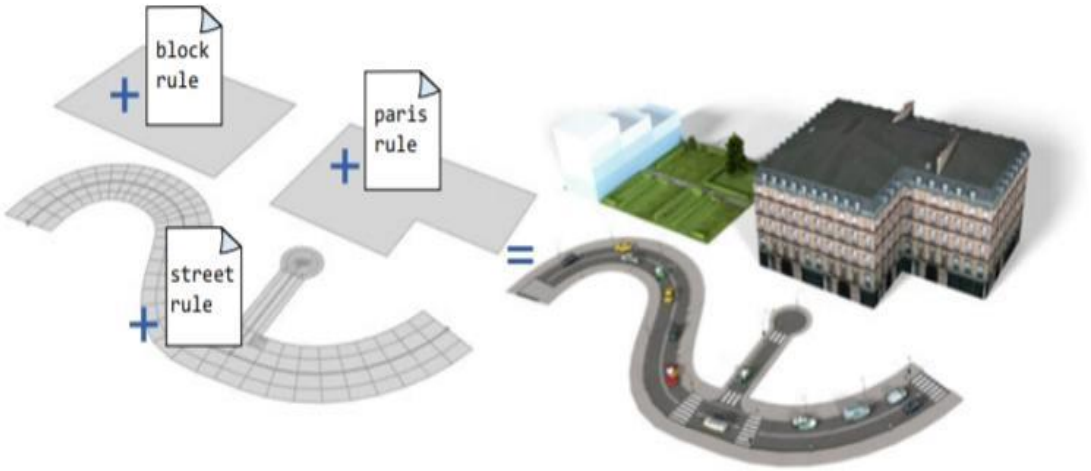
اشکال دو بعدی + قوانین = مدل‌های سه بعدی

به دلیل ماهیت سلسله مراتبی، سیستماتیک و اغلب تکراری محیط‌های شهری، مدل‌سازی شهری مبتنی بر قانون به طور کلی نیروی محرکه‌ای برای مدل‌سازی رویه‌ای عمومی بوده است. به گونه‌ای خلاصه‌وار، ما متذکر می‌شویم که سایر سیستم‌های مبتنی بر قانون در سایر حوزه‌ها بسیار موفق عمل می‌کنند. مهم‌ترین این حوزه‌ها سیستم‌های تجاری هستند، همانند SpeedTree (2019) با هدف تولید سریع درختان، جنگل‌ها و نرم‌افزار Grome (ویکی‌پدیا، ۲۰۱۹) برای ایجاد زمین‌ها و مناظر هستند. برای هر حوزه متفاوت، تکنیک‌ها و قوانین متفاوتی مناسبی اختصاص داده شده است. همانطور که خواهیم دید، قوانین و عملیاتی مورد استفاده در نرم‌افزار سیتی اینجین، به دقت تنظیم

شده‌اند تا امکان مدل‌سازی سریع و دقیق ساختمان‌ها و خیابان‌ها را فراهم کنند. قبل از تصمیم‌گیری جهت استفاده از کانال ارتباطی مدل‌سازی مبتنی بر قانون، باید مزایا و معایب آن را با کانال ارتباطی مدل‌سازی سنتی سنجید. برای مدل‌های کوچک‌تر یا پیچیده‌تر، مدل‌سازی دستی ممکن است سریع‌تر و ارزان‌تر انجام شود، و زمان لازم برای طرح قوانین ممکن است بیشتر از زمانی باشد که برای انجام مدل‌سازی دستی صرف می‌شود. مدل‌سازی مبتنی بر قانون به‌ویژه برای هندسه‌های پیچیده دشوار است چرا که شامل تصمیم‌گیری‌های بسیاری جهت مکان‌یابی و ارزیابی در فضای این نوع مدل‌سازی است. تبدیل هر تصمیم به یک قاعده و قانون و اطمینان از اینکه این تصمیمات و قوانین متقابلاً در هر شرایطی به طور مناسب تاثیرگذار باشند، می‌تواند پروسه‌ای زمان‌بر باشد. ما متذکر می‌شویم که بسیاری از مسائل توجیهی در این فصل با استفاده از ابزارهای مدل‌سازی دستی با سرعت بیشتری ایجاد می‌شوند – و تنها زمانی که با تعریف مقیاس‌های بیشتری نسبت به تعداد مشخص و ثابت از مقیاس‌ها در مناطق بزرگ‌تر، مدل‌سازی مبتنی بر قانون به مدت زمان صرف شده برای ایجاد قوانین امتیاز مثبتی را به عنوان پاسخ مناسب ارائه می‌دهد.

توانایی برای نوشتن یک فایلی از این قواعد و قوانین مهارت جدیدی است که باید مانند هر مهارت دیگری آموزش داده شود، مطالعه شود و به خوبی از آن نگهداری شود. آنجایی که این نوع از مدل‌سازی یک نوع جدیدتری از تکنولوژی است، یافتن کارکنان واجد شرایط، خصوصاً به دلیل نیازمندی به طراحی شهری، دانش اولیه و ابتدایی جبر خطی و همچنین توانایی (en) کدگذاری قوانین مورد نیاز ما در یک زبان برنامه نویسی، دشوار است. گذشته از این دشواری‌ها، مدل‌سازی مبتنی بر قوانین می‌تواند یک زنجیره ابزار قابل تغییر، سریع و حساس برای توسعهٔ طرح‌های شهری از مدل‌سازی تنها یک ساختمان، طراحی‌ها در مقیاس دانشگاه‌ها و فضای باز، تا شبیه‌سازی‌ها در مقیاس همسایگی و شهری ارائه دهد. هنگامی که قوانین ما سودمند و در دسترس باشند، می‌توان هندسه مورد نیاز را به تعداد بالا را به راحتی و با سرعت بیشتر ایجاد کرد. تغییرات و اصلاحات در سناریوها را می‌توان بی وقفه انجام داد. هم مدل‌سازی اطلاعات ساختمان («آیا روی ساختمان‌ها دودکش و پشت بام در نظر می‌گیریم؟»)، قالب ارائه (Webviewer، VR)، و هم ویژگی‌های قانون («این ساختمان چقدر ارتفاع دارد؟») می‌تواند به لطف مدل‌سازی مبتنی بر قانون، یک‌باره در کل شهر به‌روزرسانی شود. Esri’s CityEngine یک سیستم نرم‌افزاری برای مدل‌سازی مبتنی بر قوانین در حوزهٔ شهری است. که یک محیط بصری برای اعمال قوانین، ایجاد قوانین جدید و بررسی نتایج فراهم می‌کند. همانطور که بواسطهٔ ArcGIS Pro مشاهده کردیم، این انتقال یک کانال ارتباطی

بسیار قدرتمند با پشتیبانی از تمام فرمت‌های اصلی صنعت ایجاد کرده است. این زمینهٔ تجاری تحت بستر مورد نیاز برای ایجاد ابزارهای تحلیلی جریان کاری سیتی اینجین- اشکال دوبعدی به سیستم وارد می‌شوند، جایی که از قوانین برای تبدیل آنها به مدل‌های سه بعدی استفاده می‌شوند. این مدل‌ها، خروجی‌های مدل سه بعدی هستند که می‌توانیم آن‌ها را در فضای سیتی اینجین مشاهده کنیم یا به وب یا VR انتقال دهیم.

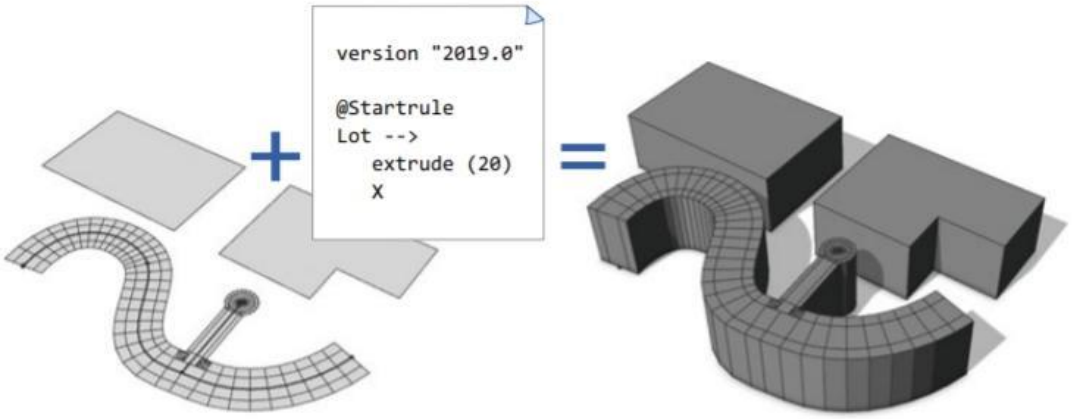


شکل ۱. الگو مرکزی سیتی اینجین اعمال قوانین بر روی اشکال (خاکستری، چپ) برای ایجاد مدل‌های سه بعدی (راست) است. این رویکرد قادر به ایجاد انواع بسیاری از مدل‌های قانون محور است.

بنابراین، فرآیند مرکزی برای مدل‌سازی در سیتی اینجین، اعمال قوانین بر روی اشکال با هدف ایجاد مدل‌ها است (شکل ۱). یک قانون CGA یک فایل نوشتاری شامل لیستی از دستورالعمل‌ها است. در شکل ۲، یک قانون ساده را معرفی می‌کنیم که یک شکل را در مدلی از یک منشور سه بعدی خارج می‌کند. در حالی که این قانون فقط شامل پنج خط دستورالعمل است، فایل‌های قوانین پیچیده می‌توانند هزاران دستورالعمل داشته باشند. هدف این فصل این است که یک تور مقدماتی گسترده از این نوع سیستم مدل‌سازی با بررسی عمیق در موضوعات مختلف اجرا کند. به توصیف اشکال، قوانین، ابزارهای تجزیه و تحلیل و مسیرهای انتقال فایل‌ها و مستندات از سیتی اینجین ادامه

می‌دهیم. پس از خواندن این فصل، افراد تحت سبک یادگیری جنبشی ترغیب می‌شوند که چند روزی را از طریق آموزش‌های سیتی اینجین ارائه شده توسط Esri (2019a) صرف کند. به طور مشابه، اسناد آنلاین Esri منبع ارزشمندی از جزئیات فنی است (Esri 2019b).

منشأ ساخت اشکال در مدل‌سازی



شکل ۲. یک فایل قانون CGA ساده (در مرکز) به چندین شکل مختلف (چپ) برای ایجاد مدل‌های سه بعدی مرتبط (راست) اعمال می‌شود. این قانون، منشوری به ارتفاع ۲۰ متر روی شکل ایجاد می‌کند.

سیتی اینجین دو جریان کاری را برای ساخت فوری تمامی شهرها با کاربران به تعداد کمتر فراهم می‌کند. City Wizard (File → New ...→ CityWizard→CityEngine) از یک گردش کار کاملاً رویه‌ای برای ایجاد تعداد قابل توجهی از اشکال با قوانین پیچیده فقط با چند کلیک استفاده می‌کند. البته، City Wizard یک شهر کاملاً تخیلی است؛ اگر بجای آن بخواهیم از مجموعه‌ای از اشکال داده محور استفاده کنیم، می‌توانیم از کلید import Map استفاده کنیم (فایل دریافت اطلاعت نقشه ...). این ابزار تصاویر ماهواره‌ای، نقشه‌های ارتفاع، ردپاها و شبکه‌های خیابانی را برای ساخت و ایجاد اشکال و زمین‌ها برای یک منطقه واقعی بازگذاری می‌کند (شکل ۳). با این حال، از آنجا که هیچ منبع داده مشترکی برای قوانین ساختمان وجود ندارد، فقط قوانین ساده ارائه شده است. City Wizard و Map Import هر دو از اشکال برای مدل‌سازی سریع کل شهرها استفاده می‌کنند، اما کنترل محدودی روی اشکال و قوانین به ما می‌دهند. ما به بررسی روش‌های کنترل شده تری برای ایجاد و ساخت اشکال ادامه می‌دهیم. اشکال معمولاً چند ضلعی‌های دو بعدی هستند که روی زمین قرار دارند. بسیاری از

کاربردها و پیچیدگی‌های سیتی اینجین ناشی از روش‌های مختلف ایجاد اشکال است. منابع داده‌ای مختلف برای اشکال، یک نمای کلی از جریان‌های کاریِ مدل سازی‌های مختلف موجود در سیتی اینجین ارائه می‌دهند:

- برای ایجاد یک مدل سه‌بعدی از یک منطقه موجود، ممکن است از مجموعه‌ای از قطعات ساختمانی(FileGD، DFX، Shapefile، یا OBJ) حاصل از منبع داده‌های مکانی به عنوان شکل استفاده کنیم.
- برای برنامه‌ریزی در جهت ایجاد یک منطقهٔ شهری جدید، ممکن است اشکال خودمان را ترسیم کنیم، به عنوان مثال افزودن گوشه به هر قطعه زمین در یک زمان مشخص. ساده‌ترین راه برای ترسیم یک شکل، استفاده از ابزاری تحت عنوان ابزار ترسیم شکل مستطیل می‌باشد، که این روش امکان ترسیم و قرارگیری مستطیل روی سطح زمین با کلیک کردن و کشیدن دو گوشه آن را فراهم می‌کند. برای افزایش دقت، ممکن است طرح کلی این اشکال را از تصاویر وارد شده به سیتی اینجین ردیابی کنیم.

