GAMASUTRA
The Art & Business of Making Games

PRINT

Postmortem: *Indigo Prophecy*

By David Cage

[Published in edited form in the June/July issue of sister magazine[Game Developer](#) *with magazine-exclusive concept art, Gamasutra is proud to present the full, extended postmortem for Quantic Dream's fascinating* Indigo Prophecy/Fahrenheit*, as described by creator David Cage.]*

Introduction

Working on an original project is both the worst curse and the most exciting thing that can happen in this industry. It's a curse because you go through periods of terrifying doubt. For two years you abandon all notions of a private life and forgo many hours of sleep because you are permanently looking for the right path, being simultaneously haunted by the idea that you may be completely mistaken and may be taking enormous risks.

It's also a fascinating experience because an original project allows you to construct your own vision, to test your convictions on the ground, to experience an extraordinary professional and human adventure with a team, with all the alternating periods of torment and euphoria that this implies.

Indigo Prophecy was all of that. This adventure taught me an enormous amount of things by forcing me to think about my vision of the future of this medium and how to make it evolve, sometimes by remaining true to its still-young traditions and sometimes by breaking away from them.

To sum up such a rich and intense experience in just a few pages by dividing it between positive and negative is a challenge that is almost impossible to meet. If I nonetheless try to do so, it is first of all because I have a special affinity for the impossible and also because I imagine that others may be able to learn from this experience with its share of successes and errors.

What Went Right

1: The Story - Playing with Archetypes

The writing of the scenario itself required a considerable amount of work (the final game design ran to more than 2000 pages…) and it was very well received (except for the end of the story, a point I will soon get back to).

Creating Empathy

My main aim was to write a story that would be capable of "absorbing" the players into the experience, making sure they kept a firm grip on the controller until they knew what was going to happen. The story thus became the driving force of the experience, both the main motivation inciting the player to play and the reward for playing.

I deliberately chose simple and popular starting points: an American city today, an ordinary man who finds himself confronted with extraordinary events, a series of unexplained murders. It seemed to be essential for the story to be easily and immediately accessible, without any need for explanations.

Characterization was my second priority. Initially I set out to create strong empathy with the main character without the players even realizing it was happening.

Unused Cover Art for *Indigo Prophecy*

The idea of the hero being a murderer in spite of himself generates spontaneous sympathy in the player. The stress situation imposed from the very beginning immediately plunges the player into the story and simultaneously making him Lucas Kane's accomplice (I had to fight hard with the Marketing Department, convinced that the player would not feel any empathy for a murderer, which proved to be wrong as it turned out. The narration enabled us to get over this difficulty without any problem).

But I still had to convince the player to take an equally rapid interest in the other characters (apart from Lucas Kane, the player could also control the two other main protagonists, police officers Carla Valenti and Tyler Miles). My goal was obviously for the player to have no "favorite" character, but to be equally pleased each time he encountered them.

I chose to use classic archetypes in order to work on my characters: Carla is a tough young police officer but discreetly sexy, totally immersed in her work to compensate for the lack of any emotional life. Tyler Miles is a cool young black cop who is sincerely attached to Carla, full of humor but serious in his work.

Quantic Dream.
LUCAS : prop 1
22/06/2003.
fahrenheit.
WaB.

Concept Art for Lucas

Many other secondary characters followed the same rule, the gruff police captain, the still-in-love ex-girlfriend, etc. All these simple characterization elements enabled us to present the characters very quickly while giving the player the impression he has always known them.

The danger with archetypes is that you have to avoid caricature (I hope I succeeded), otherwise the story loses all credibility and the player loses all interest. Most of my work on characterization consisted in enriching the initial archetypes in order to give them some real depth. Lucas Kane has self-doubts and has a complex relationship with his brother, a priest. Carla suffers from claustrophobia and chronic anxiety. Torn between his strong sense of duty and his love for his girlfriend, Tyler is involved in a complex relationship.

I also presented the main characters in their personal lives to give an idea of their context and emotional relationships. This aspect of the writing enabled us to quickly give them an added dimension: the player could see them operating in the context of the story, but he could also discover their private and personal lives, which helped contribute to the illusion that they actually exist.

Thanks to these simple techniques the player quickly feels close enough to the characters to share their feelings on an emotional level.

Bending Stories

Among the writing techniques that I developed for *Indigo Prophecy*, Bending Stories definitely received the most attention from the press. The basic idea was to solve the classic difficulty of telling a truly interactive story without generating an excessively complex tree structure. I had to find a way to enable the player to make significant choices with consequences while still maintaining the quality and pace of the story, as well as feasibility in terms of production.

The idea of Bending Stories consists in considering the story as a sort of elastic band that the player is free to stretch depending on his actions. The story retains its structure but the player can modify its length and form and thus participate in the narration. In reality the story does not change diametrically from one game to the next, all that changes is the way it is told. However, the player can see parts of scenes and obtain different information depending on the particular path he follows.

