Flypad Report

Stuart Reeves, 7/6/05

The Project

THEpUBLIC is a (£250 million?) large development taking place in West Bromwich. Besides containing cafes, restaurants, meeting spaces, and hosting films and concerts, THEpUBLIC's building will include a hands-on art space in which several exhibitors (including Blast Theory) will be installing interactives. An intention behind the construction of the building and its contents is the re-generation of the local area, and to provide people in and around West Bromwich with an inspiring place to visit *and keep visiting*. The Public, who oversee the entire project, intend for visitors to "revisit it both in the building and via the web" so that they can "build ongoing relationships with visitors," which is through the "visitor as participant." (The Public, General Brief) Sustained interest from visitors is very much part of the intended contribution to the community, and to provide a meaningful and significant space: "The Public is everyone. Everyone who's creative. Everyone who has ideas. The Public is for dreamers, thinkers, doers, lookers. You are a member of the public already." (THEpUBLIC website)

Figure 1. Artist's impression of THEpUBLIC building

As part of the exhibit area, there will be a sense of narrative throughout the visit. There are to be large scale trans-exhibit structures that support this continuous thread, such as a networked data backbone in which ongoing information about the visit is stored, shared and fed into later exhibits. The download area is intended to cater for after the visit, such that visitors can collect data related to their visit as digitally stored or hardcopy.

Experiencing Flypad

There are several floors that make up THEpUBLIC's building. Each floor is part of a large 260

Figure 3. Two of the blisters (left), and an overview of the blisters' relationship to the Tall Trees (right)

metre long spiralling ramp structure that surrounds the heart of the gallery area, a large atrium. The Flypad exhibit is situated on the 2nd floor, as part of the 'Hilltop.' This floor has three 'blisters' which bulge out into the gallery space. In each blister will be five terminals each consisting of a wide screen flat panel, a footpad and motorised camera attached to an arm extending out into the gallery space. This configuration exists within a larger scope, as noted in

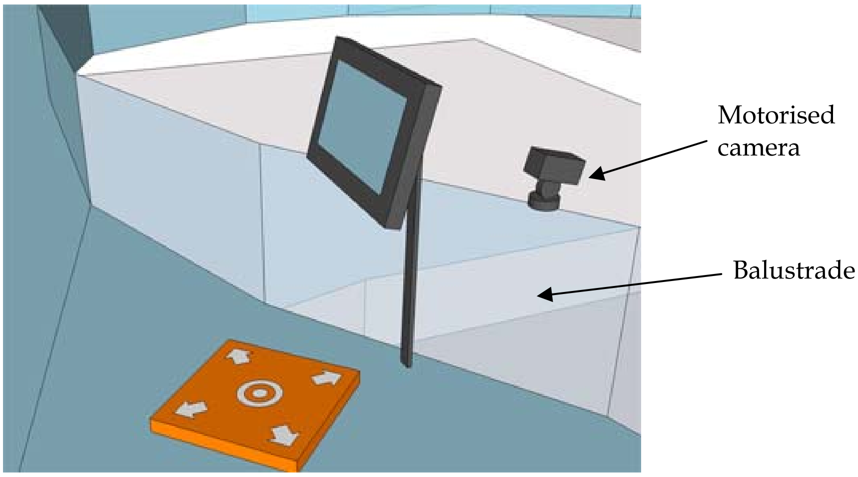


Figure 2. A single terminal setup

the intentions of THEpUBLIC brief, where the visit as a whole extends beyond the moment of contact with an exhibit. In the case of Flypad, visitors will have already created a 'data body' from personal information that was provided at the Input Trees when they began the experience at THEpUBLIC. This personnel data is manipulated and changed by the various exhibits that are visited, and provides a source for the personalisation of each exhibit. The four main data bodies are represented by:

1. External body: Actual representations of the visitor – creating still and moving images & sounds files. Up to 5–10 second clips of each media.

2. Emotional body: Associated responses e.g., “I feel green today” or “how do you describe yourself from these ....” These will be a collaborative artist’s platform.

3. Internal body: Biometric feedback such as temperature, heart rate, skin resistance –the visualisation of this data will be a collaborative artist’s platform.

4. A completely new artist’s work. The combination of data from the inner, outer and emotional will make a representational or data body.

As the visitors pass the Tall Tree canopies on 3rd and 2nd floors they can see their personnel data displayed in the foliage and how it is evolving through the journey.

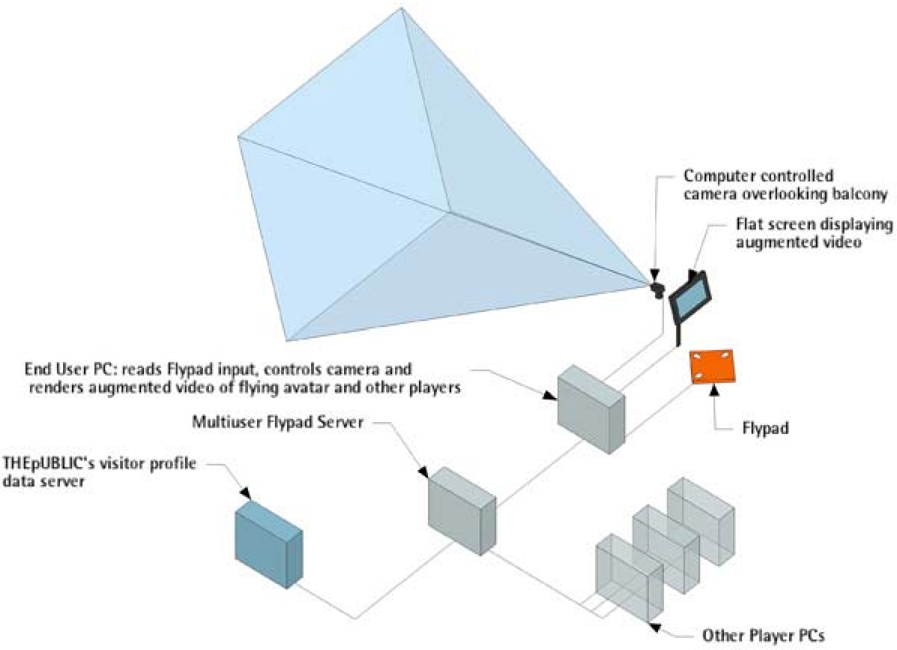


Figure 4. Schematic for a single terminal

When a user steps onto the footpad, which has five directional controls (forward, back, left, right and up), they are presented with a video feed from the motorised camera, which is pointing into the atrium space. The visitor’s avatar is then generated from their particular data body and appears on the screen to briefly teach the visitor how to play the game (the primer). After this their avatar is superimposed on the video feed, and they are able to explore the real and the virtual space in synchrony (meaning that the virtual volume players are able to explore corresponds approximately to the real volume of space defined by the atrium itself). Pushing the various controls of the footpad correspondingly ‘pushes’ the avatar in those directions. As a result, movement of the player’s avatar determines where both the virtual and the real cameras point. Players on other terminals join this shared virtual and real space, with the interactions of their and other avatars being synchronised and replicated across terminals.

