

# Chapter 2

## Video Game Development and User Experience

Graham McAllister and Gareth R. White

**Abstract** In order to design new methodologies for evaluating the user experience of video games, it is imperative to initially understand two core issues. Firstly, how are video games developed at present, including components such as processes, timescales and staff roles, and secondly, how do studios design and evaluate the user experience.

This chapter will discuss the video game development process and the practices that studios currently use to achieve the best possible user experience. It will present four case studies from game developers Disney Interactive (Black Rock Studio), Relentless, Zoe Mode, and HandCircus, each detailing their game development process and also how this integrates with the user experience evaluation. The case studies focus on different game genres, platforms, and target user groups, ensuring that this chapter represents a balanced view of current practices in evaluating user experience during the game development process.

### 2.1 Introduction

In order to design new methodologies for evaluating the usability and user experience of video games, it is imperative to initially understand two core issues. Firstly, how are video games developed at present, including aspects such as processes and time scales, and secondly, how do studios design and evaluate the user experience?

This chapter will discuss the video game development processes and practices that studios currently use to achieve the best possible user experience. It will present four case studies from AAA game developers Disney Interactive (Black Rock Studio), Zoë Mode, Relentless and mobile developer HandCircus, all based in the

---

G. McAllister (✉)  
Player Research, Brighton, UK  
e-mail: graham@playerresearch.com

G. R. White  
University of Sussex, Brighton, UK

© Springer International Publishing Switzerland 2015  
R. Bernhaupt (ed.), *Game User Experience Evaluation*,  
Human-Computer Interaction Series, DOI 10.1007/978-3-319-15985-0\_2

UK. Each case study will detail their game development process and also how this integrates with the user experience evaluation. In an attempt to represent a balanced view of state-of-the-art in game development practices, the games studios chosen focus on different game genres and target user groups.

Reader's take-away:

- Four concrete case studies of how video games are developed at world-leading studios.
- A clear understanding of the game development life cycle.
- Understanding of industry terminology, laying the foundations for a common language of user experience.
- An understanding of industry needs, both in terms of what they expect and require from usability and user experience evaluations.

In summary, the key contribution that this chapter makes to the games usability community is an understanding of the game development process and how these studios currently involve the end user.

## 2.2 Previous Work

Although the topic of evaluating video game user experience is gaining more attention from both academia and industry, it is not a particularly new area. One of the earliest papers (Malone 1981), discusses which features of video games makes them captivating and enjoyable to play. Today, this discussion still continues, and there is active research in determining which game features to evaluate and which approaches should be used.

Current approaches to evaluating the usability and user experience of video games have centred around themes such as mapping established HCI methods to video games, refining these methods, identifying guidelines and perhaps most importantly, evaluating the overall player experience. A summary of relevant literature will be discussed below.

### 2.2.1 *Traditional HCI Approaches*

Due to the generic nature of the majority of usability methods, researchers have analysed how existing usability methods can be applied to video games (Jørgensen 2004). Others such as Cornett have employed usability methods such as observations, questionnaires, think aloud and task completion rate, to determine if they would be successful in identifying usability issues in MMORPGs (Cornett 2004). Without much, if any, modification, there is evidence to support the claim that conventional usability techniques can be successfully applied to video game evaluation.

### **2.2.2 Refining Traditional Methods**

Although established usability methods can be directly applied to games, Medlock and others at Microsoft Games Studios have developed a usability process which is specific to games (Medlock et al. 2002). Their approach is called the RITE (Rapid Iterative Testing and Evaluation) method and although it is very similar to a traditional usability study, there are two key differences. Firstly, the process is highly iterative, meaning that whenever a fault is found and a solution identified, it is immediately corrected and re-tested with the next participant. Secondly, the usability engineer and design team identify and classify each issue to determine if it can be resolved immediately, or if further data is needed before a solution can be found. This can be thought of as usability “triage”.