Another important decision was the option to work with small, closed sets rather than trying to model all of Manhattan. This decision was no doubt the hardest of all to make. However, my choice was based on solid conclusions: above all else I wanted to offer the player a sustained paced, brisk action, short scenes and frequent changes of location and context. The last thing I wanted was to have the player wandering about for hours in sets that had actions every hundred meters.

I therefore decided to opt for smaller, more closed sets but with a higher action ratio per square meter. As it so happened, this decision worked out well.

2: Concepts and Game Design

The Director, the Only Project Head

People often oppose creation and production, creators being supposed to have great ideas but no consideration for production constraints, producers wanting to be on time and on budget but having no interest in quality.

This opposition becomes particularly palpable in an original project, where many unknowns derive from the nature of the concept.

I chose to structure the project around two simple ideas:

- GAME DESIGN RULES! (GDR) (Yes, I love silly acronyms…): it is the experience that dictates what is going to be developed. It is a reasonable and rational dictatorship of the game design in terms of technical choices and artistic direction, based on the initial budget and time constraints.

- THE DIRECTOR IS THE ONLY MASTER ON BOARD AFTER GOD! (DITOMOBAG).
He is the guarantee of the global consistency of the vision. He makes sure all the elements of the game contribute to creating the same emotion. His choices can/must be subjective. They may be debatable when taken individually but it is imperative that they form part of a global vision that is clear and consistent.

The purpose of this organization was to have an "auteur" approach to game creation i.e. to create a context that gives one person full power to express his/her vision. This specifically made it possible to adopt strong-willed stances without constantly seeking compromises at all costs, which would have been disastrous for a project that claims to be innovative.

For me it is fundamental to have the director embodying the vision of the project: it is extremely rare for a truly original game to be developed without the creative vision of one person (from *Shadow of the Colossus* by Fumito Ueda to *Psychonauts* by Tim Schafer or *Killer 7* by Gouichi Suda, or also *Metal Gear Solid* by Hideo Kojima).

For me this is going to be a strong trend in the industry in the years to come, making it necessary to question a certain number of preconceived ideas.

Reconciling Narration and Interaction

One of the key points in *Indigo Prophecy* was the idea of getting interactivity and narration to work together. Most games oppose these two concepts or rather, they develop them in turn: a cut scene to advance the narration, then an action scene, then another cut scene for the narration. The structure of this narrative process is very close to that of porn movies.

The greater part of my work consisted in reconciling these two, first of all by eliminating the dichotomy from the design. More often than not a game is designed by a game designer who establishes the game play mechanics. A screenwriter is then called in to find a story to make the link between the levels.

More than anything else I wanted to break with this logic by designing the story and the interactivity simultaneously. My aim was to allow the player to "play" the story, to enable it to progress directly through player actions rather than jumping from cut scene to cut scene.

It was very difficult to find a solution to this problem, particularly because it demanded that each scene contain an interesting proposition in terms of the scenario and game play.

One scene in particular was a veritable revelation for me, the one where Tyler wakes up in his apartment at the start of the game. The player shares in Tyler's personal life in the morning before he goes to work: showering, getting dressed, drinking coffee, putting on some music, having a serious discussion with his wife, then kissing her before taking his coat and setting out for work.

When I wrote the scene on paper I spent whole nights in a cold sweat: what was the player going to play in the scene? Where were the mechanics, why should such a scene have even the slightest interest in terms of game play?

After months of soul searching I was very surprised when I finally saw the scene assembled, with dialog, animation, music and directing. And to my great surprise, the scene worked. It wasn't based on the traditional mechanics of video games (objective, obstacle, ramping, reward) but on something else that I still find hard to define.

I believe the scene is based entirely on the interest of sharing in the character's personal life, developing an attachment for him, becoming slowly immersed in his story.

In my opinion this is one of the most interesting scenes in*Indigo Prophecy*. No stunts, no artifice, just "being" a character in a simple context.

That scene finally convinced me that it is possible to create an interesting experience without weapons or cars.

Getting Rid of Patterns

The unnecessary character of game play mechanics was also a great discovery for me. Most video games are based on a series of patterns that the player has to acquire in a certain order in accordance with a certain timing and over different levels (e.g. press at the right time the right button to shoot, run, jump, move about, etc.).

The logic of patterns is totally opposed to the notion of narration, which evolves and is therefore non-cyclical by its very nature. It is therefore essential to abolish the rule of patterns in order to tell a story.

Of course *Indigo Prophecy* is not totally devoid of all patterns. The action sequences are a classic example of patterns and certain game play gimmicks also reappear regularly in the game. But generally speaking, the narration is based on a structure and a mode of interaction that is essentially contextual or, to be more precise, driven by the story.

In one scene Lucas has to hide evidence while in a split screen he can see a police officer knocking at his door. In another scene he explores the scene of the crime by alternating between Carla and Tyler. In yet another he will have to combine the actions of three characters in real time by switching from one to the other. What is the common denominator of all these mechanics? They serve the purposes of the narration, are not repeated at regular intervals and occur only once in the game.