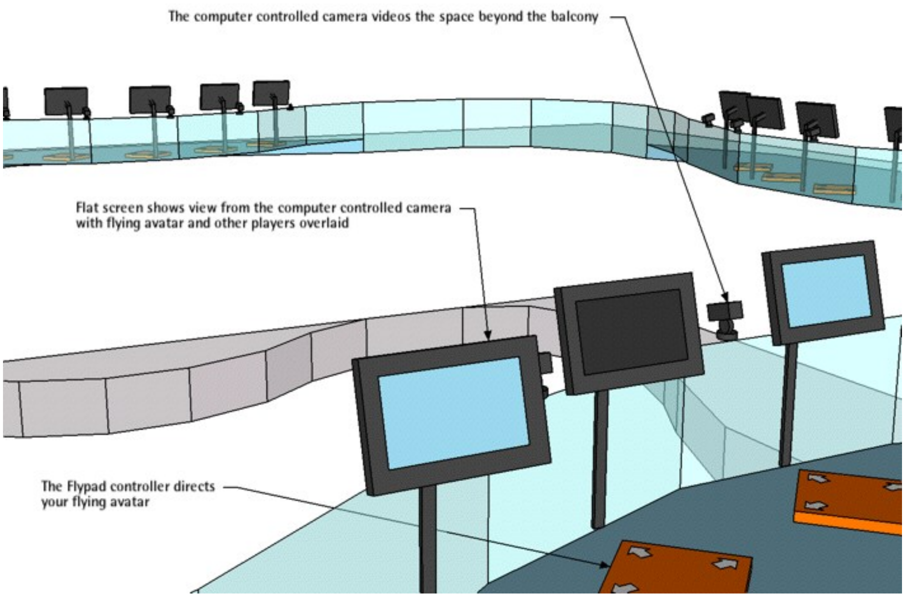


Figure 5. View out into the atrium

The game involves players flying their avatar around the space, and attempting holds with other avatars. These avatars also have particular resting positions and ways of moving when the player interacts with the footpad. The look and feel has been influenced by the Peking Opera, Mexican wrestling, facemasks from various places (e.g., Native American masks), and skydiving. The holds players may make can be between any number of avatars, the flavour of which is perhaps best illustrated by the way in which skydivers can create formations by clinging onto one another's limbs. Besides the holds being pleasing to perform, players receive mutations (e.g., swapping limbs) as a form of reward for conducting a successful hold with other avatars.

Figure 6. Skydiving visualising a collaborative hold

As players perform more and more holds with other players' avatars, so their torso grows in size, and their corresponding weight, causing them to fall to the ground more easily as the game progresses. When the player finally hits the ground, the game is over. A t the end of the visit to

THEpUBLIC, the Download Area permits the visitor to retrieve data relating to their avatar, such

Figure 7. Storyboard for the attractor and primer

as what the final mutated avatar looked like, and so forth.

Views during the game are displayed on the Tall Trees, which consists of large screens situated in midair to one side of the space. These screens are projected images across the atrium, and are prominently visible to visitors on the gallery walkways and on the floor of the atrium itself.

Partners in design

Building THEpUBLIC and populating its exhibits is a large undertaking that involves many different parties. The project's partners as refracted through the concerns of Flypad are the following:

- Blast Theory are involved in the overall installation concept, game storyboarding and avatar design. There is no expectation for Blast Theory to be product designers but we should pass all physical build and health and safety issues through BKD. This includes: Tall trees, camera mounts, Flypad design, hand held Flypad, monitor heights in relation to Flypad etc.
- The Mixed Reality Lab are creating the software for the game, and the interface to the game;
- The Public as an organisation coordinate the whole building and the gallery, and curate gallery exhibits;

- Ben Kelly Design (BKD) are the exhibition designers, responsible for designing all the custom enclosures and mountings for interfaces, and some of the gallery's infrastructure such as the Tall Trees and the design of the balustrade on the ramp;
- Allsop are the architects employed in designing the building and the ramp structure itself;
- Kevan Shaw are lighting designers involved in both the ambient light of the space as well as the lighting systems;
- All Of Us are responsible for gallery-wide interpretation interfaces which describe and explain each exhibit, the Input Tree interfaces where visitors enter/choose personal stuff about/for themselves to be used later on by each exhibit; and
- Schools and workshop groups in consultation for input on avatar and mask designs.