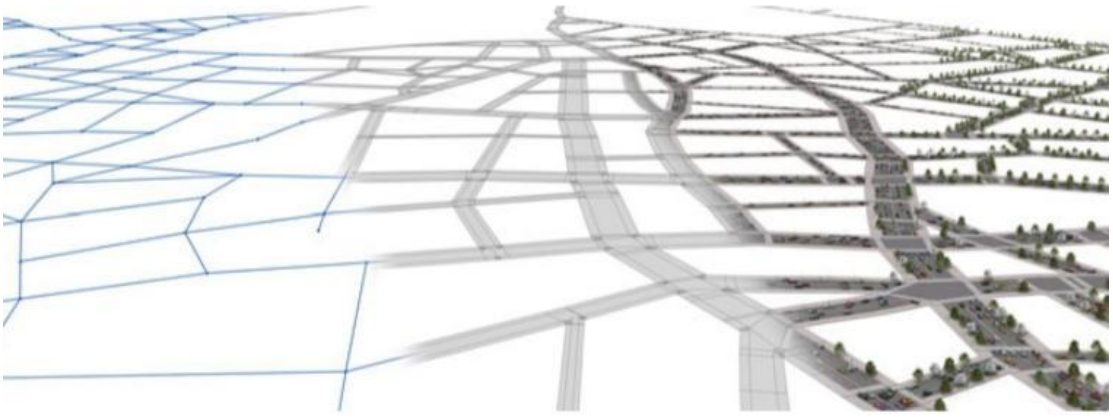
شکل ۳. شهری که در ۳۰ ثانیه با استفاده از عملکرد Map Import ایجاد شد.

- اگر می‌خواهیم از قوانینی برای افزودن پنجره‌ها به نمای خالی ساختمان استفاده کنیم، می‌توانیم ساختمان را با استفاده از ابزارهای مدل‌سازی سه بعدی دستی ارائه‌شده توسط سیتی اینجین ترسیم کنیم. این یک گردش کار غیر معمول است زیرا ممکن است اشکال افقی نباشند. چنین گردش کاری به ما اجازه می‌دهد

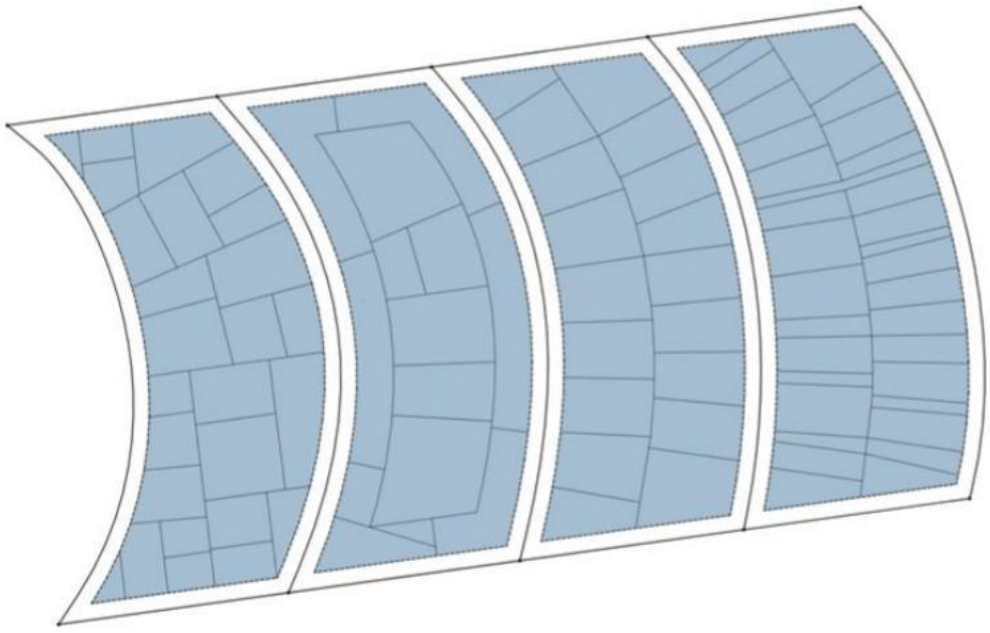
تا به صورت دستی یک ساختمان را مدل‌سازی کنیم و سپس قوانین را فقط برای نماهای خاص اعمال کنیم. سیتی اینجین دارای طیف وسیعی از ابزارها برای مدل‌سازی شکل دستی است که شامل تولید مستطیل، چند ضلعی و دایره می‌شود. هنگام مدل‌سازی یک شبکهٔ خیابانی، ممکن است یک نمودار خیابانی را وارد این نوع مدل کنیم (فرمت‌های پشتیبانی شده شامل DXF، FileGDB و OpenStreetMap) و از سیستم سیتی اینجین برای تولید اتوماتیک اشکال خیابان‌ها، بلوک‌ها و اشکال قطعه زمین‌ها بین خیابان‌ها استفاده کنیم. ما به بررسی سیستم شکل پویا به صورت عمیق‌تری ادامه می‌دهیم.

اشکال پویا: خیابان‌ها، بلوک‌ها و زمین‌ها

اشکال پویا از الگوریتم‌هایی برای تقریب فرم‌هایی که در محیط‌های شهری خود می‌بینیم استفاده می‌کنند. به همین دلیل، آنها فقط طرح‌های شبیه‌سازی شده‌ای هستند که با ویژگی‌های کلی (محدودهٔ عرض قطعهٔ ساختمان) مطابقت دارند، اما با اندازه‌گیری‌های خاص (عرض یک قطعهٔ خاص) مطابقت ندارند. ما آنها را به عنوان اشکال پویا توصیف می‌کنیم زیرا به صورت دینامیک از نمودار خیابانی تولید می‌شوند. انعطاف‌پذیری سیتی اینجین اجازه می‌دهد تا ترکیبی از این رویکردهای تولید شکل —مدل‌سازی دستی، داده محور و پویا – با هم استفاده شوند. برای مثال، خیابان‌ها را می‌توان از منبع داده‌های GIS وارد کرد و بلوک‌های بین خیابان‌ها را می‌توان به‌صورت پویا به بخش‌هایی تقسیم کرد، یا ناحیه‌ای از شهر که داده‌های GIS برای خیابان‌ها و زمین‌ها وجود دارد، می‌تواند با خیابان‌ها و زمین‌های مجاور که به‌صورت دینامیکی تولید می‌شوند، افزایش یابد. یک نمودار خیابانی، خیابان‌ها را در یک شبکهٔ خیابانی توصیف می‌کند. همانطور که در شکل ٤ نشان داده شده است، بر روی این نمودار، اشکال خیابان پویا برای پیاده‌روها، تقاطع‌ها و خود خیابان ایجاد می‌شود. لبه‌های نمودار خطوط مرکزی را توصیف می‌کنند و گره‌ها (جایی که لبه‌ها به هم می‌رسند) اتصالات خیابان را توصیف می‌کنند.



شکل ٤: قسمت چپ: نمودار خط مرکزی خیابان آبی: قسمت وسطی: اشکال خیابانی ایجاد شده: قسمت راستی: مدل‌های سه بعدی تولید شده با اعمال قوانین بر روی اشکال



شکل ۵. الگوریتم‌های تقسیم فرعی بلوک که برای ایجاد قطعات ساختمانی استفاده می‌شوند. از چپ به راست: بازگشتی، انحراف از موقعیت و اسکلت. سمت راست: اسکلت اصلاح‌شده برای بی‌نظمی زیاد و عرض قطعهٔ باریک‌تر.

بین خیابان‌ها، سیتی اینجین به صورت پویا بلوک‌ها و از بلوک‌ها، قطعه زمین‌ها را تولید می‌کند. به طور کلی، هر حلقه از خیابان‌ها یک بلوک در داخل خود ایجاد می‌کند. بلوک شامل مجموعهٔ دیگری از ویژگی‌ها است که تقسیم‌بندی آن را به اشکالی از قطعه زمین‌ها تعریف می‌کند. شکل زمین نمایانگر قطعه‌ای جزئی از کل زمین است که در آن از قوانینی برای تولید مدل‌های ساختمانی مجزا استفاده می‌کنیم. الگوریتم‌های تقسیم بلوک به زمین توسط Vanegas و همکاران مورد بحث قرار گرفته‌اند، (۲۰۱۲) و به دو بخش اصلی زیربخش بازگشتی و انحراف از موقعیت تقسیم می‌شدند. هر یک از اینها را می‌توان با ویژگی‌های کنترلی مؤثر بر مساحت، عرض و تغییرات کنترل می‌کنند، مانند شکل ۵ کنترل کرد. توالی تولید بخش مهمی از الگوی مدل‌سازی است که توسط سیتی اینجین برای اشکال پویا استفاده می‌شود: خیابان‌ها ساخته می‌شوند، بین آن‌ها بلوک‌ها پیدا می‌شوند و در نهایت در داخل هر بلوک، تعداد قطعه زمین ایجاد می‌شود. توجه به این نظم و ترتیب هنگام ایجاد مناظر شهری و شروع به ایجاد خیابان قبل از اقدام به تولید بلوک و زمین بسیار مهم است. این موضوع به این دلیل است که تغییرات کوچک در

شبکه خیابان بر بسیاری از بلوک‌ها تأثیر می‌گذارد، در حالی که تغییر تنظیمات بخش فرعی بلوک تنها بر روی قسمت‌های بلوک تأثیر می‌گذارد.

به طور مشابه، تغییر قاعده یا ویژگی‌های یک زمین، تنها بر مدل (ساختمان) آن تأثیر می‌گذارد. به خاطر داشته باشید که اشکال ما نقطهٔ شروع قوانین خواهند بود، همچنین ذکر نام قوانین شروع، به صورت پیش‌فرض برای هر نوع شکل پویا بسیار مهم است. این نام برای تخصیص خودکار یک قانون شروع (اولیه) برای شکل قلمداد می‌شود. برای مثال، با کشیدن یک فایل قانون روی شکل پیاده‌روی یک خیابان، سعی می‌شود از قاعده‌ای به نام Sidewalk استفاده شود (و هیچ پارامتری در نظر گرفته نمی‌شود)، در حالی که همان فایلی که روی شکل زمین کشیده می‌شود، از قانون قطعه زمین‌ها استفاده می‌کند.

نمودارها و شهرها

خواننده هوشیار متوجه خواهد شد که نمودارهای خیابان (خود خطوط مرکزی خیابان) پویا نیستند. نمودار خیابانی حاوی اطلاعات مورد نیاز برای ایجاد سایر اشکال پویا، به صورت دینامیکی است. همانطور که انتظار داشتیم، سیتی اینجین رویکردهای دستی–سنتی، مبتنی بر داده و رویه‌ای را برای ایجاد نمودارهای خیابانی ارائه می‌کند.

ایجاد نمودار خیابان به صورت دستی را می‌توان با ابزارهای چند ضلعی یا بدون ابزار تخصصی انجام داد. و به این شکل امکان ایجاد رئوس و لبه‌های نمودار با کلیک کردن در گوشه‌ها یا با ترسیم خیابان‌ها فراهم می‌شود. سپس از ابزار Edit Street (ابزار اصلاح خیابان، در جهت اعمال اصلاحات و تغییرات) می‌توان برای تغییر مکان رئوس، خمیدگی خیابان‌ها و تنظیم عرض خیابان یا پیاده‌رو استفاده کرد.