Pacing/ Structure of Scenes

For me the pacing of the experience was an essential element in a game based on the narration. I had to avoid a slow and laborious narration at all costs and give priority to changes in pace, climaxes, and dynamics. I also had to rethink the adventure format in order to adapt it to consoles.

Two elements were essential in this respect. The first was to structure my scenes differently. I wanted short scenes that followed each other in quick succession so that the player would not have the time to get bored. Sets, characters and situations had to change quickly in order to kick-start the narration regularly, exactly like editing a film.

Each scene was structured as a mini film with a "hook" (to get the player immediately into the scene), two climaxes with increasing dramatic intensity and an epilogue.

We can see this structure in the very opening scene with Lucas in the Diner: the murder constitutes the hook, the climaxes are the waitress' call or the police officer standing up (depending on the player's actions), the scene concludes with Lucas' departure. The scene is a mini story in itself.

The player could thus turn on the console to play for ten minutes and have a satisfying experience, whereas most narrative games necessitate considerably more continuous investment before they provide satisfaction. This structure provided a sustained pace and very considerably contributed to the impression of fluidity that the game gives.

The other decisive element was the increased number of pace changes. Generally, I left relatively few scenes in free time. I did not want the player to wander about for hours in a set and thereby destroy the pacing of the narration (thus the quality of the experience). This is why there are many events in each scene to break the pace.

I tried to multiply the situations where the phone rings, someone enters or an unexpected event takes place, in order to set the pace, but also because they create the illusion that the other characters really exist and behave in their own right.

Interface

The interface was also the subject of intense reflection all through the project. My first intention was not to turn it into a remote control as often happens in adventure games, nor an exercise in skill but rather a tool for immersing the player physically in the world.

The idea of MPAR (Motion Physical Action Reaction, did I tell you about my silly acronyms?) developed quite naturally. It consists in offering the player the possibility of accomplishing the same movement with the right analog stick as the character on the screen. The system enabled us to unfold the animation progressively, providing a feeling something close to Inverse Kinetic but without the disadvantages. The system also had the enormous advantage of being easily contextual, enabling us to use the same interface to take an object, open a door, have a drink or play yoyo.

The principle of MPAR then governed all of my reasoning with regard to the interface. With the help of the controller I wanted the player to physically feel the same thing as the character. The Track&Field interface (derived from the eponymous game) applies the same principle: as soon as an action requires force or endurance, we ask him to tire himself out alternately on the triggers. The bodybuilding scene is a fairly good illustration of this principle, with the player suffering alongside the character in order to finish the series.

The conclusion of this work on the interface was that we could create real physical immersion thanks to the interface. More than just a control mode, it becomes the physical link between the player and the experience.

After a few minutes' play most players have completely forgotten the existence of the interface because it is simple and intrudes minimally on the screen, allowing them to concentrate solely on the story and the characters.

3: Iterations Under Control: Be Ready to Change the Plan

First Goal: Make the Plan

It is a nearly impossible challenge to preserve one's initial vision over two years of development and an incommensurable number of difficulties of all sorts. It is absolutely essential to protect what constitutes the essence of the project without being sidetracked by the innumerable unforeseeable factors of development.

I always have very detailed and complete game designs before I begin production. The way the game operates is described on paper in the smallest detail and overall the final result differs quite little from the initial document.

I tried to have a maximum number of certainties on paper before starting production, particularly so that asset development could start on healthy bases. I also needed them in order to be sure that I would not run up against a brick wall in the middle of the development.

For the aspects that caused me to have doubts or questions I worked on setting up a functional solution while allowing for some iterations in the course of development. They related very largely to playability aspects or the interface, for which it was difficult to make decisions without having progressed with the development.

This approach proved to be particularly effective for this type of project: it enabled me to have a solid base while still retaining sufficient flexibility to make discoveries in the course of development without perturbing the production.

Second Goal: Improve the Plan

These iterations took different forms during the development, ranging from methodological research to rewards for perseverance and including accidental discoveries.

MultiView and Mental Health are two of the very rare fundamental ideas of the game that came into being during the development. One morning a member of the team came in with one of the very first episodes of the TV series “24”. The first scene of one of the first episodes hit me straight away: we saw a character using a computer, with a window showing the screen, another showing the worried face of the character, a third showing a general view of the room. I was won over by the idea that we could show the interface and the emotion on the face of the person using it. I also quickly saw the full potential of such an interface for game play, particularly by showing several places at the same time while still leaving the player in control. MultiView was interesting both in terms of narrative and game play, making it a particularly interesting feature for a game like *Indigo Prophecy*.

Convincing the team to implement it was another story. I have to admit that when I got the idea, the PS2 engine was running at 5 frames/second with a single set. The idea of opening as many as four windows simultaneously seemed to be science fiction at the time, not to mention the loading and memory problems that would necessarily be involved.

At such a stage in development, it is a serious asset if you are both lead designer and CEO at the same time… I decided that MultiView was going to be one of the key features of the game and that it was essential to adapt the game design, the technology and the production in order to accommodate this, which the team accomplished at the cost of much work and an amazing feat of determination and faith in the future.

