

## Postmortem: Pixelogic's *The Italian Job*

By john li

Pixelogic Games was founded in 1996 and have been developing successful Playstation Games for over 6 years. Under the management of Bryan Reynolds, Pixelogic has successfully designed and completed three original PSOne titles on time and within budget, our goal is to carry on creating many more great original 'AAA' games.

Armed with a cult film licence and knowing that *The Italian Job* (TIJ) already had a huge fan base which would mainly interest the general UK public, we could only hope we could produce a game that would not only do the film justice by not disappointing any fans and also come out with a game worth playing.

When developing any title, there are always tendencies to make the game-play too difficult & hardcore, but due to the nature of this title, a decision was made to go down the pick up & play route so the game was more accessible to a larger audience. There are so many serious driving games out on the shelves that cater for the hardcore racing fan; we wanted *The Italian Job* to be a driving game with a difference rather than just an all out racer.

### What Went Right

**1. The Getaway** The most important part of the game had to be "The Getaway" sequence; to re-create all the stunts from the film was going to be our biggest challenge, the aging PSX is going to have to work overtime to display three detailed Minis, police cars, pedestrians, and general street furniture in a grid locked Turin.

Development of our in-house Continuous Ordered Scenery Streaming (COSS) technology enabled continuous loading of level data that allowed for huge playable cities for the player to freely drive around in. Getting this technology in place early on was a major factor on how the whole development of project would run and how the finished game would play. Once this technology was proven, the artists could then confidently begin to build the vast environments required such as London, Turin and "The Escape Route".



The design and planning that went into "The Escape Route" was probably the best design of all the levels. The art team had the job of re-creating Turin, they started by mapping out the general road layout before positioning all-important landmarks and escape route scenes from the film, then all the extra bits from the movie that were relevant were added and the art team tried to include all of them.

There were so many unknowns to deal with while implementing the actual mission as not everything can be totally designed within a document so one of our favourite techniques which we spend a lot of time on is to design, create and judge game-play 'on-the-fly' during development. Certain memorable scenes of the film would not have necessarily translated to good gameplay, the overall number of scenes in the film was vast and included hiding in car parks, outwitting police on rooftops, and driving down church steps. When analysing the actual film, these scenes of the escape had no connection to each other, it was just a series of stunts performed in minis edited together and most of the stunt sequences weren't even filmed in Turin. We had to fill in the gaps in-between each stunt and scene location to make the whole escape route work as one complete mission. Due to the length of the final route, our initial idea was to break it up into three sections but after more testing and prototyping it was decided that the player was going to have to drive the whole escape route in one sitting in order to recreate the heavy tension from the film of getting out of Turin in three Minis. Carefully positioned cameras were used to capture main action sequences and cinematic scenes while trying to keep it true to the style of the film; the pacing of the whole escape had to be right so numerous cameras were created, moved, removed, recreated and tweaked until the pace of the whole mission began to flow and the final gameplay began to feel like the movie.

Once the route was finalised, the next job was to populate it with a grid locked traffic jam. We ended up having to create a completely new level based on Turin with all the unused parts of the city removed since the amount of polygons onscreen was beginning to grind the game to a halt. As more objects were added, the escape route had to be constantly optimised right through to the end of the project to ensure a good frame rate.

The final Getaway mission consisted of over six minutes of constant gameplay full of real time camera cuts and it was worth all the hard effort; this was going to be our flagship mission so it had to impress. We knew that anyone who plays *TIJ* will be waiting to get to this famous chase sequence and we could not afford to disappoint, we all still enjoy playing this mission now which must mean something about the amount of

effort and time that went into producing it.



Scene from "The Getaway", the game's key mission.

**2. Voiceovers.** About halfway into the project, everything was beginning to come together; we were starting to have a good playable game on our hands and were ready to start implementing speech to set the scene for all the missions. Our publishers managed to get hold of Phil Cornwell who has a huge repertoire of impersonations under his belt, mainly know from "Stella Street", a popular comedy sketch series in the UK. Phil also does quite a good impression of the lead character 'Charlie Crocker' from the film. It was important that we had a decent voice for 'Charlie Crocker' as we knew fans of the film would have moaned if this voice was not right.

The voice scripts for the whole game were decided quite early within the initial design document by Bryan and myself and were only slightly reedited to fit into the finished missions. More importantly, Bryan went down to London's famous Air Studios for the voiceover recordings and since he wrote most of the scripts himself, he could therefore explain the context of the scripts and exactly how we intended the lines to be spoken. Also, the foresight on the day to include extra one-liners and generally allowing the voice actors to have a bit of fun and freedom while recording meant we ended up with more speech samples than we bargained for. Some heavy editing was required to fit over 400 long speech samples into memory with all the other general SFX. The extra resources inspired the designers to add extra speech to missions where appropriate and gave a more involving humour throughout game; Lorna chattering about her shopping in the "Mafia Mania" mission is a good example of this.