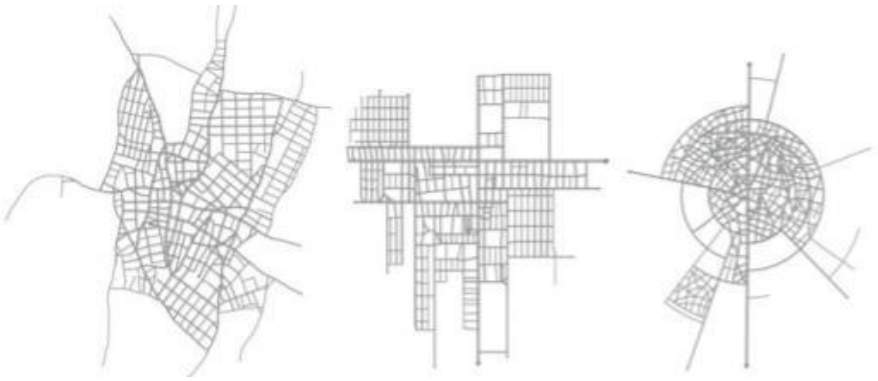
یک روش جایگزین برای رسم نمودارهای خیابانی به طور مستقیم، وارد کردن یک نمودار موجود از یک منبع GIS است. فرمت‌های پشتیبانی شده عبارتند از DXF، FileGDB و OpenStreetMap. سیتی اینجین می‌تواند ویژگی‌هایی مانند عرض خیابان را در برخی از این قالب‌ها تجزیه و نقشه‌برداری کند، که می‌تواند از تخصیص دستی با ابزار Edit *Street* جلوگیری کند. کار با منابع داده‌ای مختلف می‌تواند تجربه‌ای را به همراه داشته باشد زیرا هر کدام دارای ویژگی‌های متفاوتی مانند فاصله بین گره‌ها یا وجود بخش‌هایی با نمودارهای منحنی شکل هستند. در جهت کمک به کار کردن با این نمودارها، ابزارهای مختلفی برای ساده کردن نمودار (نمودار ساده

کردن نمودار...)، تراز کردن نمودار با زمین (نمودار تراز کردن نمودار به زمین)، یا حل تلاقی لبه‌های نمودار در پل‌ها و زیرگذرها (گراف) در دسترس هستند.(گراف ساخت پل‌ها...)

برای ساخت شبکه‌های خیابانی بزرگ در جایی که منبع GIS در دسترس نیست، سیتی اینجین ابزار Grow Streets را ارائه می‌کند که مجموعه‌ای از خیابان‌ها و همچنین بلوک‌ها و قطعه زمین‌هایی را که در بالا توضیح داده شد را به خوبی ایجاد می‌کند. منشأ الگوریتم‌های Grow street مورد استفاده، در مقاله پریش و مولر (۲۰۰۱) توضیح داده شده است، اگرچه این الگوریتم‌ها اکنون تا حدودی از جزئیات منتشرشده فراتر رفته‌اند. به طور خلاصه، L-Systems به عنوان یک سیستم نرم‌افزاری که به خودی خود حساس است (Prusinkiewicz and Lindenmayer 2012) برای رشد و ایجاد خیابان‌های اصلی و فرعی استفاده می‌شوند. لبه‌های تازه رشد کرده برای اتصال به بخش‌هایی از شبکه‌های موجود می‌شکنند. با ترکیب الگوهای مختلف رشد برای هر دو خیابان اصلی و فرعی، طیف گسترده‌ای از شبکه‌های مختلف را می‌توان رشد داد، که در شکل ٦ نشان داده شده است.

ابزار Grow Streets همچنین اجازه می‌دهد تا نوع زیربخش بلوک پویا مشخص شود. هنگامی که نمودار خیابان واقعی وارد مدل‌سازی شد یا نمودار غیر واقعی و مصنوعی رشد کرد، از ابزار Edit Street و Street Creation می‌توان برای اصلاح یا تنظیم دقیق داده‌ها استفاده کرد. چندی از کاربردهای این نمودارها در جهت ایجاد مدل‌های خیابانی فراتر از استفاده معمولی آنهاست. قوانین مناسب را می‌توان برای ایجاد ساختارهای نمودار مانند مختلف از جمله دیوارها، راه آهن‌ها و خطوط برق مانند شکل ۷ استفاده کرد.

تا به اینجا ما یک مرور کلی از روش‌های متعددی را که از سیتی اینجین می‌توان برای ایجاد اشکال مختلف استفاده کرد، دیده‌ایم؛ اما ما همچنان به بررسی چگونگی دستیابی به قوانینی برای تبدیل اشکال خود به مدل‌های سه بعدی ادامه می‌دهیم.



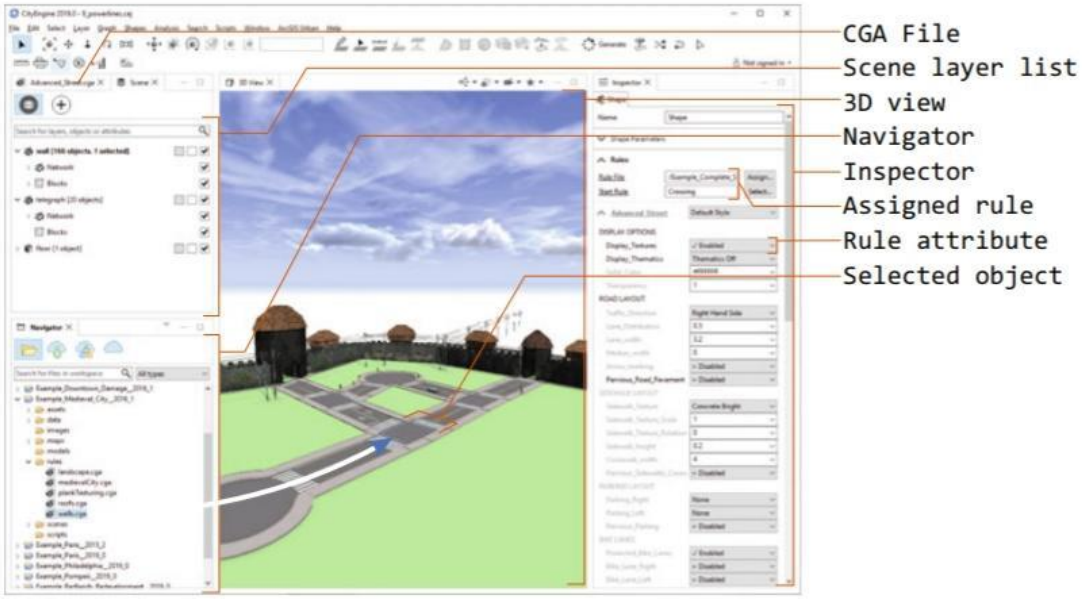
شکل ٦. طیف گسترده‌ای از الگوهای خیابانی را می‌توان با انتخاب الگوهای اصلی و فرعی خیابان تولید کرد. سمت چپ: ماژور ارگانیک و مینور شطرنجی. وسط: شطرنجی بزرگ و شطرنجی مینور؛ سمت راست: ماژور شعاعی و مینور ارگانیک

شکل ۷. دیوارها، خیابان‌ها، نرده‌ها و خطوط برق تولید شده از قوانین اعمال شده بر روی اشکال نمودار پویا.

نوشتن قوانین CGA برای سرگرمی و سود

قوانین سیتی اینجین در زبان برنامه‌نویسی Computer Generated Architecture (CGA) به معنای معماری حاصل از کامپیوتر نوشته شده است. نوشتن یک قانون ساده CGA می‌تواند سریع و بدون دردسر باشد؛ با این حال، نوشتن یک قانون واقع‌بینانه و یا قابل تغییر یک فرآیند پیچیده است. مجموعه‌ای از قوانین موجود ارائه شده است و قوانین بیشتر را می‌توان به صورت آنلاین یافت. سریعترین مسیر برای ایجاد یک صحنهٔ سه بعدی از نقشهٔ دو بعدی، ترکیب و پارامترسازی قوانین موجود است، بدون اینکه خودمان کد CGA بنویسیم. قوانین از پیش نصب شده را می‌توان در پروژه ESRI.lib یافت. مجموعهٔ دیگری از قوانین که برای شرایط مختلف به خوبی تنظیم و نوشته شده را نیز می‌توان به صورت آنلاین آن‌ها را یافت و تحت آموزش آنلاین قرار گرفت.

(راهنما ← دانلود آموزش‌ها و مثال‌ها).



شکل۸. عناصر رابط کاربری سیتی اینجین. نارنجی: عناصر مهم رابط. آبی: کشیدن یک قانون روی شکل انتخاب شده برای تولید یک مدل سه بعدی

در نهایت، بسیاری از بسته‌های قوانین تولید شده توسط کاربر (فایل‌های RPK منفرد حاوی قوانین و منابع) با کیفیت‌های متفاوت را می‌توان به صورت آنلاین یافت («جستجوی محتوای ArcGIS» با کلمهٔ کلیدی CityEngine؛ Esri 2019c). بررسی قوانین موجود روشی قدرتمند برای درک اینکه چگونه می‌توان مدل‌ها را با استفاده از زبان CGA تولید کرد. از آنجایی که نوشتن قوانین ممکن است زمان زیادی را صرف کند، استفاده مجدد از قوانین موجود در صورت امکان توصیه می‌شود؛ البته باید بدانیم قبل از نوشتن کد CGA باید از مجموعه قوانین استفاده کرد. برای اعمال یک قاعده و قانون یا مجموعه‌ای از قوانین، ممکن است آن را از قسمت ناوبری به داخل شکل مورد نظر بکشیم که در شکل ۸ نشان داده شده است. با انتخاب گروهی از شکل‌ها قبل از کشیدن، ممکن است قانون را همزمان به تعدادی شکل اختصاص دهیم. گزینه‌های مختلفی برای انتخاب اشکال بر اساس لایه وجود دارد یا قانون شروع را می‌توان با کلیک راست روی یک شکل پیدا کرد. پس از تخصیص یک قانون، یک تأخیر کوتاه وجود دارد درست زمانی که قانون برای ایجاد یک مدل تدوین و ارزیابی می‌شود. اگر بخواهیم کنترل بیشتری بر روی مدل‌سازی داشته باشیم، Inspector شامل گزینه‌های دقیق‌تری از جمله فایل قانون CGA، قانون شروع، و ویژگی‌های قانون ذکر شده قبلی برای ترسیم شکل است.

قوانین نوشتاری

در حالی که افسانه‌های «کدگذار» و «مهندسین نرم‌افزار» ممکن است برنامه‌نویسی را به یک هنر الهی ارتقا داده باشد، اما واقعیت بسیار ساده‌تر از یک هنر الهی است. CGA در مقایسه با زبان پایتون، زبان ساده‌تری است که مبتنی بر چند عملیات اساسی است که مکرراً برای نوشتن یک قانون اعمال می‌شوند. ما متوجه شدیم که دانشجویان مقطع کارشناسی می‌توانند قوانین خود را پس از چند جلسه با سیتی اینجین ایجاد کنند. کسانی که تجربهٔ زبان‌های پیچیده‌ای مانند C یا C++ را دارند، باید روش CGA انجام کارها را بیاموزند که کاربردی‌تر از آنچه قبلاً به آن عادت کرده‌اند است. گویش CGA مورد استفاده در سیتی اینجین از نسخهٔ ارائه شده در انتشار دانشگاهی و اولیه (Müller et al. 2006) تکامل یافته است؛ و هنگام مقایسهٔ قوانین با نسخه‌های مختلف باید دقت لازم رعایت شود. ما در اینجا از فرصت استفاده می‌کنیم تا اصطلاح "شکل" را در سیتی اینجین تشریح کنیم. از این برای توصیف هر دو شکل ورودی (شرح داده شده در بخش‌های قبلی) و اشکالی که بین قوانین در CGA منتقل می‌شوند، بیش از حد استفاده شده است. سیتی اینجین به این اشکال میانی به عنوان "اشکال CGA" اشاره می‌کند؛ و در اینجا از

اصطلاح هندسه استفاده خواهیم کرد. این سردرگمی تأسف‌بار تا حدودی ناشی از منشاء آکادمیک سیتی اینجین است، جایی که اشکال ورودی ما وجود نداشت.