### **2.2.3 Heuristics**

Nielsen’s usability heuristics (Nielsen 2007) have long served as general guidelines for creating usable applications or websites. However, various researchers (Federoff 2002; Desurvire 2005; Schaffer 2008; Laitinen 2006) have constructed sets of heuristics which are specific to video games, and compared their effectiveness to Nielsen’s. They found that their heuristics are most useful during the early phases of game development. Despite the existence of specific game heuristics, questions remain about their specificity and utility (Schaffer 2008), and feedback from developers suggests that they are too generic to be of much use.

### **2.2.4 User Experience**

According to Clanton, the overall deciding factor of a good game is game play (Clanton 1998). Trying to specify what makes a good game is not a straightforward task, and Larsen has tried to unpack this problem by examining how professional game reviewers rate games (Larsen 2008).

Others have addressed the key criticism of heuristics by wrapping them up in a unified process (Sweetster 2005). This process, which they call GameFlow, can be used to design, evaluate and understand enjoyment in games. Meanwhile, Jennet has conducted a series of experiments to measure and define the immersion in video games (Jennet 2008). The concepts of immersion and flow in games are often related to involvement or enjoyment, but they appear to have subjective and imprecise definitions, thus making them difficult to design for and measure. Furthermore flow seems to be only applicable to describing competitive gaming, despite other research pointing to the diversity of emotions a player can experience during play (Lazzaro 2007), and indeed the diversity of players and games (Schuurman et al. 2008).

### **2.2.5 *Game Development***

Most research in the area measures games towards the end of the development life cycle. Although this may be suitable for fixing small changes on time, it is not sufficient for altering key game mechanics. If new techniques are to be designed which can evaluate a game during its development cycle as well as the final product, then a better understanding of the development life cycle needs to be obtained.

One reason for the lack of tailored techniques which could be applicable to all stages of game development is that the game development process itself is not known in detail to the HCI community. Federoff work shadowed a game development team for five days and has reported some details of the development process (Federoff 2002). However, the main focus of this research was to construct game heuristics, not report on the development process per se.

The next section will discuss the general characteristics of game development including the development life cycle and relevant industry terminology.

## **2.3 Introduction to the Game Development Life Cycle**

In the video game industry, it is nominally accepted that the development life cycle is constructed from the following phases, though in practice those occurring prior to production are often contracted or skipped entirely:

### **2.3.1 *Concept***

Game concepts can be initiated either from the publisher, who provides finance, or the development studio, who are responsible for the day-to-day production of the game. Once a general concept has been agreed between the two parties a small development team of perhaps 5 staff may spend 1–2 months producing an initial Game Design Document and visual representations such as still images or a movie to communicate the vision for the game. Additionally, a rough budget and plan is produced, including milestone agreements which define the production commitments of the developer, and the corresponding financial commitments of the publisher. This would normally represent a phased or iterative delivery of the product, where only a first-pass of each feature is completed prior to evaluation and feedback from the publisher. Later in the schedule a second delivery is made which is a more concrete implementation. Agreements made at this stage are still subject to adjustment at any future point.

### **2.3.2 Prototyping**

During the early stages of development many different aspects of the game may be prototyped simultaneously and independently. These provide examples of features such as menus, physics and vehicle handling, or could be technical demos such as grass rendering or other components of the game or graphics engine. In order to define a visual benchmark, the art team may construct virtual dioramas, which are models of events that players will experience during the game. Some of these prototypes could be interactive, others could be non-interactive movies demonstrating an example from which the interface for this part of the game could be developed.

This initial phase can take between 3–6 months, by the end of which these prototypes and concepts are evaluated, and if the project is given a green-light it moves into pre-production.

### **2.3.3 Pre-Production**

Following design approval, the game development team enters the important pre-production phase, during which time fundamental game mechanics are proven and problematic areas are identified. The purpose of this phase is to try out ideas quickly without getting bogged down in issues of final presentation quality, to identify risks and prove the important aspects of the game concept.

