Procedural System Assisted Authoring of Open-World Content for Marvel's Spider-Man

David Santiago
dsantiago@insomniacgames.com
Insomniac Games, Inc.
Burbank, CA

ABSTRACT

Crimes and vignettes are placed throughout the game city space using procedural systems to find appropriate locations. Editing of roads or buildings necessitates re-authoring the placement of dynamic encounters in that area of the environment. In many cases the dynamic encounters may have already undergone "final" manual adjustment to the space. To accommodate iterations on city layout, we add three main improvements to our procedural systems: preserve hand-authored work if it continues to meet specifications; place encounter components with higher fidelity; and provide the artists and designers guidance for crimes and vignettes needing more attention.

CCS CONCEPTS

• **Mathematics of computing** → *Mathematical analysis*; • **Computing methodologies**;

KEYWORDS

video games, procedural systems, asset instancing, authoring algorithm

ACM Reference Format:
David Santiago. 2019. Procedural System Assisted Authoring of Open-World Content for Marvel's Spider-Man. In *Proceedings of SIGGRAPH '19 Talks.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3306307.3328158

1 INTRODUCTION

Procedural systems are used to author at least the first pass of the open world in Marvel's Spider-Man video game on the PS4. Curves representing the street and alleys are used to generate the streets, curbs, sidewalks, alleys, plazas and building spaces. Those streets are then procedurally populated with pedestrians, vehicle traffic and props (mailboxes, streetlights, hydrants, no parking signs, etc.). The buildings are blocked in by designers using primitives, and then procedurally modularized (building kit components instanced on surfaces) as a starting point for artist iteration and polish. Once an area's plan and elevations are defined, the open-word dynamic encounters (approximately 3000 vignettes and 3000 crimes) can be procedurally placed based on desired densities and metrics.

SIGGRAPH '19 Talks, July 28 - August 01, 2019, Los Angeles, CA, USA

ACM ISBN 978-1-4503-6317-4/19/07…$15.00
https://doi.org/10.1145/3306307.3328158

2 DYNAMIC ENCOUNTERS

Dynamic encounters are modular scenes including over 30 types of vignettes and over 29 types of crimes that can appear at various times and locations during the game. Vignettes are non-gameplay environmental scenes such as car accidents and arrests. Crimes are interactive gameplay encounters such as chases, jewelry heists, drug deals and rooftop break-ins. Each dynamic encounter has specifications detailing how it should be placed in the world. A combination of metrics, desirability, preferred locations, density of encounter, proximity to other encounters, barrier and obstacle constraints, and other gameplay factors must be considered when placing each vignette or crime. As each encounter is placed there are fewer and fewer desirable or possible locations to place the next encounters. Some crimes and/or vignettes may occupy the same space as they are not simultaneously active. However, maximizing the variety of locations for each type of encounter is important.

Figure 1: An Arrest Vignette

Figure 1 shows a type of arrest vignette. The simple placement rules put them at least 150m apart on alternate sides of the street and greater than 9m from a crosswalk. Police cars should face in the direction of traffic in the left traffic lane or right parking lane depending on side of street. The vignette itself is contained in a prefab file comprised of all the necessary components in default positions that might be satisfactorily functional in-game solely with appropriate placement. The placement rules create the distribution in Chinatown seen in Figure 2.

Each Encounter Placement Module in the procedural system comprises operators to apply the parameters for placement. Inputs to the module include the ground geometry, sparse data sets that define the roads, alleys and sidewalks, building roofs and walls, exclusion volumes, and the locations occupied by previously placed encounters. The module first computes all the locations where the