Mental Health is another example of a feature that appeared in the course of development. My initial stance was that it would not be necessary to integrate life or morale gauges because the player's feeling would be enough. He would feel spontaneously encouraged or demoralized depending on his actions.
When the game was assembled it quickly became apparent to me that this would not be the case and that we would have to materialize the

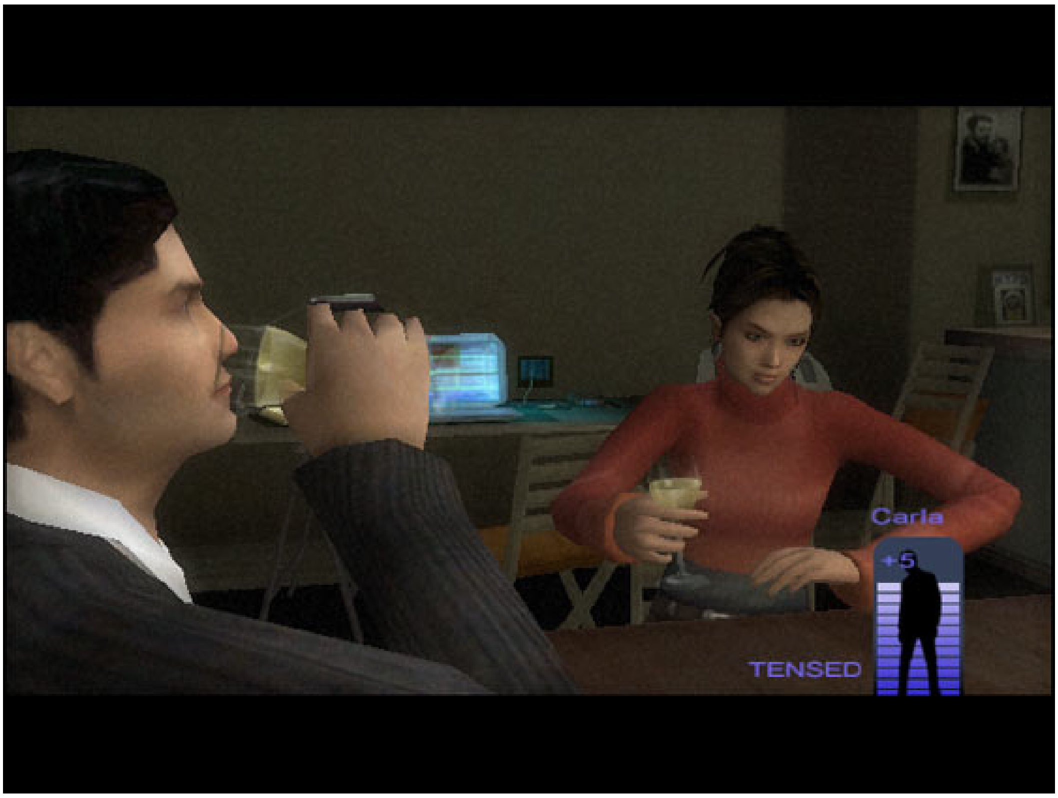
emotional state of the characters in some more concrete form for the player. The addition of this little gauge was finally quite important for the experience of the game by making the psychological state a simple game play mechanic that was perfectly consistent with the narrative aspect.
The decision to implement it was based on the fact that it had few consequences on the development and a significant impact on the experience of the game. The main interest was that all the actions were already present in the initial game design. All we had to do was give them an economy and consequence with Game Over sequences.

The implementation was particularly well scheduled and organized and everything went very smoothly without any glitches.
From a Game Design point of view, this somewhat late implementation gives a slightly superficial result in places. I would no doubt have given it more importance and would have integrated it more profoundly in the game (particularly by giving it more repercussions in the rest of the game) if I had had the idea sooner. But in the end of the day, Mental Health constitutes an interesting mechanic that improved the experience of the game.

The Game as a Living Entity

To conclude, I consider the game during the development phase as a living entity. It must be given solid bases but we must also give it room to breathe and remain on the lookout for discoveries that can happen along the way. Exploring a fascinating pathway that opens up in the middle of development is always a risk that has to be evaluated impartially, particularly in terms of its repercussions for the production. It is always a question of a "vista", even daring, for the project manager who makes the decision, and he cannot afford to make any mistakes in terms of the consequences.

In the case of *Indigo Prophecy*, I believe this freedom to stay in tune with the project during its development played a fundamental role in determining the final result. It is also no doubt a luxury that not all productions can afford.



4: Vectors of Emotion

The cinematographic approach in *Indigo Prophecy* was an essential aspect of the game concept from the very beginning. The idea was to manage to recreate a richness and diversity of emotions comparable to film by using similar mechanisms (narration and characterization), but ones that are also peculiar to the medium (interactivity, immersion).

The goal was straight away to use all possible vectors of emotion. Beyond the story itself, the emotional vectors we identified were the virtual actors, the directing and the music.