### **2.3.4 Production**

During the main production phase the team will be scaled up to full size and would tend to spend in the order of 12 months producing all of the characters, levels, front end menus and other components of the game. Often during this stage the team will produce a “vertical slice”, which is a high quality, 10–15 min demonstration of a small sample of the game.

In addition to the core team of programmers, artists, designers, and audio engineers, game developers also include a Quality Assurance (QA) group who are responsible for testing the game. This is essentially functional testing rather than usability or experiential testing. The QA team are keen gamers with a good understanding of the market and what to expect from a high quality game. As such, in addition to functional bugs which are entered into a database and addressed in a formal process, testers may also identify ‘playability’ issues which are informally discussed with the rest of the development team. Final issues of usability and playability are the responsibility of the producer and designers. QA teams are often only scaled up to full size toward the end of the production phase, through Alpha and Beta.

### **2.3.5 Alpha—Beta—Gold**

Toward the end of production the game progresses through a series of statuses which indicate how close development is to completion.

In order to achieve Alpha status all content in the game should be represented, but not necessarily be of final quality. Placeholder content is common, but the game should exist as a coherent whole.

From Alpha a further 6 months would typically be spent advancing through Beta status until the game is finally available for release, with attention turned to bug fixing and finalising the quality throughout.

By Beta all content and features should effectively be finished, with all but final polishing still to take place. Nothing further will be added to the game, only tweaking and final adjustments. In particular this phase is focussed on bug fixing. After Beta the developer and publisher consider the game to be of a shippable quality and submit a Master candidate disc to the format holder (i.e., Microsoft, Nintendo or Sony) for approval.

Each game that is released on any of their consoles has to first be approved by the format holder's own QA team. Strict technical and presentation standards define how all games on the platform should deal with issues of brand-recognition as well as certain HCI guidelines. For example, which controller buttons to use for navigating dialog boxes, the format and content of messages to display to the player while saving games, or where to position important interface elements on the television screen.

The approval process usually takes 2 weeks, but in the event that the submission candidate fails, a further submission will have to be made once the development team have resolved all of the faults. How long this takes depends on the severity of the issues, but once the team have successfully dealt with them another 5–10 days will be required for the re-submission approval process. Conceivably, further submissions could be required until all issues have been resolved.

Once approval has been given, the publisher uses the Gold Master disc to begin manufacturing and distribution, which takes between 1–4 weeks. Typically a unified release date is agreed upon with all retail outlets, which requires an additional week to ensure stock is delivered from the distributors to all stores in time for simultaneous release. In the UK this results in retail outlets releasing games on a Friday.

## **2.4 Case Studies**

This section presents four case studies from world-leading developers Black Rock Studio (part of Disney Interactive), Zoë Mode, Relentless and HandCircus. Each studio will discuss their development process, time scales and how they involve end-users.

*Case Study 1—Black Rock Studio* Black Rock Studio specialise in developing racing games for Xbox, PlayStation (PS) and PC. Their latest game, *Pure*, was released in September 2008 to critical acclaim. We interviewed Jason Avent, the Game Director of *Pure*, who attributes the high ratings to not only the talent of his team, but also the usability evaluations that were conducted during development.

### **2.4.1 Game Development at Black Rock Studio**

Game development at Black Rock Studio typically takes between 18–24 months, with a phase breakdown as follows:

- Prototyping (3–6 months)
- Pre-production (6 months+)
- Production (6–12 months)
- Alpha, Beta, Submission, release (4–9 months)
- Testing by the format owner (10 days)
- Disc manufacture (2–4 weeks)

The total development time for *Pure* was approximately 20 months, which was at the lower end of the range of each phase. Delivering the product while keeping to the lower end of the range was attributed to the team's experience and their agile development process. Each of these phases will now be explained in more detail.

