

DEVELOPMENT

Can you tell us a little about Reflexive's "one-day-quick" prototype model?

(Zach) The goal of "one-day-quick" prototypes is to evaluate game mechanics and determine whether or not the game holds its own when assembled with only the bare essential components.

We usually spend about 2-3 hours discussing ideas that people have and when one sparks interest we start brainstorming aspects of that idea. It really is a free-for-all. It is very easy to get carried away, however, some of the best ideas come from having no boundaries. We keep a list of the games and ideas on a big white board in the conference room and when we have all emptied our brains out, the programmers decide what games they think have good potential for getting done in a quick and dirty manner.

Then it is off to work! The programmers start getting the game to a playable point, just to get something in the game.

Are these "one-day" projects built on top of existing game engines?

(Simon) We have several application frameworks setup for use as a quick prototype starting points:

The first is based upon the OpenGL utility toolkit; it allows us to quickly put together prototypes with wire frame or solid 3D graphics using basic geometric primitives.

The second framework is based upon Reflexives core foundation libraries, providing the programmer with common data structures, a unified data handling property system, keyboard and mouse input handling, simple access to auto-switching frame buffers, sound and music interfaces, plus timing functionality.

The third framework is built upon the second, adding built in object and level editing features based around a set of basic object types we call 'props', plus a visual scripting system. A prototype built using this framework tends to be less programmer centric, allowing a designer to build objects and scenes at the same time as a programmer is developing whatever custom components are required by the preliminary design document.

The third framework may look like an obvious choice for most prototypes, but this is often not the case. We try to think as out-of-the-box as possible during brainstorming sessions, and we may then pick the more outlandish ideas for quick prototype development to see where they will lead us. While the 'prop' based framework is very powerful, it still imposes a few restrictions, these are sometimes undesirable when you want the creative process to be as free flowing and unrestricted by normal development methodologies as possible.

WIK AND THE FABLE OF SOULS

IF EXCLUSIVE INTERVIEW | CONCEPT ART BY JEFF McTEER

Gish, Wik and the Fable Souls Win Grand Prize in 7th Annual Independent Games Festival

Thursday March 10, 9:00 am ET
SAN FRANCISCO, 2005

-- Gish and Wik and the Fable Souls have been selected as the top titles of the Independent Games Festival, the Sundance Festival of the video game world. Gish and 'Wik' were awarded in the open and web downloadable categories, respectively, at a ceremony at the Game Developers Conference in San Francisco.