Virtual Actors & Motion Capture

The work on Virtual Actors was absolutely fundamental to*Indigo Prophecy*. From a technical point of view, the creation of virtual actors capable of communicating emotion was a veritable challenge. Although the writing would facilitate the task, the characters still had to be technically capable of expressing these emotions.

Motion Capture was an immense aid in this respect. Quantic Dream has its own in-house MoCap studio, without which *Indigo Prophecy* would probably not have been possible. In addition to the usual technical difficulties linked to MoCap, the main problem I found myself confronted with was how to maintain coherent characterization for each character. For Lucas Kane, for example, several actors were required for the body animation (adventure, body dialogs, stunts), another actor lent his voice and the facial animation was then recorded by a puppeteer. It therefore took eight actors to give life to Lucas. And as their performances were staggered over more than 18 months, my work as a director was essential in order to preserve the coherence of the character throughout the game.

Facial Animation

Work on the face was also extremely complex. I had experimented with different techniques in my previous game OMIKRON-THE NOMAD SOUL, ranging from facial Motion Capture to triggering blendshapes with a midi piano. Given the quantity of dialogs and characters (more than 2h30 for more than 70 different characters) and the level of emotion we were aiming at (entirely mobile faces), we had no choice but to find an appropriate solution.

We finally opted for a system of "digital puppets". A glove is used to capture finger movements and each finger is assigned to a blendshape animation. With two gloves a professional puppeteer could thus dynamically control each blendshape dynamically and in real time, combining and blending them in a fluid and natural manner. He listened to each line twice, then recorded the animation while viewing the result in real time, lip synch and face simultaneously. The system also allows for retakes or to re-record only a part of the face.

Movie Maker Module

The quality of the directing was obviously a crucial aspect of the game. It was essential for us to provide quality comparable to that of a film. We studied different possibilities, cameras created in the modeling tool or by script, finally opting to develop an internal tool called “Movie Maker Module” or “M3”.

M3 is a real time workbench that is integrated into our scripting tool. It offers a system of tracks per type of data (cameras, animation, sound, post-render, FX) with a time line, just like an AVID or PREMIERE workbench. The result can be viewed instantly in the game engine.

The user can not only place cameras but he can also play with the animation or position sounds in order to have real control of the whole scene. The result is a binary file that can then be used by the script at any time.

The main goal was to create a WYSIWYG tool, the principal advantage being its extremely simple and intuitive interface and workflow. The idea was that it required no special technical training to use M3, so the tool was accessible for people having cinematographic training.

The Music

This subject has received ample attention in the past, but this post-mortem would not be complete without mentioning the music as a vector of emotion in *Indigo Prophecy*. I must admit that it took several unfortunate tries with talented musicians before I found what I was looking for. What I wanted was real film score, a combination of sensitivity and nuance, something that would lend an added dimension to my story.

The work of Angelo Badalamenti, David Lynch's composer since Blue Velvet (he also composed the theme music for Twin Peaks, amongst others), fulfilled my dreams. I asked him to work as if for a real movie, forgetting the fact that it was a video game. We gave him a script, visuals, a very detailed description of the scenario bible and the characters and I spent a lot of time with him communicating my vision of the story and the feelings I wanted to get across.

We worked by theme for each character, these being played out in accordance with the emotions of the different scenes.

The quality of the emotion Angelo brought to *Indigo Prophecy* through his music really gave the game an added dimension. Some composers of film scores have experience and talent that are still rare in video games.

Many games think that the music must be based on Carmina Burana or John Williams in order to have its place in a video game. Personally, I am convinced that it can lend a unique quality of sensitivity and emotion that we are still struggling to express graphically.

5: Good Management of the Usually Underestimated Tasks

There is a large quantity of hidden tasks in all development. They seem negligible at the start but often prove to be very considerable at the end of production because they were badly anticipated.

These are few examples of minor tasks that we anticipated correctly:

Localization: It would take a whole book to describe the quantity of various and diverse problems that can arise in badly anticipated localization. After OMIKRON, which was a particularly heavy game in terms of localization, *Indigo Prophecy* set a new record with nearly three hours of audio dialogs and lip synch, in addition to a considerable quantity of written texts, all translated into six languages, including Japanese. The localization went smoothly, mainly thanks to a good anticipation of the tools (Excel import/export of texts from the dialog and text scripting tool).

Auto-Builders:On a cross-platform containing a considerable quantity of data and six languages, it rapidly became a priority to automatize the builds. All the game data was stored on a central server along with the history and versions for each platform and language. Before leaving at night, a tool was launched to automatically reconstruct the specified version by recuperating the latest validated version of all the game data from the server. In the morning the QA team had a new version of the game for testing.

Intranet: We made a significant investment in our Intranet server in order to be able to deal with a certain number of classic development problems. We specifically integrated the following sections directly accessible from an Internet navigator:

- Library: all documents shared by team members, with management of user rights
- Shared Scheduling and follow-up of task lists for the whole team
- Declaration and follow-up of requested vacations
- Bug Report: simple and adapted to our needs, it was used both for the project and for debugging tools.