یک فایل قانون CGA یک سند متنی حاوی مجموعه‌ای از قوانین است. یک قانون مشابه یک تابع یا روش در سایر زبان‌های برنامه‌نویسی است. هر قانون با نام و مجموعه‌ای از پارامترها مشخص می‌شود: X(1) قانون متفاوتی با X(1,2) است. همانطور که قانون اجرا می‌شود، می‌تواند عملیات‌های مختلف و همچنین قوانین دیگر را فراخوانی کند. این عملیات نظیر توابع کتابخانه در سایر زبان‌های برنامه‌نویسی است. همانطور که قوانین اصلی از عملیات مذکور برای ایجاد هندسه‌های جدید استفاده می‌کنند، هر کدام را با عنوان قانون فرزند طبقه‌بندی می‌کنند. اگر این قانون وجود خارجی داشته باشد، سپس قوانین بر روی هندسه‌ای تحت عنوان هندسهٔ جدید اجرا می‌شود. برخلاف تعریف‌های ارائه شده دانشگاهی CGA (مولر و همکاران، ۲۰۰۶)، هیچ مفهومی از اولویت قوانین برخوردار نیست؛ قوانین صرفاً بر اساس قانون اصلی آنها ارزیابی می‌شوند. هر قانون یک قطعه از هندسه را به هندسه‌های جدید (یا هیچ چیز) تبدیل می‌کند؛ نتیجه یک مدل شبکه‌ای سه بعدی متشکل از تمامی هندسه‌ها است که نمی‌توان بیشتر از این در جهت تشکیل هندسه پیش رفت. هندسهٔ اولیه همان شکل ورودی به مدل‌سازی است که قانون اولیه (گاهی اوقات با حاشیه‌نویسی @Startrule مشخص می‌شود) روی آن اعمال می‌شود. این قانون همچنین به ویژگی‌ها دسترسی دارد، که اجازه می‌دهد تا رفتار قانون توسط کاربر یا منبع داده سفارشی شود. ویژگی‌ها و پارامترها به همان روشی استفاده میشوند که زبان‌های برنامه‌نویسی دیگر از متغیرها برای سفارشی‌کردن رفتار آن استفاده می‌کنند. ارزش و مقدار اکثر مشخصه‌ها را می‌توان با عملیات‌های مختلف تنظیم کرد و خواند. گاهی اوقات ویژگی‌ها به عنوان یک گزینه و محتوا اضافی برای عملیات با هدف تعریف و اصلاح رفتار در نظر گرفته می‌شوند. به عنوان مثال، جهت‌گیری غالب و اطلاعات مبدأ در ویژگی‌های scope و pivot کدگذاری می‌شوند. الگوی معمول برنامه‌نویسی در CGA این است که هندسه را به طور مکرر بسط و سپس تقسیم کنید. هنگامی که عملیات تقسیم نسبت به محور y انجام شود، این راستا نسبت به جهت‌گیری ارائه شده توسط محدوده و مکان‌های محوری ذخیره شده در ویژگی‌ها ارائه شده است. قانون ایجاد یک مدل ساختمانی ممکن است با شکل یک قطعه زمین شروع شود، و با عملیات اکسترود (Extrude) گسترش یابد تا هندسه منشوری به بلندی ساختمان ایجاد شود، و سپس از عملیات comp برای تقسیم منشور به وجوه مختلف استفاده شود. قسمت بالایی مدل به سمت بالا منبسط می‌شود تا سقفی با عملیات roofGable ایجاد شود، در حالی که وجوه جانبی با استفاده از عملیات تقسیم به طبقه

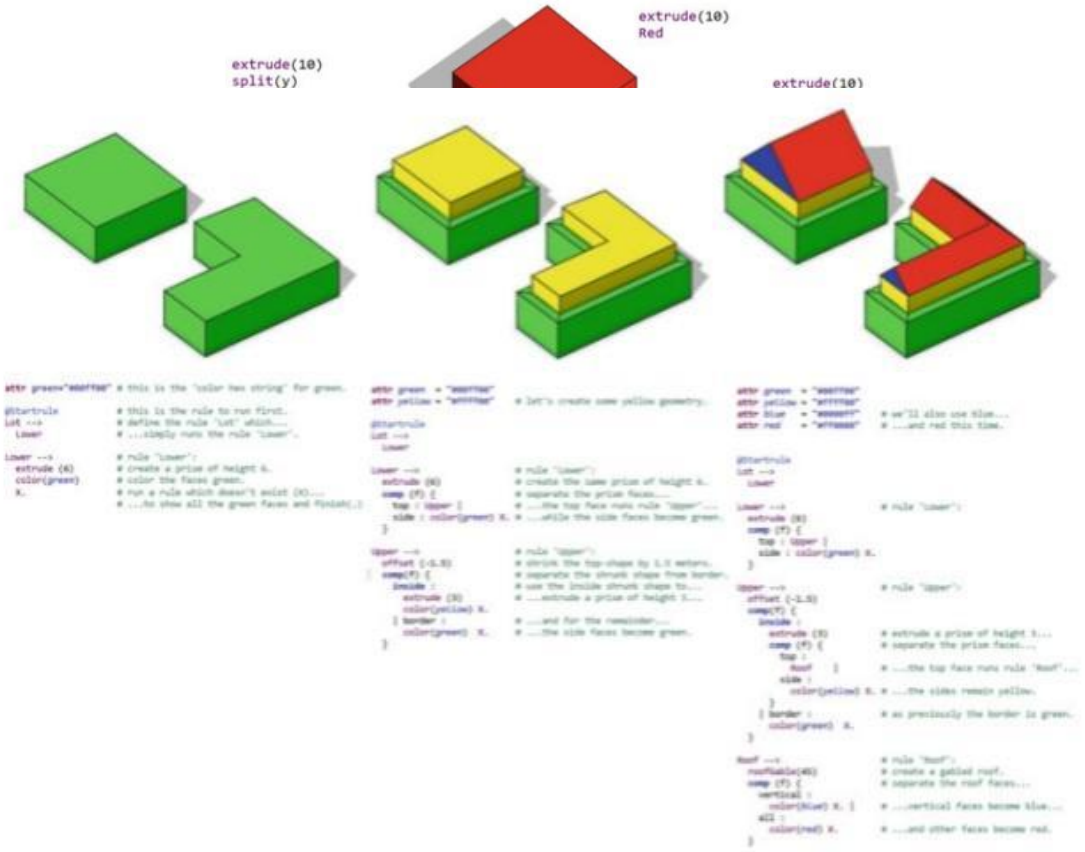
و سپس پنجره تقسیم می‌شوند. عملیات اکسترود (Extrude) دیگری در نهایت پنجره‌ها را به داخل نما فرو می‌برد. ما همچنان به مطالعهٔ این گونه عملیات با جزئیات بیشتر ادامه می‌دهیم.

عملیات

یادگیری نوشتن قوانین CGA عمدتاً فرآیند یادگیری عملیات‌های مختلف و تأثیرات آنها بر هندسه و ویژگی‌ها است. در حالی که پیچیدگی قوانین موجود می‌تواند برای کاربر جدید طاقت‌فرسا باشد، مجموعهٔ فشرده عملیات *CGA* یک منحنی یادگیری کم عمق را ارائه میدهد. CGA یک زبان برنامه‌نویسی است که برای انجام یک کار همچون طراحی مدل‌سازی محیط‌های شهری و نه چیزهای دیگر طراحی شده است. به همین دلیل، ما آن را به عنوان یک زبان خاص دامنه (برنامه‌نویسی) (DSL) توصیف می‌کنیم. برای دامنه‌های دیگر، زبان‌های برنامه‌نویسی دیگری وجود دارد: ما ممکن است از L Systems (Prusinkiewicz 1986) برای تولید گیاه یا از URDF (2019) برای ایجاد ربات استفاده کنیم. از آنجایی که CGA یک DSL است، عملیات آن به دقت برای حوزهٔ شهری تنظیم شده است. تلاش‌های نظری بسیاری برای یافتن مجموعه‌ای فشرده و در عین حال گویا از عملیات انجام شد. در مقابل، زبان‌های مدل‌سازی رویه‌ای همه‌منظوره، مانند Houdini (2019) و Rhino (2019)، در یک حوزهٔ تخصصی نیستند و یادگیری آن شامل عملیات پیچیدهٔ بسیاری هستند.

شکل ۹ تعدادی از عملیات کلیدی سیتی اینجین را معرفی می‌کند. با اعمال مکرر این عملیات، می‌توانیم هندسه‌های شهری متنوع‌تری ایجاد کنیم. به عنوان مثال، عملیات بازگشت، Extrude، comp، و roofGable را می‌توان برای ایجاد خانه‌ای با طبقهٔ بالایی فرورفته و سقف شیروانی استفاده کرد، مانند شکل ۱۰.

یکی از مشاهدات مهم این است که CGA شامل عملیات حلقه یا تکرار نیست. برای دستیابی به هندسه تکرارشونده (مانند پنجره‌های نمای ساختمان یا درختان در امتداد خیابان)، می‌توانیم از عملیات تقسیم با تعدیل‌کننده ستاره (*) برای تقسیم یک شکل اصلی به تعداد تکراری از اشکال فرزند با قوانین مشابه استفاده کنیم. این موضوع در شکل ۱۱ نشان داده شده است.



شکل ۹. CityEngine بیش از ۶۰ عملیات دارد. در اینجا، ما انتخابی را که برای ورودی به شکل مربع به کار گرفته‌ایم (قسمت خاکستری رنگ)، و همچنین استفاده از نمونه را نشان می‌دهیم. قوانین بی‌اهمیت با نام رنگ‌ها (قرمز، آبی و غیره) نشان داده نمی‌شوند، اما در فایل قانون گنجانده می‌شوند.

شکل ۱۰. توجه داشته باشید که چگونه با یک قانون ساده شروع به مدل‌سازی می‌کنیم و به تدریج آن را گسترش می‌دهیم تا هندسه‌های پیچیده‌تری را با پیروی از پارادایم بسط–سپس–تقسیم ایجاد کنیم. متن سبز رنگ بر روی نظراتی تأکید می‌کند که توسط سیتی اینجین نادیده گرفته می‌شوند، اما به انسان در درک کدگذاری کمک می‌کند.

در مثال آخر ما هندسه‌ای برای خیابان‌ها ایجاد می‌کنیم. برای ایجاد خطوط بزرگراه، می‌خواهیم محور طولانی خیابان‌ها را که ممکن است خمیده باشند، تقسیم کنیم. نوع UV عملیات تقسیم این امر را محقق می‌کند. در نهایت، ممکن است بخواهیم نقشه‌های بافت (تصاویر bitmap) را به جای رنگ‌های ساده با استفاده از عملیات بافت، مانند شکل ۱۲، به هندسه خود اضافه کنیم.