Chronic Logic for Gish and Reflexive Entertainment for Wik and

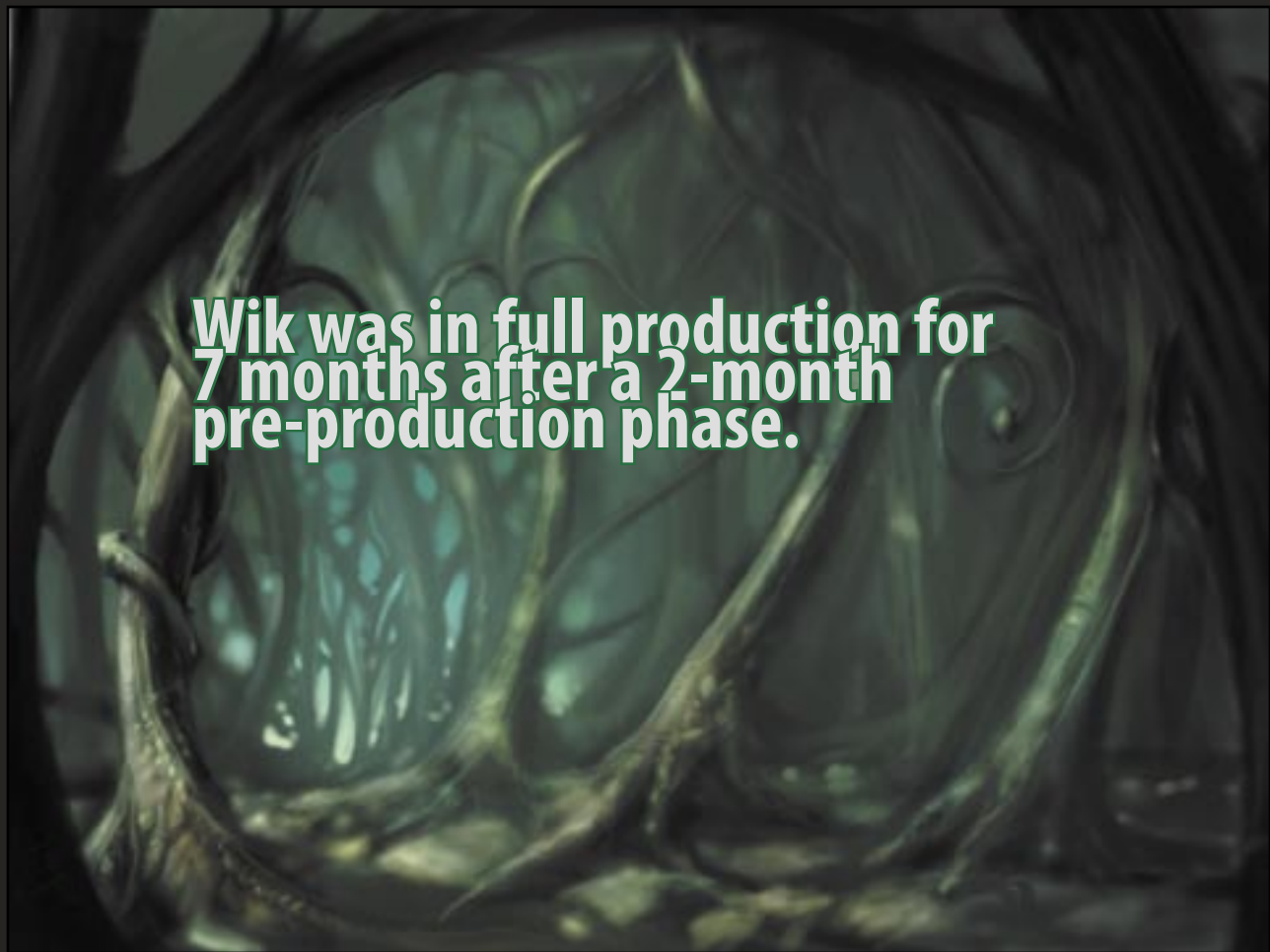
the Fable Souls were on hand to receive the \$15,000 Seumas McNally Grand Prize. Game development teams competed for more than \$40,000 in cash prizes.

"The IGF is an outlet for independent developers daring to take creative risks and push innovative thoughts," explains Simon Carless, chairman, IGF. "This year's winners are a group of professionals who stepped out of the

ordinary and ventured into a whimsical and ingenious space."

Independent game developers, similar to independent filmmakers, usually work with shoestring budgets. With the growing community of independent game developers emerging over the past decade, the IGF has taken center stage at recognizing these creators.





Wik was in full production for 7 months after a 2-month pre-production phase.



Wik is a forest dweller; a solitary creature whose only real friend is a faithful old mule named Slotham.

What type of pre-production time do you invest in a title that is considered during your “one-day-quick” model?

(Zach) Pre-production for the game mechanic of a prototype usually takes a few hours of brainstorming. Pre-production for the prototype art is usually about a half hour of the artists getting a general theme. The trick to “one-day-quick” models is that we know they are not going to ship that day.

If the prototypes go into full production, the pre-production time grows to a month or more. Over that period of time, mechanics may change and the art may take a totally different direction than initially planned. When a game goes into full production, new factors and concerns arise. The reality of making games kicks in and for better or worse can deviate from the original prototype. Questions like: Is this game marketable? Can we create the amount of professional assets required in

a reasonable amount of time? Can we make the game accessible enough to new users? All of these questions need answers before the game concept is considered final.

Even then however, exploring small tangents during development often leads to new ideas and can radically change the overall outcome of the project. The idea is to never settle with an idea if it is not as great as we think it could be. Leaving the creative doors a little open while continuing to stay on schedule provides a great environment for creative ideas to take hold.