Scene from "Mafia Mania".

**3. Where the crow flies.** Our decision to guide the player by only giving them an arrow that pointed in the general direction of each mission goal was derived after much iteration.

Some people like to use maps and some don't and the inclusion of an onscreen map was thrown out due to several other reasons, we were already pushing the onscreen polygon count so it would have caused unwanted slowdown and after watching more and more people play games which included a map, we discovered they generally distracted the player from concentrating on their driving. We wanted to create an instant pick-up and play environment.

We also had grand ideas of subtly colour coding roads and buildings as you drove past but eventually introduced the current arrow system

which pointed the player in the direction of 'where the crow flies' This arrow system seemed a bit too simple so we wanted to try and improve upon it and made the arrow point down each individual road, showing the shortest route to any given destination. This new arrow was in place for quite a few months and the game just didn't feel right. People complained about how linear it felt and even though time limits were quite generous and the arrow automatically adjusted if the player missed a turn, the player still thought they were playing the game wrong if they missed any of the turns. Our decision to return back to the original arrow design was the only way to go for the feeling of freedom and simpler game play - as long as we minimised and prevented those annoying moments where you ended up at dead end road, who needs a map or to be told exactly where to go? It may not have satisfied everyone's needs but it suited our style of gameplay and was well received by players.

**4. Good marketing relations.** It's so important to make sure that any product you work on is portrayed to the public in the correct way. Our publishers marketing department already had some knowledge of *TIJ* and most of them have watched the film, they just needed to be convinced about the game in development. We supplied them with hundreds of in-game screen shots, level data, art resources and early playable missions to project our vision of the finalised product. This proved to be a lengthy but worthy exercise as the more information and data that was sent to them during development, the more enthusiastic they became about the game and the better they did at marketing. This also worked the other way, as more decent press was released, the more it boosted the teams morale when the going got tough and added more pressure to produce a great game and live up to the expectations of previews and general press.

A marketing budget was put aside by our publishers which allowed for posters, t-shirts, key fobs, bags, and PR involving sweet talking journalists, etc which created enough hype that *TIJ* soon became one of the most anticipated games on the PSOne. Even young kids who knew nothing about the film were anticipating the game release, the only thing that could let it down now was a delay in development or a badly executed game on release.

One major market we had to satisfy was the mini owners community, this involved good research and even getting the thumbs up from the Mini Owners Club on final vehicle models and handling characteristics. One person even sent an email after our first screen shots were released stating that they were pleased to see that we had the correct number of spot lamps on the front of the minis.

**5. Team dynamics.** One of the main obstacles while developing any game is how the team works with each other, you can't allow any one individual get disheartened as it will begin to spread throughout the team and soon enough, morale will drop. There were a few lows during *TIJ* involving reiterating work and throwing a lot of it away, this is part of the development process and you can begin to lose sight of the main goal although ideally this should be avoided. Since the core members had a few years under their belts working together, we already knew the most basic thing we could do to prevent low morale was to back each other up, not to slag other peoples work off, find ways of making good out of the bad and generally use good communication skills to solve any problems. Unfortunately, not everyone has the ability to communicate sensibly but you can always fall back on keeping it professional while working. Importantly, and it is encouraged, especially with the newest members is to join in the out of work socials down the pub; here we can talk about things and discuss problems, a sort of group therapy session, some of our best design ideas come from a pub gathering. It became more and more important that everyone worked efficiently together or meeting milestones would require late nights, however we did hit all our deadlines except for a small accident in the later stages, which is explained later.

Our work ethics at Pixelogic is to get everyone involved where possible, all ideas are important, especially at the initial stages and as the project developed, people began to stay more within their own areas of expertise, it was good as we all respected each other opinions and working with talented hardworking individuals pushes you to do your best and more, this also has an impact on quality control.

Forging good relations with the publisher and having a project leader who backs you up when your work might not live up to specifications of the milestone agreement, which could be due to many different reasons such as unfinished technologies or other requirements were more important during that month, etc. The main objective was to avoid doing work for something which wasn't ready to be implemented, this is partly to blame on scheduling but you cannot foresee everything and some things in milestones will need reshuffling during the project, for example, do we need all the vehicles modelled before car handling is finished?