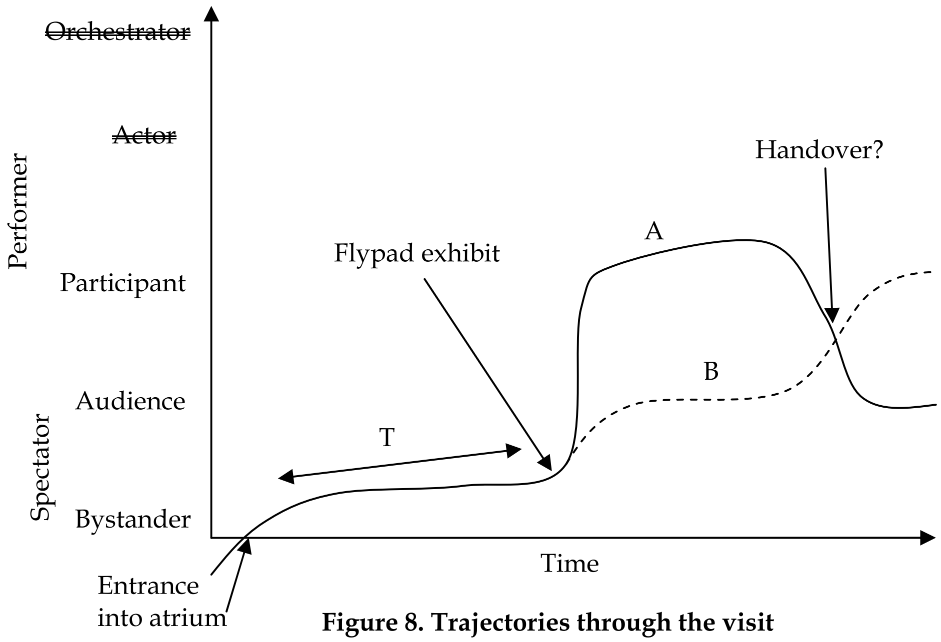


Figure 8. Trajectories through the visit

The development process takes place between various combinations of the partners that Blast Theory comes into contact with. There are meetings and informal discussions internal to Blast Theory that are arranged week by week if needed, with the agenda normally being set by whatever is most pressing. Meetings typically address various issues to do with the design and implementation process, from avatar design through to health and safety concerns. Blast Theory also work closely with the MRL, with Blast Theory presenting game ideas, and going through several prototype demonstrations and development meetings in which the implementation details of game ideas are considered. Meetings with The Public typically involve presentations by Blast Theory of large-scale ideas to the gallery team. Discussion is arbitrated by Andrew Chetty, who is the chief gallery curator and producer, who also arbitrates interactions between BKD and Blast Theory. These discussions are typically centred around exhibit design, where each side presents issues or proposals and provides feedback. Finally, meetings have been held between All Of Us and Blast Theory in which the focus of discussions have been feedback provided by Blast Theory as to All Of Us's storyboarding of the Input Trees.

Coordinating the project with many partners is difficult. The requirements and desires of Blast Theory are contingent upon the manoeuvring space provided to them by the partners. For example, space in the exhibit is particularly tight for each terminal, and the design of the terminal is constrained by the fact that space for a walkway must exist alongside the terminals. Decisions made by the partners provide an area in which to work; agreed-upon design documents and specifications provide an interface between the differing exhibits and the main building. For example, lighting in the building is contingent upon architecture, and affects the possibilities open to exhibitors with regard to using their own lighting (e.g., projectors).

Asking questions

Members of Blast Theory were interviewed about the performance and experience issues tied up with Flypad, with the discussions revolving around four main topics. Although the interviews focussed on these things, time was spent discussing emerging issues not directly questioned for, and then were brought up in subsequent interviews as further points of conversation. The topics themselves drew on previous research (Reeves, 2005), given that Flypad is a public exhibit for Augmented Reality. The experience from the spectator's perspective is inevitable (and integral) to the design of the experience and in considering this, further design points unfolded as a natural result. The topics of conversation relating to all of these issues were:

1. Flow of visitors around the terminals;
2. Views accessible to the player during the game;
3. Relationship between bystanders near the terminals and players;
4. Relationship between visitors on the ground floor area and the displays of the Tall Trees.

The experience

The visit as a whole begins when the visitors enter the atrium area. It will be possible to see the Tall Trees displays, the cameras on arms, and the rest of the terminal setup on the 2nd floor, possibly including players using the footpads. We can graph the trajectory of a visitor through the exhibit, and the various roles they might assume.

There are no 'actors' (i.e., 'front of house' performers) in the work, and neither are there any real orchestrators of the experience other than those curators maintaining the exhibits each day. Visitors begin the experience as 'unwitting' bystanders to Flypad, during which they may see some of the goings-on involving the Tall Trees, cameras and movements of any current players.

> "In terms of people down on the floor itself, we hope that that will be an attractor, I think that the other thing we have talked about is the cameras which will all be mounted on arms extending from the balustrade. They will be moving when the game is in motion and when you lookup I think you will be very conscious of these twelve cameras, and if they are all rotating in space, if all twelve people are playing, it will be a very powerful driver." (Matt)

There is a long period then (T) from the entrance into the atrium, through the other exhibits, until the visitor enters the space in which the footpads and terminals sit. This period is noted as being an issue for the design:

> "As people come in at that level they've got to get their ticket, then go up in the lift to the third floor and then work their way round, so it's quite a long time, even if you said 'that's fantastic, I want to play that immediately,' it would be twenty minutes to half and hour before you get to the flypad ... There's only a limited amount awareness [of Flypad] that we can play with, because [visitors are] not just physically remote, they're temporally quite remote from the experience." (Matt)

When visitors arrive at the 2nd floor, they flow around the terminals, becoming audience members to existing players, or perhaps become participants themselves. Trajectory A illustrates a visitor entering the exhibit and flowing directly into playing the game, i.e., becoming a participant, whereas B shows another visitor spending time as an audience to some current players.

Flow around the terminals

The organisation of the terminals around the blisters present particular problems with regard to visitor flow. Before players step onto a terminal and go through the brief primer screen, there is the issue of how visitors actually get the to terminals, and what the attractor screens display and when. THEpUBLIC had particular concerns about the flow of visitors and how they are guided to and around the exhibit:

> "'One of the things we'd like you to think about is how visitors travel through the space and how you'll control the time they spend there.'" Because there's a limited time people are allowed." (Nick)

The ramp up to the floor Flypad is on

> "invites people to approach in a very linear way, in fact it doesn't invite them, it forces them to approach the flypad in a very linear way." (Matt)

This sets a challenge for the way in which the Flypad design was conceived:

> "The whole work itself springs from the architectural location. ... One of the properties is this huge void of space that the atrium determines, and the other is that it's a U-shaped arrangement so that everyone is looking in towards one another." (Matt)

And so the "linear" approach might cause a "clump at the first blister," (Matt) rather than the reasonably even distribution across all blisters that is desired in order to exploit the sense of space and "encourage people to think about the space between [them]." (Nick)

> "I think it will be really imbalanced if you fill up round one side and you're playing side-by-side, of course a space where in the virtual world you can see the other side and there's actually nobody there" (Ju)

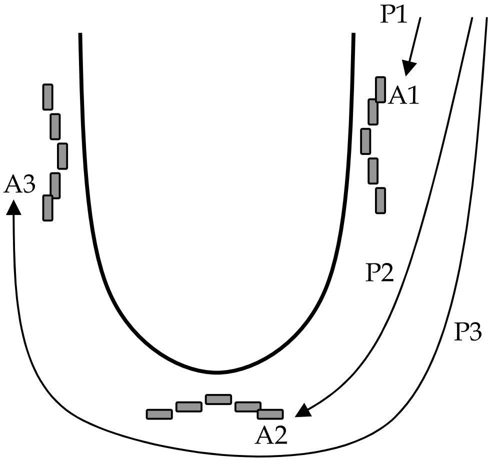
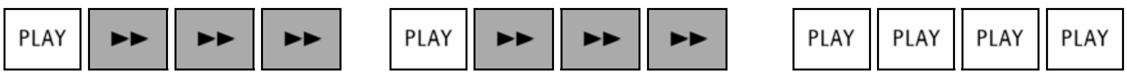


Figure 9. Flow around the terminals

The attractor screens and a sequence of locking terminals down then becomes critical in managing flow. The diagram below illustrates how the attractor may be displayed only on particular free terminals in order to ensure the distribution of the visitors. In this example, P1 (the first player) arrives at the first terminal displaying the attractor (A1). P2 then arrives, but the attractor is displayed on a more distant terminal, A2, and so forth.