What is the development cycle like- i.e., is it one-day/concept art/game engine/debugging/iterations? What type of time is invested into a speculative title like Wik? Is that determined up front? Is it always “develop a game in 6 months”?

(Simon) Once a quick-prototype is selected for full development, the project is assigned a producer, who, in the case of Reflexives smaller projects, also fills the role of lead programmer. We flesh out the preliminary concept treatment, creating a design document that outlines the game-play as envisioned at that point, it includes any story that may be necessary, target system specification, and lists the art, sound, and music assets that will be required in the final product.

The producer then works closely with the art lead on the project to define an art style, this in turn drives selection of appropriate in game technologies and asset creation tools required to realize that style in the game. The lead programmer then builds the foundation for the new project from scratch again, using one of the framework applications described earlier. We do not try to re-use the code

developed in our quick-prototypes; this is key, since they are generally coded without any concern for maintainability or optimized for performance.

Once the foundation for the project is in place, which usually means that the core play mechanic is working, and there is a level editing system in place, the project enters its full production phase. During this time artists, musicians, and level designers create and hook up assets, fleshing the product out. Wik was in full production for 7 months after a 2-month pre-production phase.

Once the majority of the game assets are in place, most of the team moves into full production on another project, while a smaller crew of mostly programmers is left to finish up. Post release there are often minor patches or fixes which need to be released, and