A breather space in the middle of the project was scheduled in, where we all sat back and analysed what we had achieved and what was the way forward, this proved to be extremely valuable as car handling and physics came across as the weakest link to gaining good game play and therefore an extra couple of months of programming time was arranged to correct this. The outcome of the new handling technology improved the game no end, we now could have highly customised handling characteristics, and all 14 vehicles that feature in the game realistically behaved differently to each other, which helped to mould the whole game.

---

## What Went Wrong

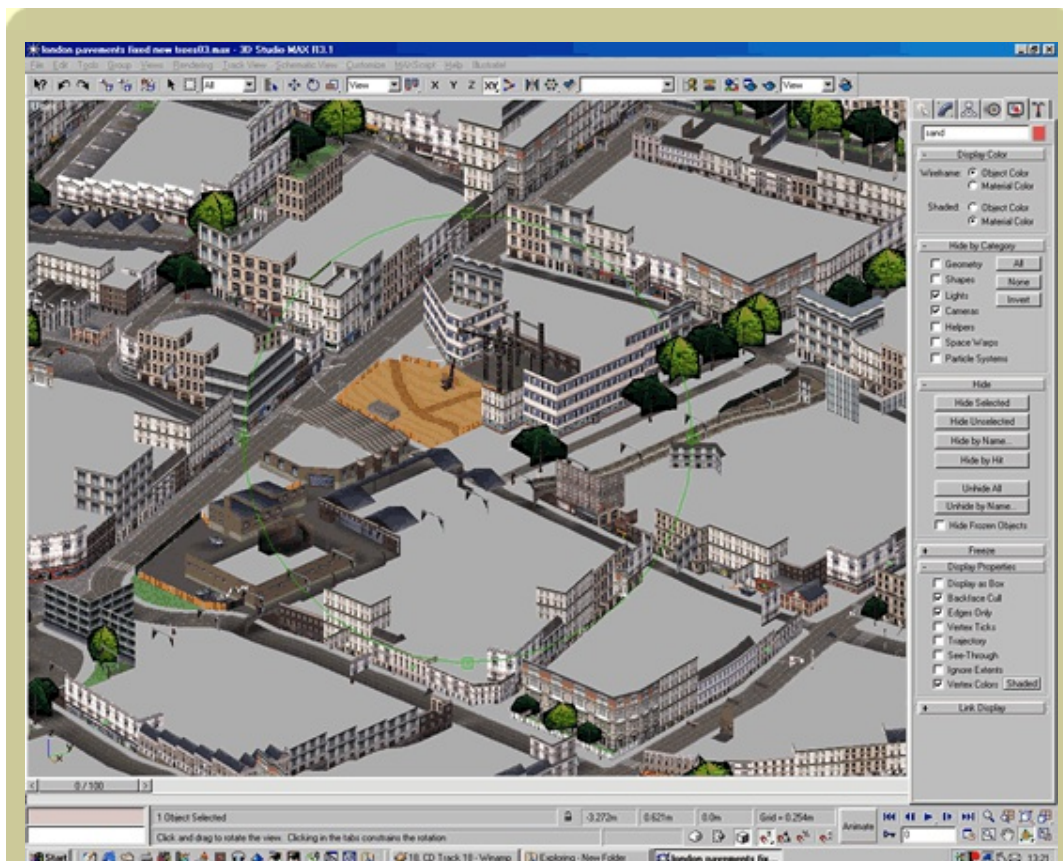
**1. The Chicken and the Egg.** Large amount of art resources were taken up to coincide with the development of COSS, which will allow for large free-roaming environments to be loaded 'on-the-fly', but to test this new technology to its limits, levels needed to be modelled to a near finished state. This whole process involved a lot of art being reiterated and the fact that London ended up being built before the loading technology was finalised resulted in four 'wasted' months, the pressure was on to have a level up and running so playable missions could be implemented, publishers wanted to see a game, these first four months would better have been used finding a good quality/style for the levels.

Basically, all the artwork done on the London in the first four months was scrapped except for the general layout and the whole city had to be modelled again, this new version of the city was done in double time and so there was no time for quality assurance. After a couple of months of our artists slogging away, it became clear that the almost complete new London level just didn't reach our graphical standards so the artists had to start building pre-fabricated buildings with textures and replacing all the poor quality buildings with the improved ones. All this re-working of environments also affected the mission schedule so playable missions were no longer playable and had to be re-tweaked or even re-designed by the designers to make them work.

This turned out to be a good learning experience in how to deal with massive levels but if there was a little less pressure in the beginning while COSS was being developed, we may have actually kept the first few months of work.