### **2.4.2 Prototyping**

During the prototyping phase, the publisher's marketing team employed a recruitment agency who produced a detailed online questionnaire about the game concept and distributed it to approximately 100 people within the target demographic. Following the online survey, a focus group study was conducted with 3 groups of 4–5 participants to discuss the game concept. The study was run in a lounge environment, with two members of staff, one taking notes from behind a one-way mirror, and the other sitting with the participants to facilitate the discussion.

The team also decided to build a "Pre-Vis" (pre-visualisation) prototype for one of *Pure's* key features, player customised vehicles. This took the form of a non-interactive movie showing each part of the bike attaching to a central frame. This was not a technical demo, but rather just a visual benchmark or reference from which the interface for this part of the game could later be developed.

### **2.4.3 Pre-Production**

*Pure* had originally been intended to just be an incremental advance on the previous title in the series, but over the course of 7 months the game concept evolved

through several different designs. The initial idea was to make the most authentic quad bike racing game on the market, but this was abandoned in favour of a concept tentatively called *ATV Pure* which avoided realistic sports simulation in favour of informal, social, off-road racing. The final concept, called simply *Pure*, was only settled upon by the time the team were half-way through pre-production. Each of the preceding versions contributed some design features to the final game, but many were prioritised so low that they were never included in the released title. The design strategy for *Pure* was to focus on implementing a few core features to a very high standard, rather than attempt many features to a lesser standard.

By 12 months into the development cycle, the team had fixed their vision for the game and were only addressing issues that supported 4 key aspects: massive airborne tricks, customisable vehicles, 16 player online games, and so-called “FHMs” (which for the purposes of this paper we will describe as “Flying High” Moments). These FHMs represent the core experience of *Pure*: when the player drives round a corner at high speed, only to realise that what they thought was just a bump in the road turns out to be the edge of a cliff face, leaving them flying thousands of feet up in the air. This is not only a highly exciting part of the game, but also a key mechanic as it allows the player plenty of time to perform tricks in the air, which eventually results in a higher score and other rewards.

These concepts were graphically documented in a number of ways specific to Black Rock, but which have similar implementations in other studios. An important tool for summarising the concept of the game and keeping the team on track is the X-Movie, used by some of the world’s largest games publishers such as Electronic Arts. The X-Movie for *Pure* was a video showing a bike rider on an off-road race track jumping high into the air. Black Rock maintained the idea of “X marks the spot” when it came to their “Core Idea Sheets”. These were large display boards hung around the team’s development area showing the 4 key game aspects superimposed onto a bull’s-eye target. At the centre of the target was the FHM feature, as this was intended to represent the essence of the entire game. It is worth highlighting that this essentially puts the user experience (of excitement), as the single most important criteria for the game.

#### **2.4.4 *Alpha to Release***

For *Pure*, Black Rock did not employ a QA team throughout the entirety of the project, instead they only used 1 or 2 testers in the month leading up to Alpha. From Alpha, this was increased to 5 staff to deal with both the Xbox 360 and PS3 versions of the game (which were simultaneously developed). Furthermore, the QA team was only concerned with addressing functional testing rather than questions of usability or user experience.



### 2.4.5 *Post-Launch*

The publisher's marketing team conducted studies after the game had been released and sold most of its units. The purpose of these studies was to identify what consumers liked or did not like about the game, and what made them purchase it. Similar themes were discussed with consumers who did not purchase the game.

After release of the game, some informal analysis was conducted of official reviews and user comments on forums. Avent asserts that with the exception of some aspects of the game, few usability issues were mentioned in game reviews. Most user's comments related to features that were intentionally excluded by the team for practical reasons (such as the excessive technical overhead of including replays and split-screen multiplayer).