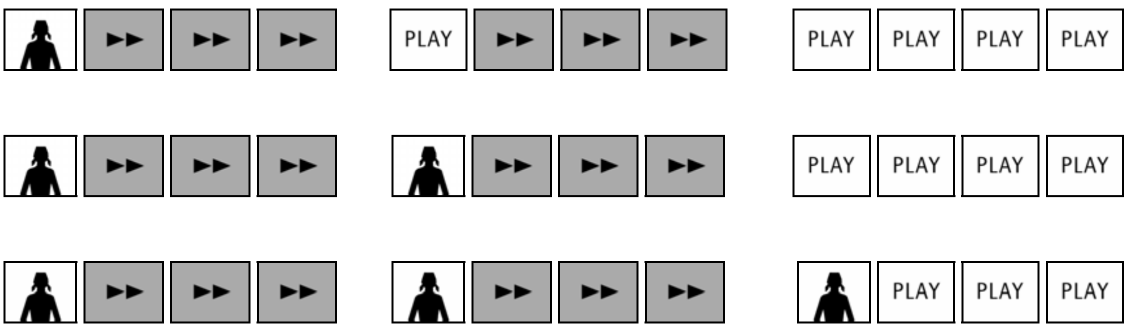
Here is an example flow schematic for the approach of several visitors. When visitors arrive only one terminal is free to play immediately at the first two blisters:

Blister 1. Blister 2. Blister 3.



Additional players are encouraged to find a terminal further down the ramp:

Blister 1. Blister 2. Blister 3.



As terminals on each blister become occupied more terminals are unlocked so there is always at least one space available at each blister.

Blister 1. Blister 2. Blister 3.

There is also the issue of exactly how transitions between being visitors standing near the terminals, to visitors using the terminals in the game, occur. The support provided by the building for identifying visitors will be an RFID system; tags will be carried by each visitor in order to update their data body as they travel through exhibits. RFID readers can possibly be used in Flypad to determine – to an unknown level of accuracy – which visitors are standing in which locations. Tags may be used to "actively pull people onto terminals by name" (Nick) or pick someone at random, or be made available to anyone and subsequently determine the identity of the player. However, this is contingent upon the RFID system:

> "Lots of those [issues] will be determined by how the RFID behaves and what becomes logistically easiest to do. But in terms of swapping, then I think ... at the moment we don't have a sense of what state someone will be in by the end of the game, whether they'll be slightly hysterical ... or really disappointed because their avatar will have collapsed, or they'll be "had enough of that, get off there," so how that moment [of transition] is dealt with is hard to describe ..." (Nick)

So the moment of handover between two players is relatively unspecified at this time, and is reliant upon the fluidity of the ending and how it fits into the cycle of transitions between players.

> "We've got an ending, which is only a theoretical ending, which is where you can't fly any more because you become too heavy. It just makes it harder and harder and then eventually you have to give up ... but that's only a theoretical ending, we haven't got any sense of how that would feel or whether it would make you really frustrated or whether it should let you fly off ..." (Nick)

Here there is a particular relationship to those locked-down terminals and the displays of the Tall Trees, which are provided to Blast Theory as a set of resources integrated into the building. THEpUBLIC brief informed Blast Theory that the Tall Trees "should show what the Hilltop was or might be going on there," (Nick) and so the feeds from locked-down terminals and at other times game views can be displayed on the screens to provide an awareness of some elements of the game (a "partial view" (Ju)). The Tall Trees are "a way of seeing the game as a whole" (Nick). Physically, the displays are "made large to attract people" (Ju)

> "So the people who come into the building on the ground floor, it's a public area, so it's a non-ticketed area, it's almost like people look up and see what is going on in the gallery. That was the main brief from THEpUBLIC ... We approached it from doing spectator clients for Uncle Roy ... there is a sense that there wouldn't be enough terminals available for everyone to play, so it's a way of looking and learning about what it is without having to step up and play it." (Nick)

In the implementation, the Tall Trees should display recordings of recent or demo games overlaid onto live video. When people are playing, the live game should be displayed on the Tall Trees, in order to show holds and perhaps wide-angle views of the action. In conflict with this, however, are the technical limitations imposed by budget and time. For example, budget constrains preclude a further terminal to act as an independent "spectator client" with its own view on the action, and so the feed from existing terminals must be exploited. Any terminals that are not being used may then act flexibly as the independent spectator client.

Performer experience

The view players experience from the position of being on a terminal is important to the experience. There are several things potentially in the visitor's view as they stand on the pad besides the view out into the general atrium space (Figure X). The screen obviously obstructs a large portion of the view, and in some ways is detrimental to the effect intended in Flypad:

> "Because it's AR ... there's almost a sense that what [Flypad] is trying to do is give you a sense that ... this virtual person is floating around in a real space, so in a way for there to be a screen at all is a bizarre re-representation of something that should already be there, conceptually you should be able to look out into the space and see the thing, and so the screen is almost like a stand-in for what you would see if you were able to see." (Nick)

> "What we're trying to do is make sure that the virtual representation and the real space which sits around it are as seamlessly interlinked as possible, that there's a very fluid relationship between the two, that as you move your eye from one to the other, that's very easily achieved. And that the

