Blockage: A Global Game Jam Reflection

John Nesky, Ashley Zeldin, and Gabriel Gaete | Mar. 28th, 2012

In Blockage, the player navigates a snake through an isometric world from a level's entrance to its exit, leaving behind a husk that carries consequences for the next level.

Created at the fourth annual 48-hour Global Game Jam, Blockage was a crash course in rapid prototyping and level design for Gabriel "Gabo" Gaete, John Nesky and Ashley Zeldin.

* * *

Seven hours had gone by already, and with 41 hours left, our idea had devolved into a point-and-click locate slideshow.

Ashley intervened, citing the lack of mechanic and the focus on funny, not fun. "We're not thinking of the gameplay, and especially not the player."

So we went back to the Wacom tablet.

Back at theRobertZemeckisCenterfor the Digital Arts at theUniversityofSouthern California, Gabo pitched a play on the idea of the jam's theme, the Ouroboros icon of a serpent eating its tail.

The new plan made use of the 3D isometric graphics engine John had implemented in Flash the night before—along with our initial concept of cyclical gameplay in which past actions affect future outcomes—putting a puzzle-oriented spin on the classic snake game.

* * *

The snake emerges in a room where all surfaces are painted with a grid, and the player's object is to direct the snake to the exit on the other side.

The snake is a series of discrete block segments, growing one segment at a time in the direction the player indicates. The snake is able to climb a stair one grid unit high, but cannot scale walls taller than one unit.

As in the classic snake game, if the snake crosses its own path—eating itself—the level restarts. Taking the Ouroboros theme one step further, the snake's body persists after the level is completed, filling space as a husk in the following level. Players must navigate around the husk, adapting to the consequences of their past actions.

* * *

For such a puzzle game, we required a way to design a variety of solvable levels, so John's next priority was developing a level editor in which we could rapidly create new levels and save, then reload and edit after play testing. He based the level editor on the same code as the game itself, so team members could preview levels the way they would look in the game. Both the editor and game could load external files on the local computer so that other members of the team would be able to test their ideas without needing to recompile the whole game, allowing them to work independently.

Once the level editor was completed, Gabo set to work on the levels.

As a first-time level designer, courageously diving in with little guidance, his first instinct was to imagine the craziest puzzles possible. The distinguishing aspect of gameplay is how solving each level depends on the player's path in the previous level, which in turn depends on the level before that, so the most challenging puzzles likely would involve planning multiple levels ahead *Inception*-style.

While interesting, this approach to level design proved to be premature. We had been talking about the details of the game and the levels for hours and were comfortable playing the game, but between the novel husk (snake skin) mechanic, unusual play controls and the precognition required to progress through the levels, new players were overwhelmed.

We did not want to compromise the husk mechanic, so we revisited the level design and controls instead.

Gabo's first attempt at remaking the levels was unsuccessful. Though testers understood the mechanic, they said that even the initial levels were too hard for a newcomer to navigate. Another revision met the same pitfalls, with even more challenging levels such that one mistake could alter the course of successive levels, forcing frustrated players to backtrack several levels to correct it and seeing their success through the intermediate levels go to waste. Players not only didn't want to have to confront failure from the get-go, but also didn't want their hard work to be all for naught. A player tossed the keyboard and walked away after playing another of Gabo's mods.

The simple answer was to design the levels to have just one solution. But Ashley pointed out that players want—and deserve—freedom, and insisted on, if not multiple options to solve most levels, a measure of flexibility to explore the level and deviate from the most direct solution.

John, a more experienced game designer, stepped in to help design initial levels that introduced the husk and made reaching the goal without failure or too much reversal possible.

* * *

Striking a balance between freedom and constraint was key to the game play. Giving the player too much freedom to forge a trail would leave a messy husk for

the player to negotiate in the following level. Conversely, forcing the player to take only one path would take all the fun out of the game.

Therefore, we decided the first level would simply be a flat open space, allowing the player freedom to experiment with and master the directional controls before moving onto a level featuring the core mechanic.

The second level, meanwhile, would have a raised floor, allowing the player to see the husk from the previous level while safely lifting the snake above it.

The third level would give the player the opportunity to interact with the husk, but leaves an easy path to the exit.

Not until the fourth level would the player need to use the husk to reach the exit. The exit of the fourth level is right above the exit of the third level, which means that the husk from the third level would form a bridge that the player could always take to the fourth level's exit, safely guiding the player to the exit while reinforcing the significance of the husk mechanic in the game.

The following levels would begin to test the player's understanding of the concepts already presented.

* * *

Our friend Jack Bogdan suggested juxtaposing a preview of the next level on the screen to help players anticipate future challenges. However, we cannot expect players to be able to watch two levels at once; our design of the introductory levels should allow players to focus only on the current level. Players should not have to start planning ahead until they are comfortable with the core mechanic.

Yet he had a point.

To communicate the idea that the player's actions have an effect on the next level—and how—we decided to include a scaled preview on the right side of the screen communicating how the husk would change the paths available to solve the next level.

Post-game jam, we added a similar thumbnail of the past level to the left side of the screen for the sake of symmetry. A positive consequence of this visually balancing feature is that, if a player is having trouble solving a particular level, it's easy to look back and figure out how to change the solution to the previous level in such a way that makes solving the current level possible before returning to that previous level. Merely keeping the future level in mind when solving the current level may not be enough; as more hints available to the player, the deeper the level design can be.

* * *

Considering the time constraint, we made the decision to embrace a minimalist aesthetic, opting to spend time creating levels rather than art from a content perspective.