Some lessons must have been learnt, the hard work at the beginning wasn't all a waste of time as Turin was built in a similar fashion but actually took less time to build even though the completed city was quite a lot bigger than London.

London finally contained over 150,000 polygons and Turin consisted of over 250,000 polygons. The layouts for each city were based on real road maps so the authenticity of how these cities came across were as accurate and true to life as possible.





## London, as modeled in 3DS Max

**2. Joy riding.** During the final stages, just two weeks before submitting the gold master, Robin one of our programmers had been working over the weekend tying up some loose ends. He left the office, got into his car and started on his way home, meanwhile some delinquent joy-riders were screeching around a corner so fast, they had veered onto the opposite side of the road resulting in a head-on collision with Rob.

The rest of the team found out about the accident on the following Monday morning, Rob was in hospital, strapped down so he couldn't move. The news from his mother was, he could have seriously damaged his spinal column and the fear of Rob being paralysed was in the air, everyone was in anticipation of the X-ray results.

A call was received later that day, again from his mother, the doctors said he was in the clear but he wasn't allowed to move and had to stay in hospital until further notice, a sigh of relief was heard from everyone in the office.

Luckily, after a few days he soon started to recover and was sent home to rest, we sent him cards and joked about ideas of how we considered transporting his PC from the office to the hospital and rig it so he could look at a screen and code while still being laid on his back -- we wouldn't really do a thing like that.

The game still had to be finished so after a few phone conversations we sent the PC to his house in case he got bored and he painfully continued to finish coding his part of the game from his bed.

This delayed the completion of the game by two weeks; I don't think you could ever prepare for such an event in a development schedule, it's definitely one Pixelogic will always remember. Rob has now fully recovered; he only twitches now and again.

**3. Fast-loading blues.** There is nothing worse than having to wait for a game to load, especially if the mission only lasts a couple of minutes.

Our loading system before the fast loader involved having a separate executable for each mission, which had to be loaded in and executed. Once executed, the executable file had to initialise all of its variables and load all of its graphic and sound data, this proved to take too long and something had to be done to reduce the load times.

The basics of the fast loading to be developed involved taking a snapshot of the memory contents once a stage had been loaded with the graphic and sound data initialised, this snapshot could then be loaded at any time and the stage would effectively be restarted. At first, this method was thought to be too complex for a simple problem, and it did not solve the problem of having separate executables for the front-end and every single stage in the game.

More research into solving similar problems involved using memory overlays, which would mean the front-end, and the basic game code would be contained in one executable. Unfortunately this method was not compatible due to the complexity of our current game engine and adding this sort of functionality would have meant completely re-writing the game engine, a task we simply didn't have time to do so it was back to the original snapshot idea.

A basic fast loading system was written within the next few days, which worked fine on the PC but would not work when run from a CD. After days of hair pulling, the problem was traced to a combination of bad CD's and dodgy playstations that had been dropped one too many times on the floor. After more testing and code optimisations, the new loading system worked fine and the loading times began to become more reasonable. The only drawback was that it still had not solved the problem of separate executables for each stage and this became a real pain as we had two main cities which were shared across over 86 stages, so if small level changes had occurred, this meant having to recompile 40 or more stages again before re-taking the snapshot. It got even worse than this, if there was an in-game front-end change; all 86 stages had to be recompiled and run before a snapshot could be taken.

Compile times have always been long, but not as long as we anticipated. Our problems did not stop there, we tried to get a system going where we shared the compiling between two machines to cut the time down, this itself contained many problems as the two sets of snapshots simply wouldn't work together when burnt onto a CD. After many late nights and living out of the office with frustration and another added problem of our CD burner breaking down, the two PC's compiling problem was traced to fact the game data on each of these machines were not totally synced. A streamlined compiling method eventually surfaced that did not involve too many late nights and compiling could be then left to run on one machine overnight until the last couple of weeks of the project where game testers were brought in to help iron out any left over bugs, so compiling became an everyday occurrence until the game was finished. The result of all these problems encountered finally gave us a more efficient system of compiling a whole game build successfully and reliably.

**4. SFX.** The sound code and how we added sound from our previous projects enabled any member of the team to add sound to collisions, objects, vehicles, and levels quickly and easily, however on *TIJ* the code structure of objects differed slightly so we lost some of the flexibility

that our original sound functions gave us. More of our sound playing functions were lost as new technologies were coded such as car handling, rigid body physics, and collision code. While implementing new sound code and adding the actual sound effects to the game, problems began to occur due to lack of sound resources available, because of this some of the engines sounds ended up looping very badly. The main music tracks and basic SFX were outsourced at the start of the project but there was never a review of the ongoing work and this ended up with nobody really liking the results.