Synchronized Release of Tools and Technology: When we develop our own technology internally with a considerable R&D team, it is of primordial importance to synchronize the company's multiple tools. If the set exporter supports a new graphical feature but the scripting tool does not yet support this feature, there is a strong chance that it will crash. It is therefore essential to find an adequate solution so that the code is constantly synchronous.

We implemented a very strict discipline while developing*Indigo Prophecy*. A machine was dedicated to compiling release codes (it compiled automatically as soon as the code was modified and mailed the relevant leads when the code no longer compiled). All tools were released on the same day on the same machine from the same code of the same branch. When the code is effective, all the tools are tested. When they have been validated by QA, they are made accessible to the users who can then update their tools (direct update from the tool).

6: Changing Publishers Can Be Good

Changing publisher in the course of development is usually the worst nightmare imaginable. *Indigo Prophecy* is a sort of a special case because the developer went to see the publisher to ask to terminate their agreement.

The situation is a major classic for all developers: highly motivated people who really believe in the vision and the product sign a game. These people leave or are fired, but the game stays with the publisher. The fired people are replaced, but everything the previous people signed is necessarily of no interest. The developer can very quickly find himself in a ridiculous situation where he has signed for a game with a publisher who no longer wants it.

Indigo Prophecy had reached an advanced stage of development but no one at the publisher's seemed to understand exactly what we were doing. If I were more cynical I might have rejoiced in the situation, comforting myself that it was too late in the project to kill it, but I was really convinced of the quality of the game and I was scared the game would be released "quietly", with no marketing or promotion. Like any developer who sacrifices two years of his life for something he believes in, I couldn't resign myself to the idea that my game would be released without the necessary marketing backup.

So I went to see the publisher and, convinced of the quality and originality of *Indigo Prophecy*, I asked them to give me back my game. They were intelligent enough to accept this proposition, which thus enabled us to find a new publisher without any difficulty, one who was happy to acquire a cross-platform project at the Beta stage and just a few months away from the end of development.

Changing publisher was the most positive element in the development of*Indigo Prophecy.* We suddenly found people who were really enthusiastic about our game, convinced of its quality and who wanted to defend it and talk about it. We started a real collaboration in order to improve the game and streamline the game play. *Indigo Prophecy* improved considerably during this period, particularly thanks to the new view provided by the publisher, who was very useful in terms of regulating the pacing of the game.

The conclusion of this experience is that it is sometimes beneficial to change partners when you feel that things have ceased to progress. The "big leap" is never easy to make, but it can prove to be salutary for a project.

What Went Wrong:

1: The Story - The Bad

Generally speaking, the scenario and characterization worked particularly well but a few glitches in the writing prevented the scenario from reaching the level of quality I was aiming for.

Here are some of them in no particular order:

- one bad guy per story is enough: the Oracle was the real enemy in*Indigo Prophecy* and I think his characterization was quite good. The AI that comes into play at the end of the game only adds confusion to the plot.

- you can make people believe anything in a scenario, but only once per story. In the case of *Indigo Prophecy*, the story of Lucas, guilty of a murder he never really committed because controlled by the Oracle, was THE unreasonable proposition in my scenario, which the players accepted without difficulty. The series of new developments at the end, although built into the first scene (the crow representing the AI is a leitmotif throughout the game) constituted a series of added propositions that went beyond what the players/spectators could reasonably accept.

- I made the mistake of not devoting enough time to the last hour of the game. I was convinced (and rightly so) that the first hour of the game would be decisive for hooking the player, but I naively thought that one hour from the end the player's opinion would be made. I therefore devoted most of my time to the rest of the game in order to make it as perfect as possible.

This was obviously a mistake. I was forgetting that what leaves a lasting impression on the player is often the end, and that a bad ending can change his perception of the whole game.

It won't happen again…

2: Graphics: "Atmospheric but not Stellar" (as they said…)

The graphical quality of *Indigo Prophecy* was given an uneven reception by the critics. I think there are three reasons for this:

- *Indigo Prophecy* was developed simultaneously on three platforms, PS2, Xbox and PC. PS2 had been defined as the lead version and no efforts were spared for this console. The last months of development were devoted to graphically improving the PC and Xbox versions, but too few specific features were developed for these platforms to make these versions competitive with the best games on the market.

- the second reason was a deliberate choice to work on the graphics in terms of atmosphere rather than in terms of a technical demo. In order to create a grim atmosphere by working on the colorimetry and the grain of the image we deliberately avoided easy lens flare and other such env map effects that invade most games. The result probably lacked eye candy.

- the last reason relates to a misevaluation of needs in terms of tools. The graphics team did top quality work but the tools that would have facilitated the graphical production appeared much too late in the production or were not suited to the job. The heaviness of the technology always has direct repercussions for the graphical quality of the game: when a graphic artist spends more time trying to view his work than improving it, the result is a loss of quality.
The burden of having to develop an internal technology on three platforms proved to be too much to afford the artists the comfort they needed.