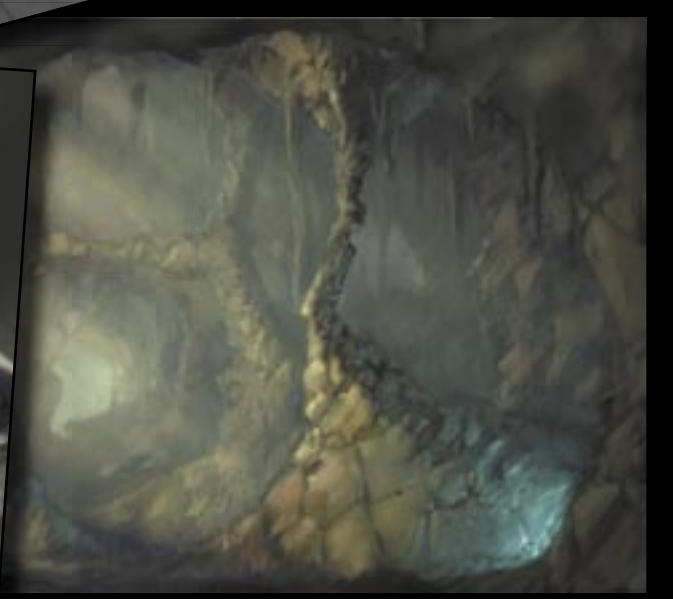
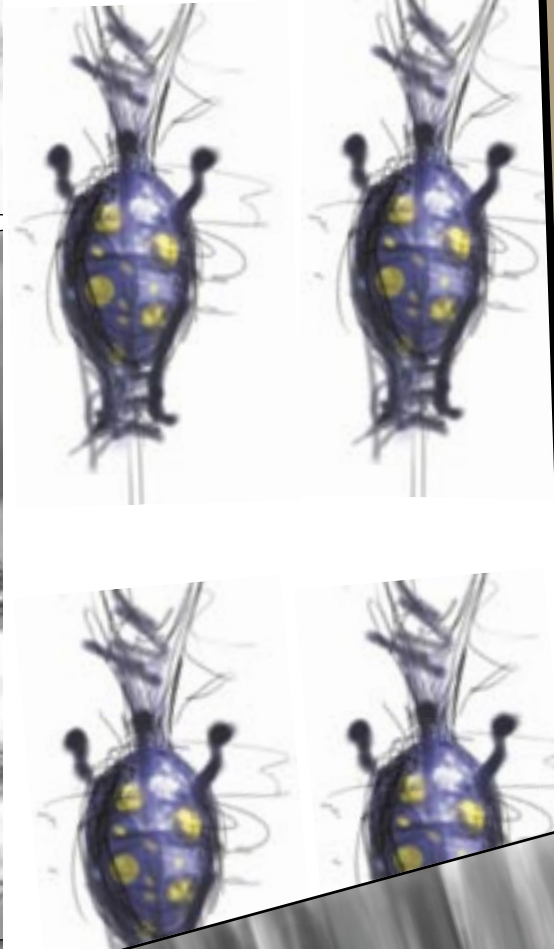
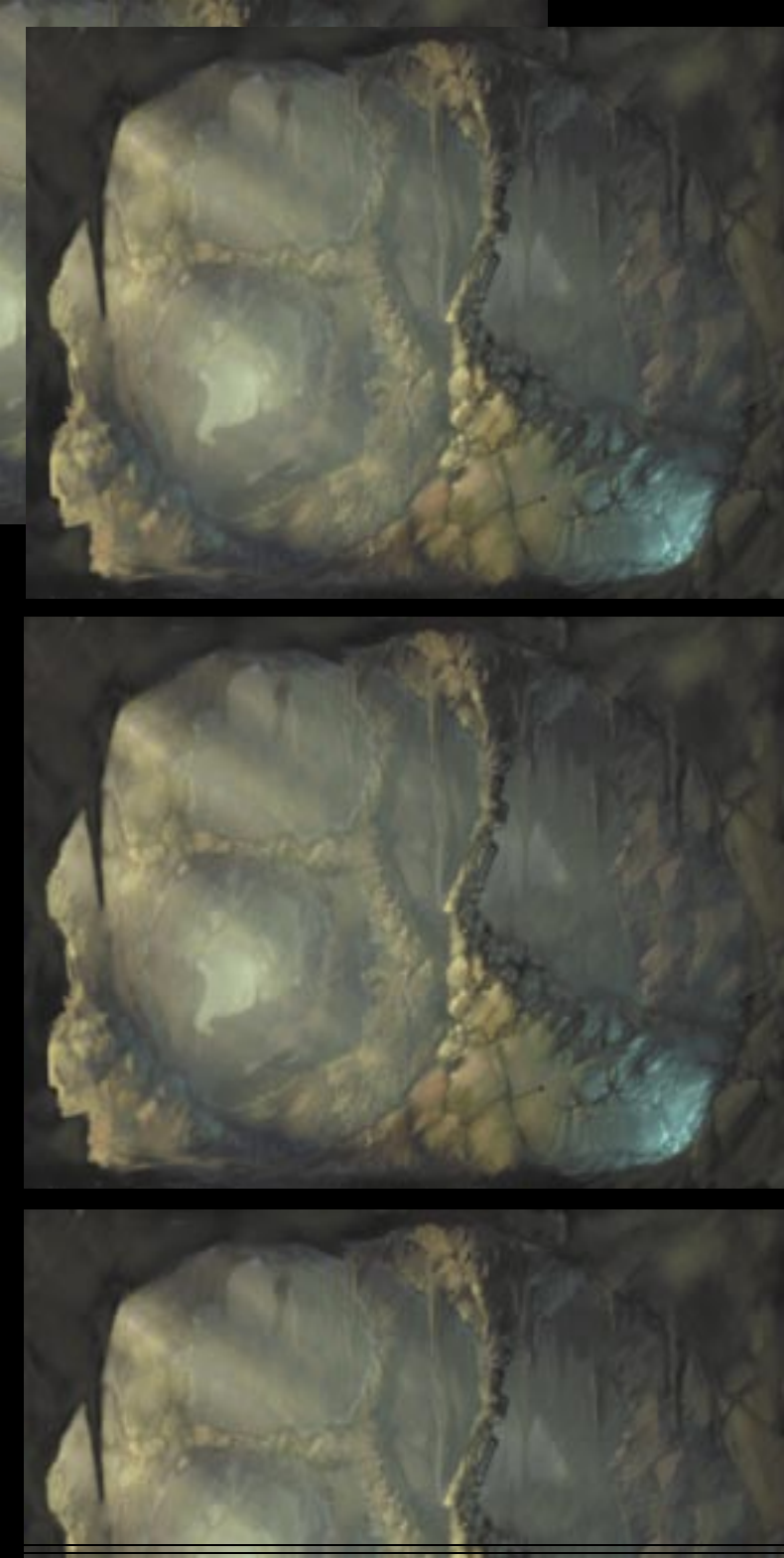
specialized builds that need to be created for the various portals we have helping us distribute our games, these often use their own purchasing systems and have custom branding needs, so a small skeleton team is often busy on a project for up to 2 months after its initial release.