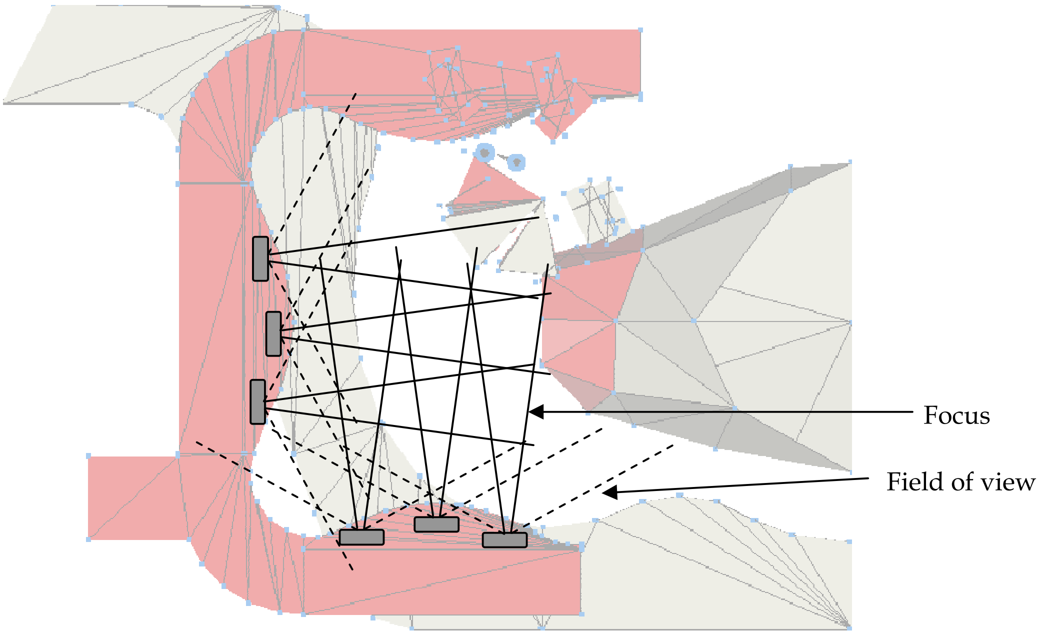


Figure 10. Fields of view in the atrium (only six terminals illustrated)

> sense of play that you will experience as you dart between real and virtual, and experience the frisson of this difference, is a very important part of the pleasure of it." (Matt)

Figures 11 and 12 illustrate the spatial character of the atrium, what is viewed, viewable and where the game action takes place.

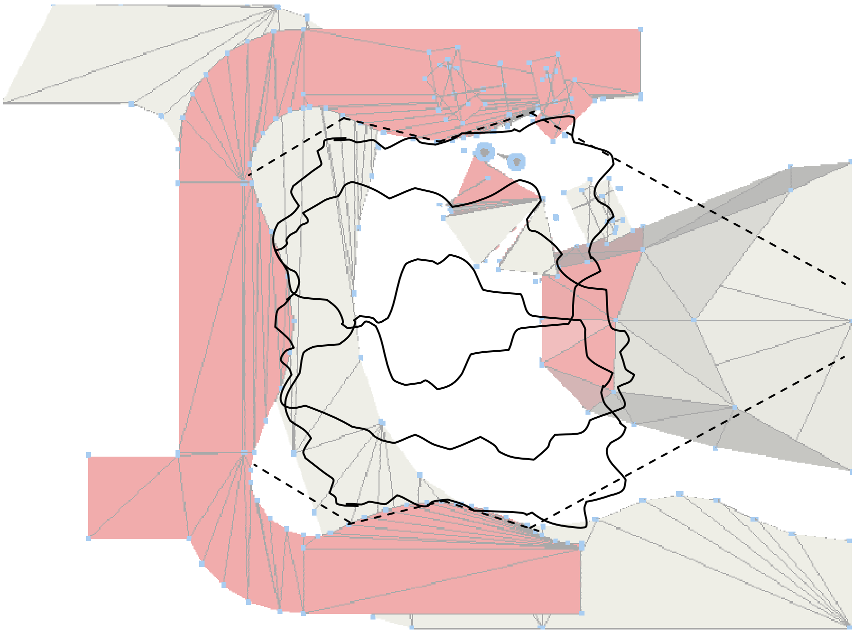


Figure 13. Areas of interaction

The 'target,' being the area in which the augmentation takes place, and the 'device,' being the screen on which the augmentation may be viewed, are separated. The 'safeness' of the target and device/display separation is part of the design thanks to two aspects:

1. It is impossible for players to obscure the camera since the cameras are located on the edge of the balustrade;
2. The gallery below will have people walking through it. The AR space/volume in which game activity takes place has been defined such that the volume avoids potential occlusions that may happen between virtual avatars and real people, or avatars and physical features, such as another balustrade or the Tall Trees.

Each set of terminals sits in one of the three blisters. There are balustrades that line each blister, and these are also a potential obstruction to the experience, which is in essence attempting to seamlessly interlink the real and the virtual aspects of play. Initially Blast Theory wanted a glass balustrade so that players could see out into the atrium and across the space as well as viewing the atrium via the screen, however this was rejected by The Public in favour of a metal mesh, therefore providing at least a semi-see-through balustrade structure. The balustrade is only one part of the terminal's configuration, however, and in reality each and every physical object that goes to make up and surround a terminal becomes important concerns when considering the player's relationship to the space of the atrium:

> "So what we look to try and do is have a screen that's as big as possible, with as small a surround to the screen as possible, on as light an arm or mount as possible, on a glass balustrade - if we ideally could - with the smallest steel posts that we could possibly have. As we've had to give ground on some of these issues so that the balustrade is metal mesh, or example, what we've then tried to do is position the screen and the pad itself as correctly as we can in relation to the balustrade, so that as an average height person is stood on it, the way in which their eye moves from the screen to the atrium around them is easily done as possible." (Matt)

There is the issue of the player's interaction with the footpad, and the way in which this was conceived. There is an empty space between the front of the player and the screen; Flypad's footpad device was employed so that this gap would not be filled. Blast Theory have reasons for using such a design that was a "transparent" "way of walking around without moving" (Nick):

> "[W]e ... like interactive devices that don't actually get physically in-between you and the experience ... we like hands-free devices that you can learn intuitively by your body that don't preference quick digit use and a necessary understanding of "computer games" ... It's about getting an experience which is physically different from just cerebral and hand" (Ju)

There are also more political and social motivations for the physical configuration of the interface used in Flypad. It is noted that the restrictions created by the interface ensure that

> "[i]t won't physically be possible to hunch over your screen and to take a real ownership over it as 'my private space and I'm busy.'" (Matt)

The shareability of the terminals also extends beyond making the interaction legible to bystanders (discussed in the next section). The "one-to-one-ness" of many existing interactive exhibits is unappealing:

> "The idea that you have your thing which belongs to you which you use is a very particular concept to attached to consumerism and all sorts of larger issues like how people understand rules of ownership." (Nick)

Physically the design is a direct descendent of the footpads in Desert Rain, however the initial design for the footpad construction had five pressure mats for each contact – forward, back, left, right, and 'boost' (for keeping the avatar off the ground). Desert Rain's footpads, on the other hand, had some tilting action as part of the construction. Switches ; the effects of this design decision were experienced:

> "you could look into all these cubicles and watch half a dozen people doing the most bizarre physical manoeuvring to try and get their pad to work ... Some people were just astonishing about what they thought would make it work, doing funny little hops, wiggling their hips back and forth" (Nick)

Since the pressure mats would not involve the tilting Ulla reports that after discussions with Blast Theory, her design moved towards tilting movements and a similar construction to the original footpad in Desert Rain. Blast Theory preferred the tilting element since it requires more bodily input than pressure pads would need. As a result, pivot points were introduced into the design (in the centre and halfway along each edge), such that the pad can be tilted towards each corner.