On our upcoming productions we have devoted a significant effort to developing the graphics and animation tools entirely in pre-production. More specifically, we have greatly extended our WYSIWYG philosophy enabling direct visualization on console in all tools.

3: Action Sequences - Almost A Good Idea...

The action sequences are a good example of a good idea that never came to maturity during the development of the game unlike, say, MPAR and MultiView. Reviews were globally good for this part of the game, which was designed from the very beginning more as a spectacular breath of fresh air in the narration than as a veritable game play challenge. I am personally dissatisfied with the result.

The initial concept was to avoid having repetitive action sequences like most games, where the player accomplishes the same actions (shooting, fighting, driving).
The narrative structure imposed a great variety of actions, animation and situations in order to preserve the cinematographic side of the experience and there was no question of imposing shoot sequences every ten minutes at the risk of totally destroying the narration.

The other important point in the project specifications was that the camera should be free to provide top quality directing (so no views from behind). Finally, there was no question of providing specific interfaces for each new action scene. We therefore had to imagine a generic interface that was equally suited to a chase scene, a game of basketball or ice-skating.

The result was the PAR system (japanese designer experimented something similar with his Quick Time Events in SHENMUE).

The final idea of assigning controls to the analog sticks and bonding them to the movements of the character on the screen came quite late in the development, too late for the appropriate tools to be developed. The implementation was thus very largely blind, and the tuning particularly long and delicate.

In addition, we failed to find an ideal visual representation for the symbols on the screen. We tested a large quantity of positions, sizes, shapes and colors and finally opted for peripheral player vision. It was an interesting option but not entirely convincing, the interface being graphically too invasive. If the player does not use peripheral vision, the eye moves from the symbols to the scene and the interface masks the scene.

A better implementation would have been possible if we had had complete WYSIWYG tools earlier in development.

4: Insufficient Overall Vision for the Technology

The *Indigo Prophecy* technology is not really spectacular. It was primarily intended to serve the experience of the game and fluidity. Most of its features were designed to improve the game experience.

For example, the game is not burdened by any loading in the course of scenes thanks to an "intelligent" streaming system that enabled the game to offer a particularly large quantity of action per square meter.

The scripting tool we used to assemble the game enabled us to construct extremely complex scenes with a great number of possible variations and a great variety of mechanisms. The MultiView and Multi-Character scenes were veritable technical challenges.

In spite of many positive points, the game suffered globally from an insufficient overall vision for the technology. This placed a considerable burden on the development and demanded not inconsiderable extra efforts that the team could otherwise have avoided.

The first mistake occurred when changing platforms. Developing a PC game is very different from a console game, particularly in terms of memory management, loads and saves. We considerably underestimated the switch from PC to console and failed to identify the difficulties correctly (we quickly focus on the frame rate, whereas the memory and loading issues are considerably more problematic).

The second mistake was an insufficient analysis of the game design. It seemed to be very simple (playing animation in scripts with conditions) whereas it finally proved to require great underlying complexity. Synchronously managing several scripts playing simultaneously in several windows but capable of interacting with each other, as in the Hotel scene, is just one example of the type of cases that had to be managed.
We also underestimated the needs of a game that uses few recurrent mechanics.

The other classic error we committed was trying to develop generic tools with a view to possible future productions rather than tools dedicated to the experience of the game we wanted to create. The initial scripting tool was supposed to enable us to script both an FPS and a tennis game. The reality quickly proved to be different from the theory.

A generic tool enables management of a great variety of cases… but none of them very effectively. The prospect of reusing a tool as is for future productions is usually a pipe dream that costs time and money in the short term with no guarantee of profitability in the long term.

In *Indigo Prophecy* we realized this early enough to be able to correct the error. We adapted the tool, making it less generic but more effective for the type of game we were developing. With a scripting tool that was better suited to the game, we considerably simplified the scripting and accelerated the development.

The reality of development in the years ahead is that engines will no longer constitute a veritable barrier, and there is a strong chance that all teams will very quickly have access to the same features.

The real technological stakes of Next Gen development are in the tools.

5: Difficulties of Developing an Original Concept

The major difficulty in *Indigo Prophecy* was being able convince before, during and after the development, publishers, the team, the marketing, the journalists and finally the gamers.

Convincing a Publisher

The difficulties inherent in an original project begin even before it goes into development with the incredibly difficult task of convincing a publisher.
Indigo Prophecy was no exception to this rule.

I often say that a truly original project cannot be signed by a publisher except if based on a misunderstanding, because if the publisher really understood what he was signing, he would never sign it.

The initial presentation of *Indigo Prophecy* was capable of terrifying even the most foolhardy of publishers (and these are few on the ground…). The challenge of the project was to create an experience in which the player would control the main protagonists of a story in which his choices would modify the course of events. No gun, no car, no action, no puzzle…

The first publishers I spoke with took me at best for a harmless dreamer, at worst for a madman. The publisher's first reaction when evaluating a new project is to look at the sales figures for games in the same category. In my case, he either considered that the game was a new genre, therefore having no comparable reference, or else that it belonged in the category of adventure games, an economically minor genre. Clearly not a good omen for *Indigo Prophecy*…

The other difficulty I ran up against was trying to explain the experience that *Indigo Prophecy* was going to be. Even today I still find it difficult if not impossible to explain what the game is like to someone who hasn't played it. The dialog inevitably ends with the person looking at me with a puzzled expression accompanied by a polite smile.
Because it really is impossible to explain *Indigo Prophecy* (you can make the test yourself…).