Tell me a little about the back-story for Wik.

(Simon) Wik is a forest dweller; a solitary creature whose only real friend is a faithful old mule named Slotham. Many years ago Wik had a happy and contented life, living with his wife and children in a quaint tree house. All that changed one awful day, but Wik’s mind protects him from remembering the full horror of those events, only allowing access to brief snippets of the terrible things that happened...

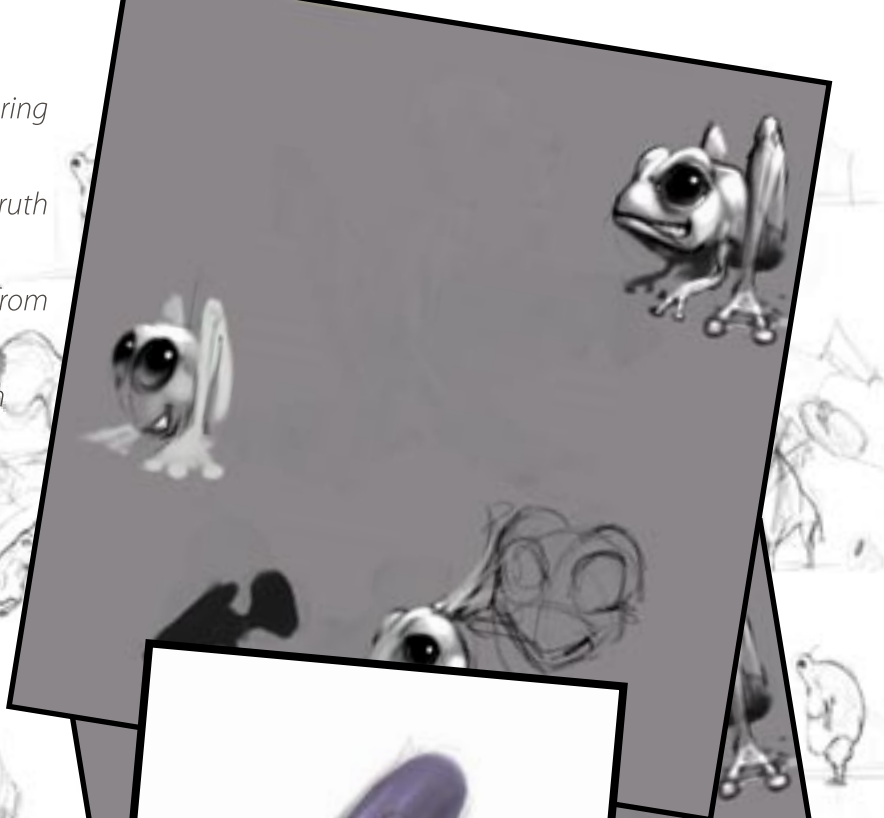
The in-game story opens the day Wik awoke to find that thousands of cute little grubs have appeared throughout the land he calls home. Most of the other creatures in the forest appear to have fallen under some kind of spell, charmed by the grubs in a way that makes them want to take a grub home and nurture it as though it was its own kin.

Wik immediately distrusts the overly cute grubs, and during the course of the in-game story he learns the awful truth that they are actually parasites that suck the life force from the creatures they trick into taking care of them, and then transform into horrific nightmare creatures. It was these same monstrously transformed grubs that happened upon Wik’s family years ago as they searched for the land they originally came from, destroying everything in their path.



Exploring small tangents during development often leads to new ideas and can radically change the overall outcome of the project. The idea is to never settle with an idea if it is not as great as we think it could be. Leaving the creative doors a little open while continuing to stay on schedule provides a great environment for creative ideas to take hold.
-Zach

Wik immediately distrusts the overly cute grubs, and during over the course of the in-game story he learns the awful truth that they are actually parasites that suck the life force from the creatures they trick into taking care of them, and then transform into horrific nightmare creatures.