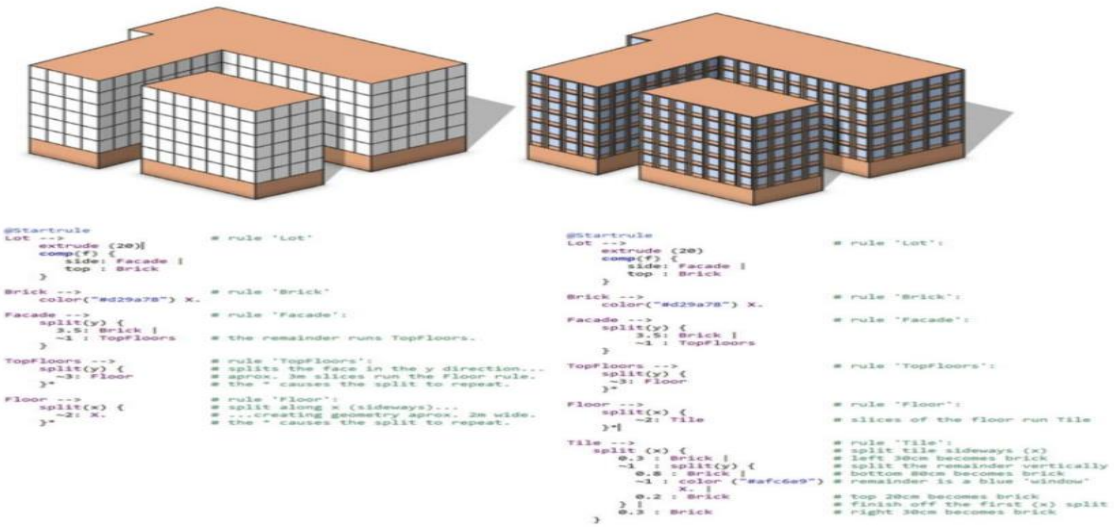
جریان‌کاری مدل‌سازی

ایجاد فایل‌های با قوانین بیشتر می‌تواند برای کسانی که تازه کدنویسی می‌کنند، کار هولناکی باشد. این مهارتی است که برای تمرین و یادگیری به زمان نیاز دارد، اما زمانی که دانش کمی به دست می‌آید اغلب باعث خوشحالی می شود:

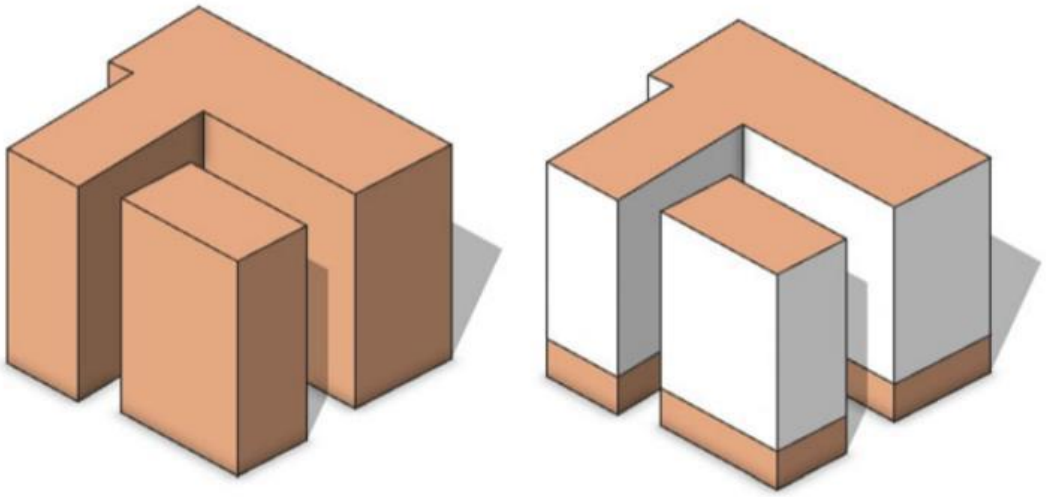
برنامه‌نویس، مانند شاعر، فقط اندکی از چیزهای فکری ناب دور است. (Brooks, 1995)

این هیجان اولیه اغلب برای برنامه‌نویسان بی‌تجربه مشکل ایجاد می‌کند؛ اعتماد بیش از حد باعث عدم درک ویژگی‌های یک پایه کد در حال رشد می‌شود. از آنجایی که بسیاری از مشکلات کوچک در کدگذاری‌ها ("اشکالات") ریشه دوانده است، انجام تغییرات حتی کوچک می‌تواند بسیار زمان‌بر باشد، چرا که نیازمند تغییر در کدها است. ما می‌توانیم برخی از راهنمایی‌ها و ابزارهای کلی را که می‌توانند به ما در ساختن برنامه‌های بزرگ CGA کمک کنند، ارائه می‌دهیم:

- بخش‌های کوچک کدها را در یک زمان مناسب بنویسید و مرتباً آنها را مورد آزمایش قرار دهید. این امر باعث می‌شود ردیابی و جداسازی مسائل بسیار سریعتر انجام شود. اگر نمی‌توانید برخی از رفتارها را درک کنید، در این راستا اغلب پیش می‌آید که قبل از اجرای کد، کد زیادی نوشته شده است، تا کد مناسب یافت شود.
- قوانینی را که به طور مجدد قابل استفاده هستند را ایجاد کنید. قانون کوچکی که ایجاد کرده‌اید و «پنجره برند Acme» را ایجاد می‌کند، اگر در فایل جداگانه نگهداری شود، ممکن است دوباره مورد استفاده قرار گیرد. CGA عملکرد واردات را برای تسهیل استفاده از این قانون پنجره در سایر فایل‌های قانون فراهم می‌کند.
- اسناد CGA ارائه شده را بخوانید (منوی راهنما مرجع CGA).



شکل ۱۱. مثالی از استفاده از قانون تقسیم برای تقسیم‌بندی یک نما برای ایجاد پنجره.



```
@Startrule
Lot -->                 # rule 'Lot':
    extrude (20)        # create a prism of height 20 meters.
    comp(f) {           # sepearate the faces of the prism...
       side: Brick |    # ...sides run the rule "Brick"...
       top : Brick      # ...as does the top.
    }

Brick -->               # rule 'Brick':
  color("#d29a78") X.   # color the shape, and display.
```

```
@Startrule
Lot -->                 # rule 'Lot':
    extrude (20)
    comp(f) {
       side: Facade |   # run the 'Facade' rule on the sides.
       top : Brick
    }

Brick -->               # rule 'Brick'
    color("#d29a78") X.

Facade -->              # rule 'Facade':
    split(y) {          # split the facade in the y (up)...
      3.5: Brick |      # ...the bottom 3.5 meters is brick...
      ~1 : X.           # ...the remainder is blank.
    }
```

ادامهٔ شکل ۱۱

- یادگیری جزئیات برنامه‌نویسی و درک آن، منجر به نوشتن هرچه راحت‌تر کدها می‌شود اما بعد از یک هفتهٔ زمانی، هنگامی که جزئیات را فراموش کرده‌اید، درک آن دشوار است. از بخش تفسیرات کد (بخش‌هایی از کد که رایانه آنها را نمی‌بیند) برای یادداشت‌برداری برای خود و اطلاع خوانندگان آینده استفاده کنید. این توضیحات و تفسیرات سیتی اینجین را می‌توان به دو روش ایجاد کرد:

//همهٔ نوشته‌ها در این خط یک مفهوم است

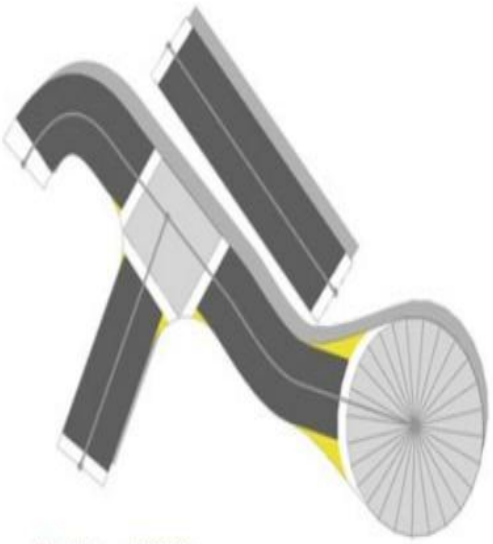
/* همهٔ نوشته‌ها بین دو ستاره یک مفهوم است */

- مجموعهٔ فایل‌های قوانین می‌توانند بسیار وسیع باشند، یا توسط افراد متعدد نوشته شده باشند، یا چندین نسخه داشته باشند، و یا حتی می‌توانند منجربه توسعهٔ این قوانین در شاخه‌ها و زمینه‌های مختلف شوند. به این دلایل، برنامه‌نویسان معمولاً از یک سیستم کنترل نسخه (مانند git با نام غیر حساس (git 2019)) برای مدیریت کد خود استفاده می‌کنند. از میانبرهای صفحه کلید و منوهای زمینه (راست کلیک) موجود در سیتی اینجین آگاه باشید. برای مثال، اگر شکلی را با یک قانون انتخاب کرده‌اید و در حال ویرایش قانون در ویرایشگر متن هستید، Ctrl + S و سپس Ctrl + G (در ویندوز یا لینوکس؛ از کلید فرمان به جای Ctrl در OS X استفاده کنید) ذخیره می‌شود و شکل سه بعدی به روز شده را نشان می‌دهد. در نمای سه بعدی، کلید F نمای کلی را برای نشان دادن شیء انتخاب شده حرکت می‌دهد یا کلید F9-F12 طبقه‌های مختلفی از اشیاء را نشان داده و پنهان می‌کند.

فراتر از آداب برنامه‌نویسی عمومی، سیتی اینجین چندین مکانیسم سفارشی برای کمک به نوشتن قوانین CGA ارائه می‌دهد. صفحات سلسله مراتبی مدل، نموداری از برنامه‌های مختلف قوانین را نشان می‌دهد (پنجره ← نمایش سلسله مراتب مدل ۱۳). این دکمه ابزار Inspect Model را نشان می‌دهد که می‌تواند برای انتخاب ساختمانی برای تجزیه و تحلیل استفاده شود (توجه داشته باشید که Inspect Model یک عملکرد متفاوت با پانل Inspector است.) نمودار به دست آمده در پانل نشان داده شده است، که کاربرد قوانین با یک فلش خاکستری نشان داده شده است. خطوط جفت قانون اصلی/جدید را به هم متصل می‌کنند. با انتخاب یک قاعده در نمودار، نمای سه بعدی هندسه حاصل را مشخص می‌کند و صفحات scope، pivot و trim برای اعمال قانون صحیح نشان می‌دهد. با کلیک راست بر روی یک گره از قانون در نمودار، گزینهٔ پرش را به بخش متناظر از CGA ارائه می‌دهد. یک قانون CGA معمولاً در مکان‌های مختلف اعمال می‌شود و بنابراین چندین بار در نمودار ظاهر می‌شود.

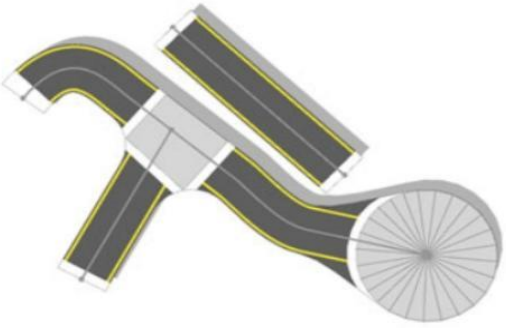
یکی دیگر از ابزارهای ارائه شده توسط سیتی اینجین، Façade Wizard است. (پنجره ← نمایش Façade Wizard). برای یک نمای ۲ بعدی، این موضوع به ایجاد عملیات تقسیم و اکسترود (Extrude) مورد نیاز برای یک نمای خوب پارامتری کمک می‌کند.

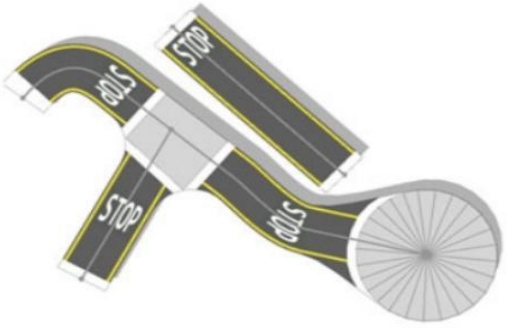
```
attr white  = "#ffffff"   # let's define white and gray...
attr gray   = "#555555"   # ...hex string colors.

@Startrule
Street -->                # for all Street (dynamic) shapes.
  split(u,uvSpace,1) {    # split the start of the street...
    0.7 : Create(white) | # ...and make it white.
     ~1 : Create(gray)    # the remainder is gray tarmac.
  }

Create(c) -->
    color (c)             # a utility rule to color...
    X.                    # ...and create geometry.
```



```
attr white  = "#ffffff"
attr gray   = "#555555"
attr yellow = "#ffff00"         # let's use yellow geometry too.

attr laneWidth = 3.5            # from the dynamic shape attr.
attr streetWidth = 7            # from the dynamic shape attr.

@Startrule
Street -->
  split(u,uvSpace,1) {
    0.7 : Create(white) |
     ~1 :
       split(u,uvSpace,2) {     # repeat the trick with UV set 2...
         0.7 : Create(white) |  # ...for white line at end of street.
         ~1 : WithGutters       # continue to process gutters.
       }
  }

/* gutters are wider parts of street shapes near junctions. */
WithGutters -->
  split(v,uvSpace,0) {          # split in v-direction on uv set 0 .
    -geometry.vMin :            # this value...
     Create(yellow)  |          # ...splits off left yellow gutter.
    rint ( streetWidth          # this equation gives the width of...
           / laneWidth) :
     Create(gray) |             # ...the central street.
     ~1 : Create(yellow) }      # split off right yellow gutter.

Create(c) -->
   color (c)
   X.
```