### 2.4.6 *Understanding the User*

*Pure* was the first title on which Black Rock employed usability tests. They began running tests with company staff, careful to choose people who were not on the development team. They then expanded to recruit other people who worked in the same building as them, but who were not part of their own company. Finally the most substantial tests began with members of the public, recruited directly from the streets and local universities. In total around 100 participants were involved over the course of 4 months, of which the final month was only concerned with quantitative analysis of the game's difficulty level and any issues that would prevent players from completing the game. The only requirements for recruitment were that participants were 14–24 years old, male, and owned at least one of the current generations of console, i.e. they were likely candidates to make a purchase. Tests were run in-house, in regular office space separated from the development teams. Up to 8 players would be present simultaneously, with 1 supervisor for every 2–3 players. One of the supervisors was responsible for taking notes, and no video data was captured as this was considered too difficult to analyse due to the very large volumes produced.

Black Rock conducted "blind testing" meaning that testers had never played the game before, and several different categories of testing were devised:

- **Free Flow.** This is an unguided test where the player is encouraged to play the game however they wish. This is particularly useful for giving an impression of how much fun the game is, because as soon as the game becomes boring they would be inclined to stop playing.
- **Narrow Specific.** In this mode the player would only play a single level, and they might play it multiple times in order to measure their improvement with familiarity. This appears to be similar to vertical prototype testing employed in usability evaluations.
- **Broad Specific.** Similar to the narrow specific test, but playing over many levels. This seems similar to horizontal prototype testing in usability evaluations.

Most of the development team also had visibility of the playtests, but generally it was only designers who observed the sessions. Avent reflects that it may have been helpful for more programmers and artists to also have been involved with observation earlier on.

Despite the absence of video data, one of the programmers on the team had implemented “instrumentation” for the usability tests. This is the process of recording quantitative data directly from the games console that describes the timings of events during the game session. This can be used to measure specific events such as lap times or how long it took to restart a game. Similar techniques have been employed by other studios such as Valve and Microsoft (Steam 2008; Thompson 2007).

At the start of a game *Pure* does not present the player with the traditional selections of Easy, Medium and Hard difficulty level, but rather dynamically adjusts the AI vehicles to suit the player’s performance while they play. During the final 2 weeks of testing, the Black Rock team focused only on balancing this dynamic difficulty system. The team were able to approximate a learning or performance curve by comparing the player’s finishing position after their initial and subsequent playthroughs of a given track. By tweaking the dynamic balance system they were able to adjust this difficulty curve to keep players challenged and engaged enough to replay races and improve their performance.

Avent strongly believes that these tests were crucially important in order to ensure a Metacritic score of 85 %, and that without them he felt that the game would have been reviewed at around 75 % instead. This is a strong recommendation for even simplistic and lo-fi usability testing.

However, reflecting on the quality of results, Avent does recognise some of the limitations with a principally quantitative approach. In particular he comments that even with a large dataset, results could be misleading or misinterpreted. For example, if the data shows that players consistently come last in a race, one response may be to reduce the AI vehicles performance to be more in line with that observed from the players. However, this may be inappropriate if the real cause lies not with the AI over performing per se, but perhaps with bugs in the vehicle physics system which bias in favour of computational rather than human control.

In order to identify such cases, Avent recognises that qualitative approaches would be beneficial and this is where more effort will be focused in the future. He hopes that future projects will take the agile development model even further, incorporating user testing throughout the development life cycle, including during prototyping and pre-production.

Furthermore, usability testing could begin as soon as the team has produces lo-fidelity prototypes. For example an HTML mockup of a front end menu; a 2D interactive demo of vehicle handling; a whitebox of a level where the visuals are of the minimum possible standard in order to test gameplay (i.e., typically without colour, and with purely blocky, geometric shapes). Indeed, while only a small amount of usability testing was carried out from the vertical slice, Avent believes that the team



<http://www.springer.com/978-3-319-15984-3>

Game User Experience Evaluation

Bernhaupt, R. (Ed.)

2015, VIII, 285 p. 42 illus., Hardcover

ISBN: 978-3-319-15984-3