However, the simple aesthetic made it difficult to distinguish features in the level. Looking for a quick solution, Gabo had the bright idea to add lighting effects to dynamically fill in the details that our flat shaded art assets lacked. The first step was to apply ambient occlusion, resulting in a gradient highlighting raised platforms and shadowing lowered pits. Later, we implemented sharp shadows directly underneath raised platforms to imply space beneath.

Further complicating matters, the husk from the previous level would often accidentally break line-of-sight with important features in the 3D environment such as the exit. We were able to mitigate this problem by making the husk semitransparent; however, at first when the semitransparent husk crossed over itself, it had a cumulative effect, becoming more opaque. So John coded a

stipulation that the husk, even when it crossed over itself, would remain at a constant opacity.

In a way, the player would be contributing to the level design, and it was our job to encourage the player's contributions to be readable.

We wanted to add more variety to distinguish the levels, so we decided to paint them different colors. Riffing off the blocky style of the game, Gabo noted that a cube seen from the isometric angle looks like a hexagon made of six triangles. He chose this image to represent the game world, assigning a color to each of the triangles. The levels are colored according to this spectrum, conveying that the player is revolving around the hexagon with each completed level and making each level more memorable.

* * *

The isometric orientation posed a problem for implementing a control scheme, and highlighted the differences in people's perceptions of space. After entertaining the idea of mouse-click navigation, we settled on keyboard control. However, we discovered that the arrow keys did not have a universally intuitive mapping to directions in the isometric world.

To Gabo and John the "up-arrow" maps to the northeast whereas to Ashley it maps to the northwest; so it was unintuitive for her to use the configuration that worked for them, and vice versa. Though she offered to defer to the majority, John wouldn't settle for that. He figured there would be other players for whom that configuration would be unplayable—which proved true—and that would run counter to the point she had made of wanting the game to be as accessible to as many players as possible. Going with one scheme would disenfranchise anyone who saw the environment like the other.

Jack suggested using the rectangular configuration WQAS so that the diagonal corners of the rectangle clearly mapped to the diagonal directions of the isometric environment, an effective solution to the perception issue. With inclusivity in mind, Jack's compromise to use the WQAS keys sounded like a good idea, allowing all of us to play without feeling disoriented.

However, some play testers found even the WQAS control scheme unintuitive. Despite the direction match of the WQAS controls, Ashley and testers found themselves rotating the keyboard in order to play. After the game jam ended, we revisited the keyboard controls, determined to satisfy everyone. First, we implemented a menu to let players choose between the rectangular arrangement and the two interpretations of the arrow key arrangement. Then we investigated international keyboard configurations, finding the REDF keys arranged more consistently than WQAS, and neatly fitting under the dominant middle and index fingers when the hands are on the home row.

* * *

Designing Blockage was a reality check that taught us the nuances of level design, as well as the value of communicating mechanics clearly to the player and of teamwork in design.

After devising the initial idea, Gabo needed to think smaller-scale and deconstruct the grand idea into the parts comprising the whole. John's assistance implementing Gabo's ideas in a step-by-step progression improved the accessibility of the level designs.

Given our time constraints, we were not able to polish every aspect of the game to perfection. For example, we had planned to make the goal of navigating to the exit a little more obvious by highlighting the exit, but we only had time to carve a square hole through the wall to represent the exit. When a player asks "What do I

have to do?" the goal isn't clear enough. The exit should be more tempting to the player, so this will likely be the next thing we fix.

Another idea never implemented, was the idea of putting a key in each level that the player must collect in order to pass through the exit door. A common game trope, keys give level designers an excuse to direct the player in more meandering paths than they would otherwise take. When the goal is simply to get from point A to point B, players will likely take a direct route, but the key adds a third point, creating more opportunity for players to retrace and possibly trip over their steps. With keys, we would be able to design more challenging levels.

Finally, after all our effort using lighting to convey the shape of the 3D environment, the locked camera angle of the isometric 2D graphics engine still limited what we could get away with. If we had used truly 3D graphics, we would be able to use the extra cues of perspective and parallax to make spacial relationships even more obvious, and it would be easier to add large obstacles if we could simply move the camera around them to let the player see where to go. However, in the time we had left to finish the game, the 2D graphics and lighting proved to be sufficient.

Perhaps we will continue to modify Blockage going forward. For now, it stands as a testament to the power of collaboration under tight deadlines.

Thanks to Raghav Bashyal, Gordon Bellamy, Jack Bogdan, Alex Kerr and Colin Windmuller.

Ashley Zeldin (@snidelyhazel) is an independent game designer and producer inLos Angeles. A project manager and community manager with the IGDA, she coordinates the IGDA Scholars program and organises more than 300 volunteers worldwide. She's still a freelance writer on the side, as she has to earn money somehow.

John Nesky (@shaktool) is a game developer in Los Angeles. While at thatgamecompany, he was Feel Engineer on Journey. An active member of the online independent game community, his most recent tinkering can seen at beepbox.co, a tool for creating and sharing chiptune compositions, which Ashley used to create the catchy music for Blockage.

Gabriel "Gabo" Gaete (@gabotron2000) is a visual designer inLos Angeles. He utilizes animation, graphic design, 3D design, illustration and doodling to assist in the visual communication of products, concepts, visions and ideas for companies and for himself. He often can be found using his Micron pens to scribble on his pad of graph paper, honing his visual note-taking skills.

---

http://newsletter.igda.org/2012/03/28/blockage-a-global-game-jam-reflection/