PRODUCTION

Was there a level-editing system designed specifically for Wik?

(Zach) The level-editing system for Wik was specifically designed around the needs that arose from creating the levels and to a larger extent the entire game. All of the interface screens, levels, particle systems and buttons were built and scripted using the level-editing system. Instead of having the main menu hard coded, an artist can literally check out the level file for the main menu and move buttons, add art images even change where the buttons take you.

We also needed the ability to construct levels without pre-rendering each of the backgrounds out. For this we required a level-editing system that allowed us to use layers to create each level. Instead of having 50+ background images, we had three per environment type with pieces of art that can be placed and arranged on each level to create the illusion that the entire level was pre-rendered.

Within each of the levels we also needed to identify the areas that Wik could land on as opposed to latch his tongue to. This was done using different types of 2D polygon objects that the level designers could place and use to differentiate between tongue latch points, and solid collision areas.

Another very important part of the Wik level-editing system was the particle effects. All of the particle effects for Wik use small grayscale images and a data driven property system that allows for complete control over the number of particles created, what image they use, their alpha transparency, rotation, scale, color and how long they live. Having this system in game allowed the artists to custom tailor effects and test them in game extremely easily.

The end result was a level-editing system that was perfect for in-house use. It met and exceeded our needs in many ways. The only downfall is on a schedule so tight, it is not very intuitive. This made releasing the editor as a fully supported feature impossible. It is however possible to enable the editor and use it to create your own custom levels, see www.WikGame.com for more details.

The story element in Wik was a nice touch. Does Reflexive have other titles in development that contain a story element?

(Simon) Right now Reflexive is focused on several projects that we feel confident will be commercially more successfully than Wik was initially. Projects like Wik & The Fable Of Souls represent a significant development cost and we need to be sure we are ready to absorb that cost before we embark on another "mini-epic" J

At what point did the tongue as a form of locomotion come to be an element of game play?

(Zach) The short answer is about four months into production. This is a great example of the "leaving the creative doors a little open" we were discussing earlier. Had we gone with the original prototype version of bug eater you would have simply jumped around the screen gobbling up bugs in your path. While this was an interesting play mechanic, it was not nearly as dynamic as Wik's infamous

tongue latching action. Originally, the tongue was just used to grab bugs and objects around the edge of the screen. When Wik began to use his tongue to swing, things got interesting. I remember thinking as soon as I saw him eat something with his tongue, "if he could latch onto things, this is a game". Moving Wik around the screen by swinging and latching with his tongue quickly became the spotlight game mechanic.

Can you tell us about how/when the idea of applying gravity to the Wik character came about.

(Simon) The application of gravity was an idea that programmer Brian Fisher had come up with originally. It was relatively quick and easy to experiment with gravity in a rudimentary form, but the full collision and physics model that was created for the final product took longer to create.

About four months into development there were serious concerns that we had not yet found that magical spark of something in the game play that made the game feel great. During a brainstorming session where we were trying to find the elusive element that would make the game feel fun, we decided that we should not be afraid of changing the play mechanic radically and that we would devote some development time to exploring the gravity experiments Brian had started. It wasn't until the tongue-latch-swing mechanic was added that we knew for sure adding gravity was a winner.

What is the Wik Style sheet?

(Simon) The reflexive core game libraries implement a very powerful unified data management system that we call "the property system". This system allows programmers to quickly and easily create in-game editable properties that level designers use to tune the behavior of the props used to build the game playfield/scene.

Wik is a pretty extreme example of our capitalizing on the property system, we created something called the Wik style sheet, which is represented inside the game editor as a window filled with tabbed headings and 86 standard 'controls' that allow every aspect of the main characters behavior to be modified. If you want to change anything from the length of Wik's tongue, to the color tones applied to his skin, you'll find it can be modified using the Wik style sheet.

What does "if you're responsible for ensuring it's fun, you need to own the concept 100%" mean?