In the end it was considered best to re-do all the sound effects ourselves and due to the unforeseen car accident of our programmer who was in the process of implementing this code, it became a bit of a rush job. Eventually with our sound functions back in place where they should be, the game began to sound how we intended.

The amount of speech and effects to be implemented in *TIJ* was far greater than any other game we had previously encountered, a lot of speech samples had to be squeezed into memory with a lot of other generic sounds through heavy editing and compression techniques.

Our mistake of underestimating the length of time for this amount of sound implementation has taught us a valuable hard lesson; we now take much more consideration of sound from the outset to allow us to make proper use of any outsourced sound resources efficiently.

**5. FMV.** The inclusion FMV was discussed at the design stage of *TIJ*, we planned to use our in-house scripting language known as the Mission Builder which allows us to prototype and create game content quickly without the need of programmers, so designers could create short scenes using the in-game engine and blend it to the corresponding missions for a seamless gaming experience.

However, the Mission Builder was never designed to produce complex cinematic scenes and new functionality had to be added to achieve this. The process of creating in game cut scenes could be quite crude since there was a limited amount of feed-back to the designer and to create a short sequence of five seconds could take up to two days which then would have to be constantly tweaked. This involved creating paths for cameras and objects within 3DS Max, then exporting these positions to the Mission Builder where speeds and simple camera behaviors could be scripted to make the scene work. Unfortunately, the designer could not see the total outcome of their work until the stage/mission was compiled and ran.

The whole process took way too long to make a scene look right, with limited camera controls and once a high standard of camera work was reached, it was extremely difficult to leave any other camera work that wasn't up to scratch, the time would have been better used to polish the missions. The amount of time required to produce this type FMV would definitely need cutting down in the future by creating better tools so our designers can produce in-game cuts quickly and confidently.

The main FMV to be shown at key stages of *TIJ* was going to be outsourced by the Publishers, unfortunately this did not get confirmed and time was getting on before we were told that we were not going to get any outsourced FMV. We quickly realised the need to produce at least a minimum set of FMV ourselves to progress the plot within the game. In-game footage had to be created, captured and edited together to create these FMV sequences in the three months left of the project. Some mayor story lines had to be explained and there wasn't much time to do this in, the final results could have been a lot better if we had known earlier that we would be doing this stuff ourselves and time would have been scheduled for this. However, an intro movie was outsourced but these key plot sequences would have been a lot more polished if they had also been outsourced from the start of the project. Production values are so important and can affect how the general public judge any product.

---

## Conclusion

All the development issues we encountered only made us stronger as a team, we solved all the problems and have learnt from our mistakes, some were unavoidable and created a cascading effect on the rest of the game schedule, it has affected how we would approach a new project from start to finish. *TIJ* was finished with a lot of late nights, work being scrapped, code re-written and levels re-built - the outcome has only strengthened the final product as the game improved with every reiteration.

This project was going to be a development ride of a lifetime, the outcome we could achieve was not even considered until the later stages where the excitement generated from the combination of good marketing, good development choices and hard work from talented individuals had crafted a fun playable game for a wide audience. Modesty crept in to prevent disappointment prior to release, but little did we know.

In the first week of release *The Italian Job* entered the UK PSOne charts straight in at No.1 and stayed at this position for over 4 weeks, it had also entered at No.3 in the All Consoles Chart and No.4 in the All Formats Chart.

After being so well received in the UK with help from a cult film licence we could only hope that the game would hold it's own in other territories. This was confirmed by excellent reviews from the rest of Europe and the USA (5/5 OPM), the game even inspired some people to watch the film for the very first time.

The original development team has also successfully completed a PC conversion of the game within six months, which involved updating graphics and technologies to suit the PC. Both versions have successfully been released in the UK.



***The Italian Job***

[Pixelogic](#)

**Publisher:** SCi and Rockstar Games

**Number of full-time developers:** 9

**Length of development:** 15 months

**Estimated Project Size:** 33Mb of source code & over 5Gb of in-game source data Platforms: PSOne, PC

**Release date:** PSOne: UK October 5, 2001

**Platform:** PSOne and PC

**Development hardware used:** Pentium III 500 Mhz, 256MB RAM, 30GB hard drives, Matrox G200 (for PSX), Nvidia Geforce 3 (for PC)

**Development software used:** 3DS Max, Photoshop, Microsoft Visual Studio, Word

**Notable Technologies for PC:** Renderware, DirectX, Bink

[Return to the full version of this article](#)

Copyright © 2016 UBM Tech, All rights reserved