شکل ۱۲. نمونه‌ای از ایجاد مدل برای اشکال خیابانی. قانون تقسیم با پارامتر UV برای تقسیم مناطق منحنی استفاده می‌شود. سه مجموعهٔ UV مختلف خیابانی از دو طرف اشکال جدا می‌شوند. در نهایت، دستورات عادی UV و بافت اطلاعات تصویری دو بعدی، علامت‌های «توقف» را ایجاد می‌کنند



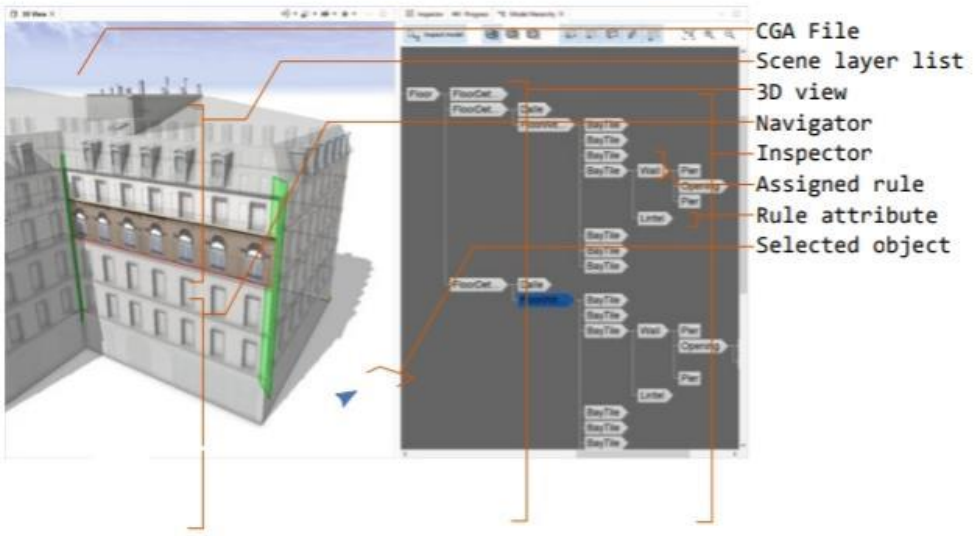



```
attr white  = "#ffffff"
attr gray   = "#555555"
attr yellow = "#ffff00"

attr laneWidth = 3.5
attr streetWidth = 7

@Startrule
Street -->
  split(u,uvSpace,1) {
    0.7 : Create(white) |
      ~1 :
        split(u,uvSpace,2) {
          0.7 : Create(white) |
          ~1 : WithGutters
      }
  }

WithGutters -->
  split(v,uvSpace,0) {
    -geometry.vMin :        # on second thought, let's...
      Create(gray)  |       # ...make the gutters gray.
    rint ( streetWidth
           / laneWidth) :
      WithoutGutter |       # create non-gutter gometry.
      ~1 : Create(gray) }

WithoutGutter -->
  split(v,unitSpace,0) {    # split along street edges (v).
    0.4 : YellowLine |      # to create left yellow line...
    ~1 : Create(gray) |     # ...central area fills remainder.
    0.4 : YellowLine        # and right yellow line.
    }

YellowLine -->
  split(v,unitSpace,0) {    # split along street edge (v).
    0.1 : Create(gray) |    # 5cm of gray...
    ~1 : Create(yellow) |   # ...then a 10cm yellow strip...
    0.1 : Create(gray)      # ...and a final 5cm of gray.
}

Create(c) -->
    color (c)
    X.
```

```
attr white  = "#ffffff"
attr gray   = "#555555"
attr yellow = "#ffff00"

attr laneWidth = 3.5
attr streetWidth = 7

@Startrule
Street -->
  split(u,uvSpace,1) {
    0.7 : Create(white) |
      ~1 :
        split(u,uvSpace,2) {
          0.7 : Create(white) |
          ~1 : WithGutters
      }
  }

WithGutters -->
  split(v,uvSpace,0) {
    -geometry.vMin :
      Create(gray)  |
    rint ( streetWidth
           / laneWidth) :
      WithoutGutter |
      ~1 : Create(gray) }

WithoutGutter -->
  split(v,unitSpace,0) {
    0.4 : YellowLine |
    ~1 : WithoutStop |        # create stop markings.
    0.4 : YellowLine
    }

YellowLine -->
  split(v,unitSpace,0) {
    0.1 : Create(gray) |
    ~1  : Create(yellow) |
    0.1 : Create(gray)
}

WithoutStop -->
  split(u,uvSpace,0) {        # split from the start of the street.
    -geometry.uMin :          # any junction areas...
      Create(gray) |          # ...become gray.
    ~1 :                      # everything else...
     split(u,unitSpace,0) {   # ...is split again in units (meters)...
       3 : Stop |             # ...to create a stop sign.
       ~1 : Create(gray) }    # everything else is a regular street.
  }

Stop -->
  normalizeUV (0, uv,         # stretch the image over all..
    collectiveAllFaces )      # ...the current geometry.
  texture("images/stop.png")  # texture with the stop.png image.

Create(c) -->
    color (c)
    X.
```

ادامهٔ شکل ۱۲

برای ارائهٔ یک قانون سیتی اینجین به کاربر نهایی در قالبی مناسب، از یک مجموعه حاوی قوانین استفاده کنید. این را می‌توان با انتخاب فایل CGA برای صدور به نرم‌افزار هدایتگر، کلیک راست و انتخاب Share As... ساخت، منابع و فراداده‌های اضافی در کادر محاوره‌ای-پنجره‌ای مشخص شده‌اند. به این ترتیب، فایل .RPK حاصل ممکن است شامل بسیاری از فایل‌های CGA مجزا و منابع دیگر مانند داده‌های موجود در فایل‌های متنی و تصاویر بافت باشد. چنین مجموعه‌ای به راحتی به عنوان یک فایل منفرد توزیع می‌شود و Esri یک سیستم ابری برای توزیع



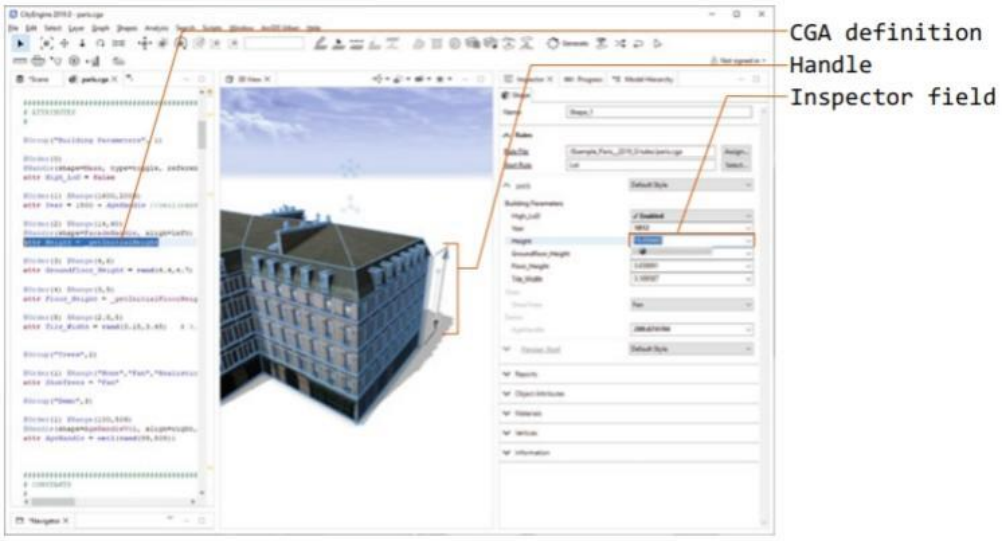
شکل ۱۳. سلسله مراتب مدل ابزار بسیار مفیدی برای تجسم هندسه است. سمت چپ: نمای سه بعدی یک مدل از شکل اول. قانون انتخاب شده مشخص شده و با یک رنگ ثابت ارائه می‌شود. نیز تجسم می‌شوند. راست: سلسله مراتب قانون، قاعده‌ای که Trim و Pivot،scopeصفحه‌های هندسه انتخاب شده را ایجاد کرده است، مشخص می‌کند. با کلیک بر روی یک قانون دیگر، هندسه (مرکز بالا) توجه کنید که برای فعال Inspect Modelمرتبط با آن قانون نشان داده می‌شود. دکمه کردن عملکرد مدل سلسله مراتبی استفاده می‌شود.

۳-٤-۳۵: ویژگی‌ها و مشخصه‌ها

پس از ساختن قوانین خود و اختصاص دادن آنها به اشکال خود، اغلب علاقه‌مندیم که بیان قاعده را با استفاده از ویژگی‌ها جهت بالابردن کارآیی نرم‌افزارها را تغییر دهیم.

ویژگی‌ها برای اصلاح ارزیابی مدل‌ها در یک برنامه تحت عنوان قاعده و قانون استفاده می‌شوند. آنها امکان تعمیم یک قانون را می‌دهند. به عنوان مثال، تعدادی از ساختمان‌های مشابه را در نظر بگیرید که از مواد مختلف با جنس مختلف ساخته شده‌اند؛ به جای یک قانون جداگانه برای هر ماده، ممکن است از یک قانون واحد با یک ویژگی برای مصالح ساختمانی استفاده کنیم. به طور کلی؛ این مشخصه‌ها می‌توانند رفتارهای یک قانون را کنترل کنند، اما به طور معمول، ویژگی‌هایی مانند ارتفاع ساختمان، سن، یا تعداد عابران پیاده ایجادشده در پیاده‌روها را نظارت می‌کنند. سیتی اینجین، بسیاری از ویژگی‌های موجود برای شکل و قانون انتخاب شده را در پنل Inspector نشان می‌دهد (شکل ١٤)؛ البته برخی از قوانین دارای ویژگی‌های بسیار زیادی هستند. ارزش ویژگی‌های پیش فرض توسط قانون تنظیم می‌شود. با این حال، کاربران می‌توانند منبع ویژگی‌ها را نادیده بگیرند تا این قانون به ورودی‌های مختلف پاسخ دهد. ویژگی‌ها در سیتی اینجین منابع متعددی دارند و وابستگی‌های متقابل بین آنها می‌تواند پیچیده باشد. منابع این ویژگی‌ها عبارتند از:

- منبع قانون (پیش‌فرض قانون)، رفتار ویژگی پیش‌فرض
- منبع کاربر
- منبع شکل (ویژگی‌های شیء)
- تصویر یا شکل محور (ویژگی‌های لایه)

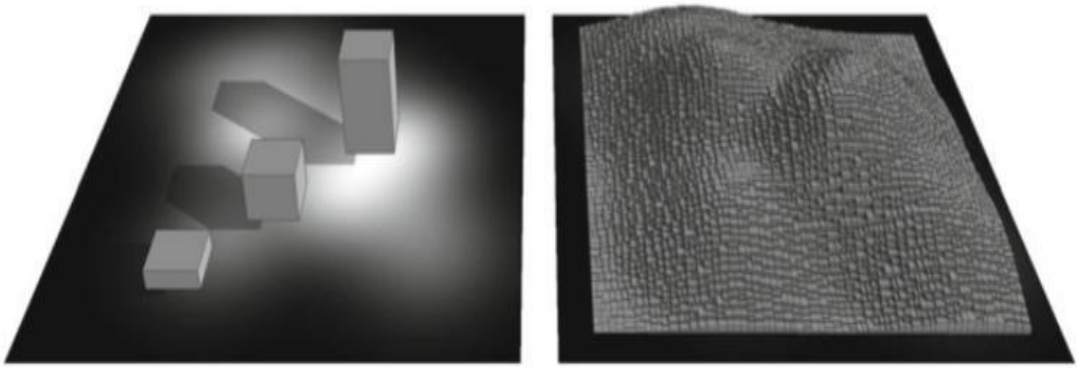


شکل ۱۴. ویژگی‌ها در فایل *CGA* (سمت چپ) تعریف می‌شوند و یا با دسته‌ها (مرکز) یا با استفاده از *Inspector* (راست) ویرایش می‌شوند.