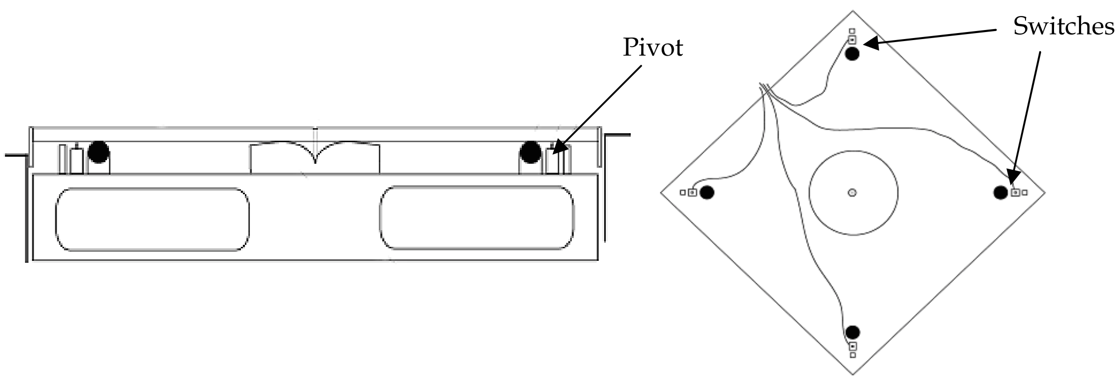


Figure 14. Schematics of the footpad

Spectator experience

Visitors will potentially 'spectate' in several different circumstances. Other visitors besides those using the terminals may be surrounding the players in the blisters, may be on the atrium floor, may be engaged in another exhibit and so forth. The Tall Trees obviously feature for each of these groups.

In particular, there are issues about the noticeability of a player's movements, i.e., whether they are going to draw spectator attention, how 'legible' or 'readable' use of the interface is, and also how 'learnable' for those about to step on the footpad. Issues such as these are influenced directly by the design of the player's experience. For example, the choices made over the use of the footpad for its interactional transparency, the balustrade and the motorised cameras all contribute to the experience for the spectator as well as changing the experience for the player of the game.

> "I think potentially people can learn how to use it in a way that people uhhh ... dance pad machines and people stand behind them practising the moves." (Ju)

> "I don't think we were interested in making something where you had to move around so much that you made a spectacle of yourself." (Ju)

> "I think it's quite important to introduce the process of learning into the work itself that with this it would probably be interpretation and introduction screens" (Ju)

> "Our work is made based on a belief that the audience has something to contribute to the finished art work. Our work does not exist without the audience or visitor." (Proposal for Flypad)

Technical and Game Issues

The layout of the entire Flypad exhibit is meshed within the larger context of THEpUBLIC's network and database. Each terminal's screen, footpad, camera and RFID reader is connected via KVM to rackmount PCs running the game software. These computers are networked to one-another and a game server via gigabit ethernet which in turn connects to the main network between all exhibits in THEpUBLIC. The game server retrieves the visitors' data bodies via this main network. Finally, the Tall Trees are connected via a KVM switch which is fed from the outputs of the terminals.

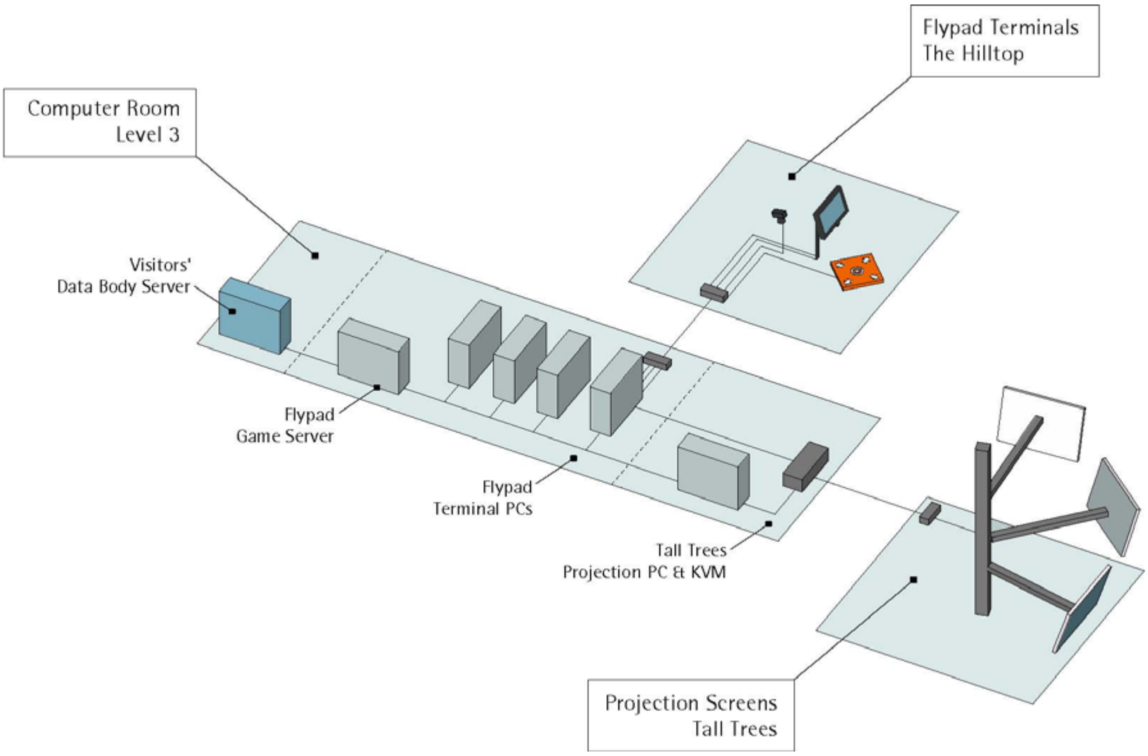


Figure 15. Schematic of Flypad

Figure 16 shows an early diagram detailing the logical organisation of the software components to be constructed in the development process, whereas Figures 17 and 18 show the physical organisation of the exhibit. The game server updates each game client via the network layer. Feeding back into the client as input are the RFID reader, footpad and camera data, the details of which are processed by the game server, updating the shared physics simulation, applying mutation/game logic and communicating with the database structure. The client also has an associated renderer for graphics output, linking in its locally replicated physics scene with the current graphics scene and video feed from the camera. In addition to this, the client has logic related to the primer and attractor and needs some communicative glue for sending and handling camera requests, and RFID requests.