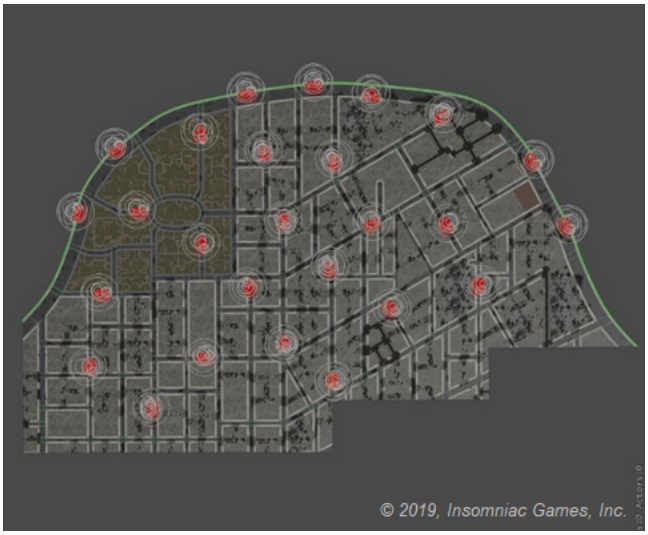

Figure 2: Arrest Vignette locations in game's Chinatown

placement metrics are satisfied. The many metrics may include distance from curb, lane or intersection, available clear area, and orientation with respect to a feature of the location. Next the occupied and excluded locations are applied as filters, and the possible locations are further reduced using ground and building data. Finally, the locations are culled to their desired density or count. Final adjustment or tuning of the vignette's assets to fit the environment is done manually in the editor as in Figure 3.

Figure 3: Arrest Vignette final placement

3 IMPROVEMENTS TO THE SYSTEM

The first improvement is creation of a tagging mechanism by which designers can indicate vignettes, crimes and elements that have undergone adjustment. Untuned crimes and vignettes may be replaced by the system when reprocessing an area. Tuned crimes may remain if all their critical features are still accommodated by the space. If not, the encounter can either be removed automatically if the location is no longer viable, or flagged for manual inspection if adjustment of the assets might remedy the issues.

A few types of guides are generated for the items flagged for manual inspection. One is debug camera coordinates to locate the issue in game or in the editor. Another is color keyed volumes to indicate the different issues to be inspected. Additionally, in-editor camera-facing annotation text placed at the 3D location explains the issue to be resolved.

Another improvement to the procedural system for placing or replacing encounters is additional artificial intelligence (AI) to help place assets used in combat. For example, there are many types of rooftop crimes that require 16m x 16m combat spaces. The original system finds rooftops with those available spaces and places and orients the crime with the rooftop edges. The new system processes the environment further to distinguish building and rooftop features to distribute the breakables and throwables throughout the combat space. This is preliminarily done by algorithmic placement to avoid the bounding volumes of previously placed elements while keeping a desirable distribution around the crime's non-player characters (NPCs). When higher fidelity is required, rays are cast from the roughly placed asset towards low resolution geometry to determine a more accurate position and orientation such as against a wall, dumpster, doorway, etc.

Similarly, procedural navigation clue placement for NPCs is improved to find traversable paths from rooftop to rooftop or rooftop to ground. The volume avoidance algorithm places the path endpoints in satisfactory positions. Raytracing finds a clear path from one endpoint to another through an intermediate waypoint. The system tries to generate at least one path to each neighboring rooftop or sidewalk. This allows crimes of the same type to be linked together enabling engaged enemies to chase Spider-Man around the city.

Iterations of modifications to the environment can easily create situations where dynamic encounter assets are no longer in valid positions (such as penetrating a wall or floating in space). Application of the new techniques in a validation scan of the open world can adjust, flag or remove invalid assets or gameplay markup.

4 CONCLUSION

These enhancements reduce the need to manually adjust the encounters during the continuous creative iterations of production. Without them, the variety of crimes distributed around Marvel's Manhattan in the final product would be greatly reduced. Systems in development rely on increased fidelity of the input data and additional recursions of the algorithms to create more accurate and varied procedurally authored content.

ACKNOWLEDGEMENTS

The author wishes to thank Marvel Games, Sony Interactive Entertainment and Insomniac Games for their support.

REFERENCES

David Santiago. 2015. GDC 2015: Procedural and Automation Techniques for Design and Production of Sunset Overdrive. https://www.gdcvault.com/play/1022217/Procedural-and-Automation-Techniques-for.

David Santiago. 2019. GDC 2019: Procedurally Crafting Manhattan for Marvel's Spider-Man. https://www.youtube.com/watch?v=4aw9uyj9MAE.