اینها را می‌توان با کلیک بر روی فلش رو به پایین در کنار یک ویژگی در صفحه Inspector و انتخاب Connect Attribute... انتخاب کرد. ارزش ویژگی‌های منبع این قانون و قواعد در فایل قانون CGA آورده شده است. این ویژگی‌ها می‌توانند تصادفی باشند؛ از این ویژگی می‌توان برای افزودن تغییرات به یک قانون که بارها اعمال می‌شود استفاده کرد؛ به عنوان مثال، هر ساختمان ممکن است با یک قانون مشخص ایجاد شود، اما ارتفاعی که به طور تصادفی بین ۱۰ تا ۲۰ متر انتخاب شده است [attr height = rand (10،۲۰)] داده شود. برای اینکه کاربران بتوانند یک ویژگی را بدون ویرایش فایل CGA تغییر دهند، ویژگی‌های ویرایش شده در Inspector به ویژگی‌های منبع کاربر تغییر می‌کنند. با این حال، ممکن است بخواهیم ویژگی‌ها را از منابع دیگری که ممکن است توسط داده‌ها هدایت می‌شوند، بگیریم. هنگامی که یک شکل انتخاب می‌شود، ویژگی‌های شیء در Inspector (در زیر عنوان Object Attributes) قابل مشاهده است. ویژگی‌های شیء می‌توانند از منابع داده ورودی به

دست بیایند (به عنوان مثال، داده‌های OpenStreetMap اغلب مشخصهٔ ارتفاع ساختمان را برای هر قطعه‌ای مشخص می‌کند) یا توسط اشکال پویا این ویژگی‌ها تعریف می‌شوند (به عنوان مثال، ویژگی‌های اتصال Start و End به طور خودکار به اشکال خیابان اضافه می‌شوند تا انواع اتصالات مجاور را مشخص کنند). ویژگی‌های لایه مقادیر و ارزش خود را از اشکال دیگر یا یک bitmap نمونه‌برداری می‌کنند، همانطور که در شکل ۱۵ نشان داده شده است. برای مثال، می‌توانیم با استفاده از نقشهٔ ارتفاع جغرافیایی به عنوان ارتفاع مرجع که توسط LIDAR هوایی گرفته شده است، ویژگی ارتفاع ساختمان را مشخص کنیم. به این ترتیب، می‌توانیم یک قانون را با استفاده از چندین منبع داده مختلف کنترل کنیم. این رویکرد به طور قابل توجهی دقت هندسه حاصل را در یک کانال ارتباطی رویه‌ای صرفاً مبتنی بر قانون بهبود می‌بخشد.

در نهایت، دانستن این نکته مفید است که ویژگی‌های چندین شکل را می‌توان با انتخاب چندین شکل به طور همزمان ویرایش کرد، مفید است. به طور متناوب، با کلیک راست بر روی اشکال در نمای سه بعدی، گزینه‌های انتخاب خودکار و گوناگون امکان انتخاب بسیاری از اشکال در یک لایه را فراهم می‌کند. Inspector ویژگی‌های موجود را برای انتخاب کلی نشان می‌دهد، و ویرایش یک ویژگی یا منبع، تغییر ویژگی را برای همه اشکال انتخاب‌شده اعمال می‌کند.



شکل ۱۵. سمت چپ: یک تصویر سیاه و سفید وارد شده به مدل به عنوان یک بافت برای تشخیص ویژگی) ساده هستند، استفاده Extrudeارتفاع سه شکل معلق مستطیلی، که هر کدام دارای یک قانون اکسترود (می‌شود. قسمت‌های سفید بافت به عنوان قسمتی با مقادیر بزرگ نمونه‌برداری می‌شوند که به صورت مکعب‌های بلند تعریق می‌شوند؛ نواحی سیاه بخش‌هایی با مقادیر کم هستند که به مکعب‌های کوتاه تبدیل می‌شوند. سمت راست: به این ترتیب، ما می‌توانیم ویژگی‌هایی را از یک بافت نمونه‌برداری کنیم تا ارتفاع ساختمان (یا هر ویژگی دیگر) را در یک شهر با توجه به یک تصویر تغییر دهیم.

جست‌وجو در فضای طراحی

به عنوان یک طراح که از سیتی اینجین استفاده می‌کند، باید بیان شود که در این زمینه تصمیمات بسیاری وجود دارد که باید به آن‌ها رسیدگی شود. قوانین پیچیده صدها ویژگی را ارائه می‌دهند، و این قوانین باید با نیازهای کاربر، دیدگاه‌های هنری و ملاحظات عملی هماهنگ شوند. از آنجایی که هر ویژگی اضافی تعریف شده به فضای طراحی بُعدی می‌افزاید، بررسی قوانین بزرگ و پارامتری می‌تواند زمان زیادی را از ما بگیرد. علاوه بر این، ممکن است بخواهیم سناریوهای متعددی را طراحی کنیم: قواعد، ویژگی‌ها و اشکال مختلف مشکلی را حل می‌کند که می‌خواهیم آن‌ها را در کنار هم مقایسه کنیم. سیتی اینجین یک رابط پایتون برای برنامه‌نویسان پیشرفته برای کنترل ویژگی‌ها (و بسیاری از عناصر دیگر که در صحنه‌ها و شرایط دیگری حضور دارند) با استفاده از کد سفارشی فراهم می‌کند. کاربردهای معمولی ایجاد انیمیشن‌های ویدیویی از ویژگی‌ها یا اجرای الگوریتم‌های جستجوی فضای طراحی سفارشی است. با این حال، اکثر کاربران می‌خواهند از چنین پیچیدگی‌هایی اجتناب کنند. تعدادی ابزار برای

کمک به کشف بصری این فضای طراحی ویژگی‌ها ارائه می‌دهد. با توجه به تعداد زیاد ویژگی‌ها در قاعده‌ای مانند مثال پاریس، دیدن نمایش بصری آن ویژگی‌ها در کنار مدل سه بعدی اغلب مفید و کاربردی است. دستگیره‌ها این قابلیت و کاربرد را با نشان دادن ویژگی‌هایی همچون ارتفاع به عنوان کنترل‌گر در نمای سه بعدی ارائه می‌دهند. سیستم handle از خطوط ابعادی نمودارهای مهندسی الهام گرفته شده است که توسط کلی و همکاران (۲۰۱۵) معرفی شده است. هنگامی که یک مدل با قابلیت handle در نمای سه بعدی انتخاب می‌شود، بسته به کاربردی که ما انتظار داریم، دستگیره‌ها در لبه‌های مدل نشان داده می‌شوند. دستگیره‌های مختلف انواع گوناگونی از مقادیر را کنترل می‌کنند: ضامن‌های بولی (مربوط به مقادیر منطقی درست و نادرست)، صفحهٔ مدرج چند گزینه‌ای، فاصله‌ها که به عنوان ارزش و مقدار خطوط ابعاد تعریف می‌شوند، دسته‌های مثلثی انتخاب‌گر رنگ در دسترس هستند. مکانهای دستگیره ها، تعریف رفتارها در جهت سازماندهی و یا شاید تغییر الزامات یک سیستم نرم‌افزاری، و ظاهر مدل با حاشیه‌نویسی @Handle در فایل قانون CGA تعریف می‌شوند. آنها توسط سازندهٔ قوانین طراحی شده‌اند و تنها در صورتی در دسترس ما هستند که نویسندهٔ قانون اجازه استفاده از آنها را بدهد. نویسند قانون اغلب برای جلوگیری از شلوغی بیش از حد صفحه، تنها ویژگی‌های پرکاربرد را با استفاده از دستگیره‌ها در معرض نمایش بگذارد.

دستگیره‌ها مقدار و ارزش یک ویژگی را در ارزیابی کلی قانون برای یک شکل تعیین‌شده تغییر می‌دهند. موقعیت‌هایی وجود دارد که می‌خواهیم یک ویژگی را در ارزیابی قاعده و قوانین ویرایش کنیم، برای مثال، یک طبقه از یک ساختمان را بلندتر از بقیه کنیم یا مکان یک پنجره را در یک نمای بزرگ جابه‌جا کنیم. در این شرایط می‌توانیم از ویرایش‌های محلی استفاده کنیم. این شرایط به ما اجازه می‌دهند تا ویژگی‌ها را با دستگیره‌ها ویرایش کنیم. ویرایش‌های محلی با انتخاب ابزار ویرایش‌های محلی ایجاد می‌شوند؛ بسته به نحوهٔ ساختار این قانون، این ابزار ممکن است به ما اجازه دهد همهٔ ویژگی‌های محلی را در یک ردیف، ستون یا الگوهای پیچیده‌تر به طور همزمان ویرایش کنیم. ویرایش‌های محلی بیشتر توسط لیپ و همکاران در سال ۲۰۱۹ مورد بحث قرار گرفته است.

همانطور که ویژگی‌های قانون را اصلاح می‌کنیم، ممکن است در تلاش برای دستیابی به یک هدف عینی مانند سطح طبقه هدف برای یک ساختمان یا گروهی از ساختمان‌ها باشیم. مکانیسم گزارش دهی سیتی اینجین به قوانین اجازه می‌دهد تا چنین اطلاعاتی را جمع‌آوری کنند و سپس یک گزارش خلاصه‌وار برای هر مدل تهیه کنند. عملیات گزارش هر زمان که فراخوانی شود مقادیری را جمع‌آوری می‌کند و مجموع کل مدل را برمی‌گرداند [ممکن است

از گزارش عملیات (200 ,"area") استفاده کنیم.] مقادیر چندگانه (مساحت طبقه، حجم اتاق، و غیره) را می‌توان برای هر قانون جمع کرد و در Inspector به عنوان یک جدول نمایش داد. اگر از عملکرد داشبورد (نمایش گرافیکی) سیتی اینجین استفاده شود، این جداول را می‌توان به عنوان طیفی از نمودارها ارائه کرد که به طور خودکار به روز می‌شوند. آنها می‌توانند نتایج را روی همهٔ مدل‌های موجود در صحنه یا فقط مدل‌های انتخاب شده نشان دهند.

با صرف زمان برای افزودن گزارش به مدل‌های خود و استفاده از عملکرد داشبورد، امکان کاوش فضای طراحی به صورت تعاملی با طیف گسترده‌ای از کاربران فراهم می‌شود. به عنوان مثال، مشتریان ممکن است از توانایی استفاده از دستگیره‌ها برای ویرایش ارتفاع ساختمان و دریافت بازخورد فوری در مورد اثرات زیربنای موجود و هزینه‌های ساخت، قدردانی کنند.

فراتر از تجزیه و تحلیل نامنظم گزارش شده، ممکن است به پیامدهای بصری طرح‌های خود علاقه‌مند باشیم. سیتی اینجین طیف وسیعی از ابزارها را برای اندازه‌گیری فاصله و مساحت در صحنهٔ سه بعدی ارائه می‌دهد (شکل ۱۶)، اما جالب‌تر از همه، محاسبات قابلیت دید را ارائه می‌دهد؛ این قابلیت بخش‌هایی از مدل‌ها را مشخص می‌کند که از یک مکان خاص در یک میدان دید مشخص قابل مشاهده هستند و یا حتی نیستند.

در نهایت، سناریوها امکان مقایسه رویدادهای مختلف را به ما می‌دهد. هر سناریو می‌تواند حاوی لایه‌های مختلفی از محتوای مدل‌سازی در بالای یک صفحه پس‌زمینه مشترک باشد. به عنوان مثال، سه پیشرفت مختلف پیشنهاد شده برای یک بلوک شهری با ارتفاع متفاوت را می‌توان در حالی که فضای شهری اطراف آن حفظ می‌شود، نشان داد. یک سناریو را می‌توان برای کشف فضای طراحی جدید کپی و ویرایش کرد.

شکل ۱۶. ابزارهای تحلیل. سمت چپ: محاسبات میدان دید که نواحی قابل مشاهده (سبز) و مسدود شده (قرمز) را نشان می‌دهد. قسمت وسطی: ابزار اندازه‌گیری طول مسیر. سمت راست: ابزار اندازه‌گیری مساحت

آنسوی نرم‌افزار: سیتی اینجین: مسیر تغییر فرمت نرم‌افزار

پس از ایجاد اشکال، قوانین نوشته شده و پارامترهای تنظیم شده برای بازسازی سه بعدی خود با دقت بالا می‌خواهیم صحنه‌های سیتی اینجین خود را مشاهده کنیم و به اشتراک بگذاریم.

لازم به ذکر است که نمای سه بعدی سیتی اینجین می‌تواند تصاویری با مدل نورپردازی با کیفیت مناسب ایجاد کند. گزینه‌هایی در صفحه viewport (تنظیمات مشاهده) برای فعال کردن سایه‌ها (مانند سایهٔ ایجاد شده توسط خورشید)، همپوشانی محیط (سایه‌های دقیق‌تر در چین‌های هندسی) و میدان دید (زاویهٔ صحنه‌ای که می‌بینیم) وجود دارد. تصاویر را می‌توان از فضای سه بعدی (نشانک‌ها ذخیرهٔ عکس فوری...) ذخیره کرد.