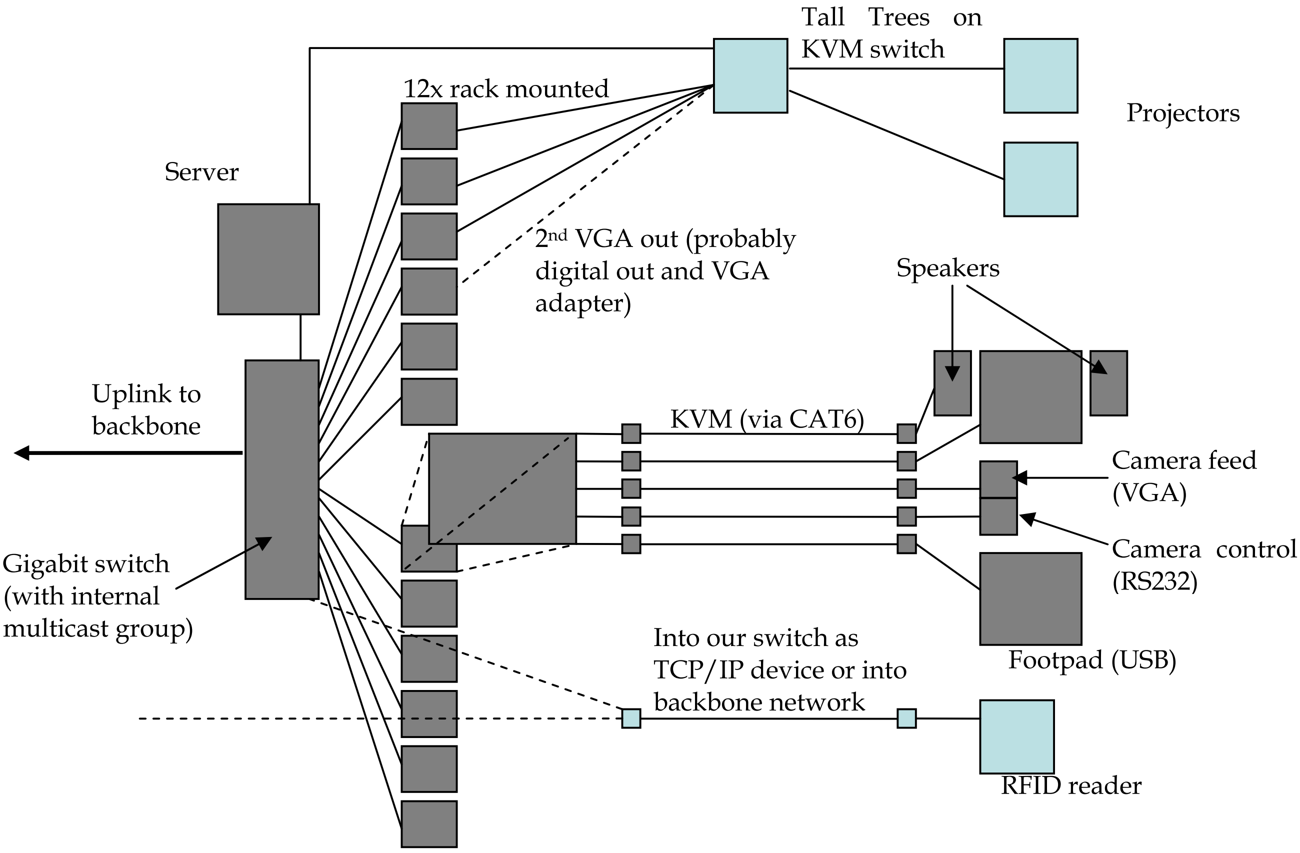


Figure 19. Schematic of the technology involved in Flypad

Physics

The decision was made fairly early on to use a real-time Newtonian physics engine to simulate interactions between the rigid bodies that go to make up avatars in the game space. Each body part of the avatar is represented as a physics volume, with one or more points jointed with other parts. In the diagram below, the 'forearm' and 'hand' bodies that make up an avatar's arm are indicated, as is the position of the joint that exists between them. There are fifteen of these volumes (e.g., head, forearm, foot), the (fourteen) joints between which are determined by a bone structure, where a 'bone' is a line between two joints. The physics bodies themselves are all

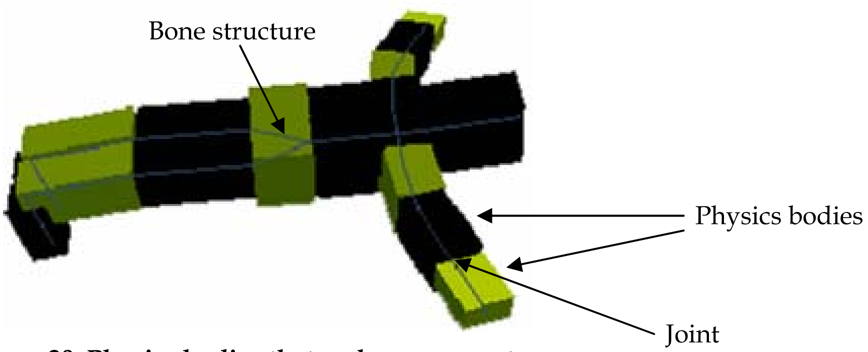


Figure 20. Physics bodies that make up an avatar

boxes; for simplicity, other shapes that were possible were avoided, such as spheres, and capsules. Various types of joints can be formed between physics bodies, such as spherical (ball-and-socket) joints or revolute (hinge) joints. The joints also have various attributes that can be set, such as springs, dampers, and twist limits. These constraints apply force to return the joint to a particular position or perhaps stop the joint moving past a certain angle.