(Simon) Hehe! I knew saying that would come back to haunt me J As the projects producer, I was ultimately responsible for whether the product was 'fun' or not. I had inherited the quick-prototype "BugEater" that Wik was initially based on from another programmer. From the beginning of the project I had not particularly enjoyed playing BugEater in single player mode, I believed quite firmly that it had been chosen for full development based upon the strength of MouseParty™ mode, which was a total blast and great fun.

Rather than accept up front that the play mechanic in Wik would need to be very different from that in BugEater, I took it on faith that in time I would begin to see whatever it was that other people had found fun in the quick-prototype, and that we could capitalize on those same elements in Wik.

At that point I realized that in order to assure that Wik would be fun to play, I needed to stop trying to re-create a concept I did not truly believe in . . .

By the fourth month of development Wik still didn't seem like a whole lot of fun to play, and when you only have a budget for six-months of full production development time, that's not a place the producer wants to be in!

At that point I realized that in order to assure that Wik would be fun to play, I needed to stop trying to re-create a concept I did not truly believe in, and that I needed to take full ownership; making whatever changes I thought were necessary to make the game fun, no matter how different those ideas may be from the original prototype.

Having tutorials built into the game. Is this a standard reflexive practice?

(Zach) Definitely, a certain amount of tutorial is required for every game. Our belief is that if someone has never played a computer game before, we want him or her to be able play our game. However, some games require more training than others. With the Ricochet series, the amount of tutorial is minimal. The game explains itself. On the other hand, a game like Wik and the Fable of Souls needs a very thorough tutorial. The player must be introduced to so many "new" game mechanics that the challenge for us becomes how to make that tutorial efficient and fun.

I read that the average download game player wants to use a mouse-only interface? That must be frustrating.

(Zach) A complicated control scheme rarely makes a game more fun. In our experience, making a game "mouse-only" usually makes it more intuitive and enjoyable. Since you do not have to worry about fifteen keys and a joystick, you can actually play a game and drink your cup of coffee at the same time.

The restrictions on the exe. file size is another constraint in your end of the industry. How do you deal with that? Will Reflexive stay a downloadable game developer?

(Zach) We know going into production that the file size is one of the most important aspects of developing downloadable games. Sound and art are obviously the biggest size hogs. Even though our art is compressed like crazy we still cannot go nuts with the number of large images we have. An entire

800 X 600 background can be as much as 100k. This obviously adds up pretty quick, so the trick is to devise a method of creation that allows for the most bang for the buck. Wik and the Fable of Souls, the Ricochet series, and Big Kahuna Reef all have over 100 levels.

With that said, the art for the backgrounds alone would be over 10 megs and we still wouldn't have any characters, music, interface or sound effects. The key is creating reusable art assets that can be arranged by the level designers. Rendering out a few backgrounds with pieces that can be overlaid to create different levels with the same pieces of art. While this adds to the complexity of creating the assets, and requires more effort on the level designers side, it saves us an incredible amount of space in the final product.

Sound is the other big size eater. We use "Ogg-Vorbis" compression, which is an incredible sound compression tool. For example a one-minute song compresses down to 330k! All of our sound effects and music go through this process. Even though we get the songs down to 330k that doesn't mean we get to go crazy with tunes. We usually have between 3 – 4 music tracks for an entire game. Which makes squeezing them in extremely tricky. All of the songs have to loop and be able to be faded in and out at any given time.

As far as Reflexive Entertainment staying a downloadable game developer, the answer is yes. Having these size constraints pushes us to develop new ways of creating assets and game mechanics that really make every piece of art and sound in the game important. There is no room for fluff and that is the way we like it.

Have you ever ported a game to a wireless platform?

(Simon) We have people in house who have worked on a variety of console platforms, coin-operated arcade machines, and PC games for retail, one programmer developed a wireless dispatching system for ambulances, but if you mean cell phone games or handhelds that talk to one another via Bluetooth, the answer is no.



If you were to do a Wik II, what would you do differently? Would there be boss fights? [For the record, I hate Boss fights. I prefer additional story complexity and more challenging levels.]