In an action game you can explain how to shoot, what weapons are used, who the enemies are, and everyone immediately understands the type of experience it is going to be.

With *Indigo Prophecy*, no one can see right away how the player can enjoy playing with only a story and choices.

Explaining the concept of an original game with no real prior references is a major difficulty that must not be underestimated. I had to deploy considerable efforts before I finally managed to generate enough excitement to sign for the project, after spending more than a year discussing it with the publisher.

Convincing During Production

Developing an original project also constitutes a mass of difficulties at all levels of production. Apart from the initial difficulty of convincing the publisher to take the risk of trying something new, you then have to convince them that the project is making progress.

Lots of publishers need to be reassured by the game play early on in the development and demand vertical slices (a full playable game level complete with all the features). This method is perfectly suited to shooters or car games where we can in fact produce a complete level and have a good idea of the final game play because all that remains is to duplicate the game play by changing sets.

The immense difficulty of the *Indigo Prophecy* concept was that a large part of the fun came from involvement in the story and emotion, which were both difficult to illustrate from a single isolated scene.

Because *Indigo Prophecy* didn't really use repetitive mechanics, playing an isolated scene demonstrated nothing except how that scene worked, without any indication of the context nor of the feeling the game would finally produce.

Moreover, it was difficult to perceive the typology of a game based on emotion as long as everything had not been set up, animation, dialogs, facial, lighting, rendering and most of all music. It was a bit like watching film rushes and trying to imagine the final result.

To optimize production, I had chosen to structure the development horizontally rather than vertically and produce all the animation, all the sets, all the characters, while simultaneously beginning to assemble. The result took time before it became visible but when it did, the whole game appeared "as if by magic" from one day to the next.

Production models should be adapted to the specific nature of the project, rather than just copying the established recipes. All games should not be produced like action games. Some creativity also has to be put in new methodologies to develop these games, more in line with their true nature and in respect with the publisher's constraints.

Convincing the Team

Managing the development team constitutes the other difficulty involved in producing an original project. When working on a FPS, the team has a very clear idea of what it is doing from the very first day of development. It has outside references that enable it to judge the quality of the game.
With an original project, this does not apply. The team had to have enormous faith in order to be able to produce *Indigo Prophecy*. Some of the team members even admitted to me that they had not really understood what *Indigo Prophecy* was all about until after the game was released and they heard their friends talking about it.

Anyone who has already directed a project knows how essential it is to have the complete confidence of the development team. If the team has reservations or no longer believes in the project, that project will fail.

For *Indigo Prophecy*, I made a point of never showing my doubts (which were nevertheless numerous and sometimes difficult to overcome all through the development) and made special efforts to share my vision of the project with the team.

Here again, the special nature of *Indigo Prophecy* and the fact that it was difficult to understand the experience of the game before everything was fitted together, did not make life any easier for me, but the team stuck with me all the way and I was genuinely touched by their confidence. I would like to take advantage of this occasion to express my gratitude to them.

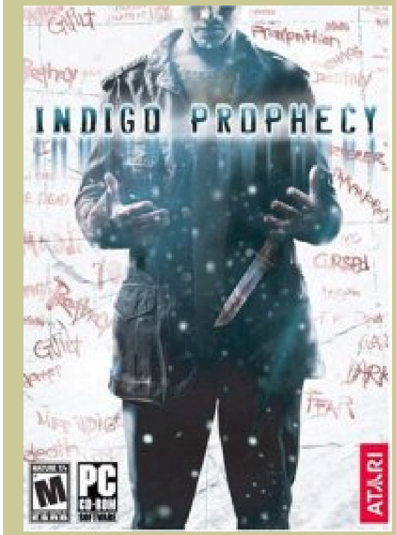
Conclusion

Indigo Prophecy was really an extraordinary experience both professionally and in human terms. I learned an enormous amount from it and it profoundly changed my vision of interactivity. I won't make video games the same way before and after *Indigo Prophecy*, and I think it also deeply changed the vision of most of the people in the team.

And although the game may not be perfect, I hope that the passion and enthusiasm that the team and I invested in it will nonetheless make it a game that is both different and sincere.

In my personal development it constitutes an important stage toward making video games not just simple toys but a veritable form of expression. I hope it has given other more talented people the desire to explore interactive narration and the formidable capacity of this new medium to create emotion.

Game Data



Publisher: Atari
Developer: Quantic Dream
Platforms: PlayStation 2, PC, Xbox

Number of full time developers: Around 80

Length of Development: Two Years

Release Date: September, 2005

Return to the full version of this article