تصویر گرافیکی در نمای سه بعدی سیتی اینجین یک تصویر گرافیکی بلادرنگ OpenGL است که مشابه آنهایی است که برای بازی‌های ویدیویی استفاده می‌شود. اگر می‌خواهیم پرداخت صحنهٔ ظاهری دقیق‌تری (PBR) داشته باشیم و آماده باشیم تا هر تصویر پرداخت صحنه شود، می‌توانیم از نوع شخص ثالث آن (مانند POV-Ray، LuxRenderer، موتور بازی Unity، Autodesk 3ds Max یا Blender استفاده کنیم.) برای ایجاد تصاویر گرافیکی دقیق‌تر استفاده کنیم. این رندرها به خودی خود قطعات پیچیده‌ای از نرم‌افزار هستند و مکانیک و هنر تنظیم نور و مواد برای ایجاد تصاویر واقعی فراتر از این فصل است. با این حال، در شکل ۱۷، ما رندر پیش فرض

سیتی اینجین را با رندر Cycles مبتنی بر فیزیکی در Blender مقایسه می‌کنیم. ما به کیفیت بالای شبیه‌سازی نور (انعکاس، سایه‌ها و توزیع مناسب رنگ‌ها) و ظاهر مواد توجه می‌کنیم. برای استفاده از یک رندر خارجی، باید مدل‌های خود را به صورت شبکه‌های سه بعدی از سیتی اینجین به نرم‌افزار دیگری انتقال دهیم. سیتی اینجین انواع فرمت‌های مختلف را برای انتقال مدل‌ها ارائه می‌دهد (فایل انتقال مدل): OBJ Wavefront یک فرمت تبادلی متداول است، اما فرمت‌های عجیب‌تر دیگر عبارتند از Collada، Autodesk FBX و Alembic. سپس یک کانال ارتباطی در برنامه‌های مدل‌سازی سه‌بعدی مانند Blender، وارد کردن شبکه‌های سه بعدی، تنظیم بافت‌ها و قرار دادن دوربین و چراغ‌ها است. در نهایت، تحت یک عملیات رندر که ممکن است در مدت زمان چند دقیقه یا حتی چند روز طول بکشد تا یک تصویر با کیفیت بالا تولید شود.

برای به اشتراک گذاشتن شبکه‌های سه بعدی با دیگران به صورت آنلاین به عنوان یک شیء سه بعدی، به جای تصاویر دو بعدی، چندین گزینه وجود دارد. گزینش به سرعت در حال رشدی از میزبان‌های سه بعدی (s Poly'Sketchfab, SketchUp 3D Warehouse, or Google) مبتنی بر وب وجود دارد که شبکه‌های OBJ را به صورت آنلاین میزبانی می‌کنند تا در یک مرورگر مشاهده شوند. با این حال، این سایت‌های سه‌بعدی معمولی از بسیاری از جزئیات از صحنه سیتی اینجین پشتیبانی نمی‌کنند.

Esri دو راه‌حل برای این مشکل ارائه می‌دهد: صادرکنندهٔ صحنه وب سیتی اینجین (فایل انتقال مدل) و برنامهٔ اختصاصی ArcGIS Urban (ArcGIS Urban همگام‌سازی سناریوها). این موضوع تضمین می‌کند که جزئیاتی مانند اطلاعات نورپردازی، سناریوهای مختلف و اطلاعات شکل برای بینندگان قابل مشاهده و تعاملی باقی می‌مانند، اگرچه ویژگی‌های ویرایش پشتیبانی نمی‌شود. Esri یک کانال ارتباطی مناسب از طریق نرم‌افزار سیتی اینجین برای میزبانی صحنه‌های وب در بستر آنلاین خود فراهم می‌کند.

فناوری‌های همه جانبه یک روند نو و محبوب در تجسم سه بعدی هستند. واقعیت مجازی (VR) محبوب‌ترین رسانه است: کاربران از یک هدست (مانند Oculus Rift یا HTC Vive) استفاده می‌کنند که حرکات سر را ردیابی می‌کند و تصاویر متفاوتی را به هر چشم نشان می‌دهد تا یک تجربهٔ سه بعدی واقع‌گرایانه و همه‌جانبه ایجاد کند.

شکل ۱۷. قسمت بالایی: پرداخت صحنه(رندر) بیدرنگ OpenGL CityEngine بدون انسداد محیطی یا سایه. قسمت وسطی: نمایی همراه با انسداد محیطی و سایه. قسمت پایینی: رندر Blender' Cycles ۱۲ دقیقه طول می‌کشد تا این تصویر را با قسمت‌های سایه روشن و شیشه انعکاسی ارائه دهد. شبکه با فرمت OBJ به Blender صادر شد.

توسعه یافته‌ترین کانال ارتباطی سیتی اینجین از موتور Unreal Engine استفاده می‌کند.

CityEngine 2019.0 شامل صادرکنندهٔ مدل Unreal Engine بتا است که خروجی آن را می‌توان از طریق جعبه ابزار Datasmith به Unreal وارد کرد.

جزئیات فنی به صورت آنلاین مستند شده است و احتمالاً در آینده نزدیک تغییر خواهد کرد (Esri 2019d). تجربهٔ استفاده از CityEngine VR یک میز شامل مدل‌ها را ارائه می‌دهد (شکل ۱۸). این مدل‌های صادر شده را روی یک میز در یک دفتر مجازی ارائه می‌دهد. کاربران می‌توانند با کشیدن مدل روی میز، مدل‌ها را کاوش کنند. به صورت اختیاری، کاربر می‌تواند به سایت‌های از پیش تعیین‌شده در دنیای سه‌بعدی تله‌پورت (انتقال اطلاعات در قالب کپی) کند تا نمای سطح خیابان از مدل را دریافت کند. این تصمیمات در ارتباط با طراحی از برخی معایب که ناشی از حرکت کاربران از طریق VR با سرعت بالا جلوگیری می‌کند. رابط رومیزی با اجازه دادن به کاربران برای ایستادن در بالای صحنه و کاوش آن از یک مکان «تقریباً ثابت»، بیماری حرکت را از بین می‌برد.

VR به عنوان یک قالب برای ارائه نقاط ضعفی دارد. اقلیت مردم هنوز بیماری حرکت یا راحت نبودن را تجربه می‌کنند، هدست‌ها برای استفادهٔ طولانی مدت مناسب نیستند و در مقایسه با مانیتورهای رومیزی همچنان وضوح پایینی دارند. با این حال، برای کاربردهایی که تأثیر فوری یا همه‌جانبه بسیار مهم است، می‌توانند ابزار بسیار قدرتمندی برای داشتن یک بحث مهیج و بررسی تاثیر اندازه‌گیری باشند.

شکل ۱۸. واقعیت مجازی سیتی اینجین یک مدل رومیزی برای هدایت با استفاده از کنترل کننده‌ها (سمت راست) ارائه می‌دهد. چندین کاربر پشتیبانی می‌شوند (هدست کاربر دوم در مرکز بالا نشان داده شده است)

نتیجه‌گیری

سیتی اینجین، چندین بخش از قابلیت‌های منحصر به فرد را به جعبه ابزار طراحان شهری ارائه می‌دهد. توانایی کار با قوانین، به جای مدل‌های دستی، می‌تواند به شدت در زمان صرفه‌جویی کند، مقیاس‌ها را به یک مقدار ثابت افزایش دهد و منجر به تعداد زیادی گردش کار جدید برای طراحی فضاهای شهری شود. این گردش‌های کاری جدید به ما امکان می‌دهند راه‌حل‌ها را در موقعیت «دفتر مشتری» به سرعت تکرار کنیم. راه‌حل‌ها را می‌توان در لحظه تجسم و تحلیل کمی کرد. چنین نوآوری‌هایی امکان بازخورد سریع‌تر کاربر و همچنین درک بهتر فضاهای مشکل و راه‌حل را فراهم می‌کند.

همهٔ گردش‌های کاری جدید با هشدارهایی همراه هستند و سیتی اینجین نیز از این قاعده مستثنی نیست. هنگامی که یک غیر برنامه‌نویس (که قوانین نمی‌نویسد) از سیتی اینجین استفاده می‌کند، با انتخاب محدودی از فایل‌های قوانین روبرو می‌شود. یک برنامه‌نویس معمولاً باید زمان قابل توجهی را برای یادگیری CGA و ایجاد فایل‌های قانون متناسب با مشکل صرف کند. با این حال، منابع قابل توجهی برای کمک به هر دو گروه از کاربران در دسترس

هستند: مجموعهٔ بزرگی از قوانین به صورت آنلاین در دسترس هستند و اسناد جامع API برای برنامه‌نویس ارائه شده است. سیتی اینجین در اصل از کار آکادمیک پاسکال مولر در ETH Zürich رشد کرد (Müller 2010). توسعهٔ مستمر محصول نرم افزار سیتی اینجین بی سر و صدا تحت‌الشعاع آثار آکادمیک که جزئیات نوآوری‌های آینده در سیستم را نشان می‌دهد (Schwarz and Müller 2015) بوده است. چنین فناوری‌ها و چنین ویژگی‌هایی اغلب بین سایر محصولات Esri و خود سیتی اینجین جریان دارد. نوآوری‌های اخیر در ارائهٔ داده‌های داشبورد و کانال ارتباطی برای واقعیت‌های مجازی منعکس‌کننده توسعه و پیشرفت هیجان‌انگیز مداوم سیستم در مرکز تحقیق و توسعه Esri زوریخ است.

منابع

Blender (2019) https://www.blender.org/. Accessed 30 July 2019

Brooks FP (1995) The mythical man-month: essays on software engineering. Anniversary Edition, 2/E. Pearson Education India

Esri (2019a) https://doc.arcgis.com/en/cityengine/latest/tutorials/introduction-to-the-cityenginetutorials.htm. Accessed 30 July 2019

Esri (2019b) https://doc.arcgis.com/en/cityengine/. Accessed 30 July 2019

Esri (2019c) http://www.arcgis.com/home/search.html. Accessed 30 July 2019

Esri (2019d) https://community.esri.com/docs/DOC-11563-cityengine-vr-experience-for-unrealstudio. Accessed 30 July 2019

git (2019) https://git-scm.com/. Accessed 30 July 2019

Houdini (2019) https://www.sidefx.com/products/houdini/. Accessed 30 July 2019

Kelly,T,WonkaP,MüllerP(2015)Interactivedimensioningofparametricmodels.ComputGraphics Forum 34(2):117–129.

Lipp M, Wonka P, Müller P (2014) PushPull++. ACM Trans Graphics (TOG) 33(4):130

Lipp M, Specht M, Lau C, Wonka P, Müller P (2019) Local editing of procedural models. Comput Graphics Forum 38(2):13–25

Maya (2019) https://www.autodesk.co.uk/products/maya/overview. Accessed 30 July 2019 Müller P (2010) Procedural modeling of buildings. Thesis, University of Zürich. https://doi.org/10.

3929/ethz-a-006397747

Müller P, Wonka P, Haegler S, Ulmer A, Van Gool L (2006) Procedural modeling of buildings. ACM Trans Graph 25(3):614–623

Parish YI, Müller P (2001) Procedural modeling of cities. In: Proceedings of the 28th annual conference on computer graphics and interactive techniques. ACM, pp 301–308

Prusinkiewicz P (1986) Graphical applications of L-systems. In: Proceedings of graphics interface and vision interface ’86, pp 247–253

Prusinkiewicz P, Lindenmayer A (2012) The algorithmic beauty of plants. Springer Science & Business Media

Rhino (2019) https://www.rhino3d.com/. Accessed 30 July 2019

Schwarz M, Müller P (2015) Advanced procedural modeling of architecture. ACM Trans Graph 34(4):107

SketchUp (2019) https://www.sketchup.com/. Accessed 30 July 2019

SpeedTree (2019) https://store.speedtree.com/. Accessed 30 July 2019

URDF (2019) http://wiki.ros.org/urdf. Accessed 30 July 2019

Vanegas CA, Kelly T, Weber B, Halatsch J, Aliaga DG, Müller P (2012) Procedural generation of parcels in urban modeling. Comput Graphics Forum 31(2pt3):681–690

Wikipedia (2019) https://en.wikipedia.org/wiki/Grome. Accessed 30 July 2019