(Simon) Yes, there would be boss fights, several were not only planned for the original game, art assets were already created for two of them, but there was not enough time to incorporate them in the final product.

In a sequel the story would continue right where it ends in the original, there are several directions it could take, but one in particular is already very appealing and would deepen the original experience rather than taking it in a completely new direction.

We would also add a MouseParty™ mode, having already experimented with several Wik's on screen simultaneously, each controlled with a separate mouse, we know how much fun this mode would be as either a cooperative or competitive addition to the game.

There has been some discussion about possibly using scrolling backgrounds in a future Wik game.

We have a number of improvements we would like to make to the in game tutorial, including the addition of a training system that would normally remain unobtrusive, offering specific advice only when it recognizes that a player is experiencing difficulty in any of a short list of problems we have noticed some players experiencing when learning the tongue-latch-swinging method of locomotion.

How much time and resources do you put into a game like Wik?

(Simon) Typically our games take between 2 and 9 months to develop, which includes the initial concept research, experiments (sometimes many) to nail the game play mechanics, and market research that we do both early on and towards the end of a products development cycle. Since each project has its own unique needs, it isn't really possible to express these amounts as a percentage of the total development time in a way that is either consistent or useful.



Wik is gaming genius. Was it a good business move?

(Simon) Thank you for the compliment. Sadly Wik has not yet recouped our significant investment in its development, so from a strictly financial perspective, no, it was not a good business move...

...But when you factor in the buzz that Wik has generated within the independent gaming community, the exposure that Reflexive has enjoyed within the game business as a whole since being selected as a finalist in the IGF, plus the pride and privilege that the individual artists, musicians, designers, writers and programmers who worked on Wik feel from being able to work for a company that is prepared to take this type of risk, I'd say it was a great business move. But then again, I'm afflicted with an extreme case of idealism :P

WIK CAN BE FOUND @ WWW.REFLEXIVE.COM

Game Credits:
 Producer & Lead Programmer:
 Simon Hallam
 Programmers: Lars Brubaker,
 Brian Fisher, James C Smith

Art Director: Jeff McAteer
 Art: Chung Ho Kan, Zach Young

Lead Level Design: Ion Hardie
 Level Design: Ernie Ramirez
 Additional Level Design: Brian Fisher,
 Simon Hallam, Zach Young

Music: Zach Young
 Sound Effects: Ion Hardie

Story: Eric Dallaire, Simon Hallam,
 Ion Hardie, Ernie Ramirez

ToolKit::

Art Programs:
 3DStudio Max
 ZBrush
 Photoshop

OTHER:
 Image Magic
 Visual Studio 6
 Slick Edit 9
 Visual Assist
 Cool Edit Pro

We also used the following
 open source libraries:
 Ogg/Vorbis, Freetype
 and Anti-Grain Geometry



THE POWER TO DESTROY TOWER PCs

Shuttle XPC systems deliver all the performance and features of traditional tower PC's in a fraction of the space. Available with the latest processor and video card technology, the Shuttle XPC shatters the myth that "bigger is better".



Shuttle recommends Microsoft® Windows® XP

FREE

Graphics Upgrade to
 NVIDIA® GeForce™ 6800GT 256MB
 (\$200 value)

starting at
\$1749*



P 2500g

Specifications	
Model	Shuttle XPC P 2500g
Processor	AMD Athlon™ 64 3500+ HyperTransport 939-pin
Operating System	Microsoft® Windows® XP Home Edition
Main memory	512MB Kingston® HyperX PC3200 DDR400 Dual Channel
Graphics	NVIDIA® GeForce™ 6800GT 256MB PCI Express x16
Hard drive	200GB 7200RPM Serial ATA
Optical drive	52X CDRW/16X DVD Combo drive
Media Reader	8-in-1 Built-in Card Reader
System Cooling	Integrated Cooling Engine (ICE) Liquid Cooling
Dimensions (L x W x H)	12.6 x 8.26 x 8.66 (in); 320 x 210 x 220 (mm)

* monitor not included

1 888 XPC 1818 www.us.shuttle.com find xpc at > BestBuy | Sam's Club | CompUSA | Walmart.com