This approach has several advantages over keyframed animation of avatar holds and movements, in particular that the motions of the avatars are not predefined, but rather calculated, and it is therefore highly unlikely that identical motions may occur. This has the advantage of opening up the possibility of emergent gameplay since the game space is no longer discrete (for all practical concerns). Forces are applied to avatar bodies to push players in particular directions, or to direct limbs towards hold positions, rather than discretely triggering a move or hold given some circumstance. A keyframed approach would mean that a pre-defined number of avatars interacted with one another using a pre-defined number of those animations, limiting the possible number of interactions available to the player. The task of defining these interactions would be an arduous one, with the limitations of the game being directly influenced by the amount of time spent designing the interactions. Since the exhibit is intended to be present at THEpUBLIC building for over five years, the amount of recurrent interest is questionable. A potential disadvantage to using physics in this way, as opposed to triggered animations, is movements and holds cannot be 'guaranteed,' however this may instead be seen as another positive feature which encourages emergent behaviour.

Many games permit emergent gameplay of different forms, from actions in-game (games like Sim City) to actions outside or surrounding the game in some way (e.g., machinima, where films are made using game engines). In particular, we are interested in emergent gameplay enabled by the use of real-time physics, however there are fewer games to draw on in this respect. There are games in development exploiting this idea, such as Spore and Clowner Strike. Spore is a game where behaviours are generatively constructed from the way you develop an animal character. Genetic developments that you gift your animal with directly impact the way it walks, fights and behaves. Clowner Strike is a modification of Unreal Tournament 2004 in which real-time physics is employed in order to enable players to collaborate around objects and perform stunts and tricks.

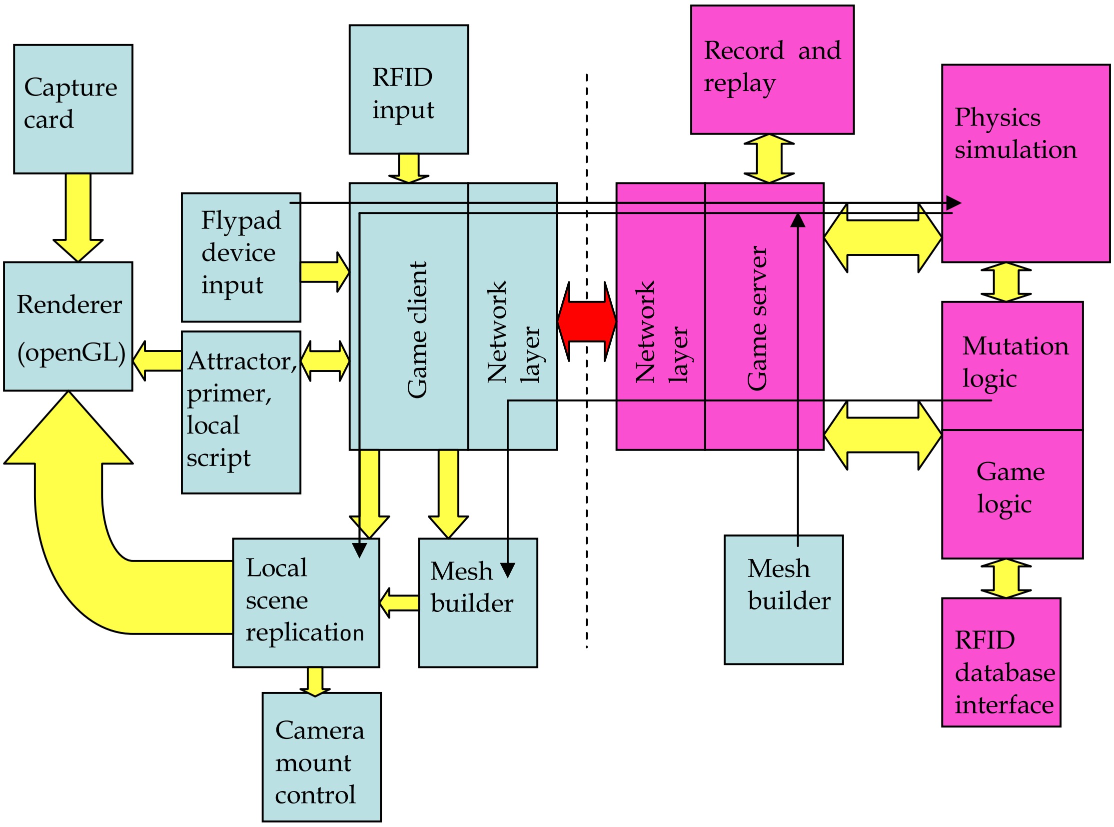


Figure 21. Software architecture

The physics bodies that make up each avatar sit inside a common physics 'scene.' Each terminal needs to be displaying the same scene, albeit from a different view. When players join the game, their terminal will be sending information about their manipulation of footpad contacts, all of which will be processed by the server. The game server then applies appropriate forces to the physics bodies of each player's avatar.

Holds with other players cause mutations and swapping of body parts, which means that skins, meshes and physics bodies could be swapped between avatars. For the physics bodies, the fact that every avatar is identical in structure means that varying sizes and configurations of body parts may be 'swapped' by exchanging the scaling of parts of the avatars' bone structures. The use of a bone structure means that the volumes of the physics bodies are not directly manipulated, rather changes to the bones determine the scale and positionings of the physics bodies.

Networking the physics

Networking physics simulations are difficult to produce. For the Flypad physics simulation, the game server maintains the authoritative version of the scene, which is then sent in some manner to each client. Any forces applied and subsequent happenings as a result of these forces (e.g., colliding with another player) are calculated by this authoritative model step by step.

It is possible to send the entire physics scene over the network to each client machine. This involves packaging up descriptors for all actors (physics bodies) and joints, and sending them as a packets to the clients. The scene information is then recreated on the terminals, exactly replicating the state of the server's physics scene. Whilst this information is relatively small in size (how many k?), the frequency of updates required to reduce the deviation between update and subsequent calculated steps on the client is high. Copying the entire scene involves networking a substantial amount of functionality that is associated with the scene, and as such becomes a tricky, complex job. This potentially results in a less stable networked physics simulation, which is unacceptable when we consider how central the physics is to gameplay. If this functionality is cut down, then it is harder to achieve determinism in the client's replication and stepping through of the scene.

In order to overcome these problems, a different approach was required. The first idea explored was packaging up all forces to be applied to physics bodies and sending them to clients. With this method, clients would need to step through the simulation in synchrony with the game server, applying forces at the right time in order to achieve a determinism that did not deviate from the shared scene. DETAIL HERE

The second idea was to simply send the pose of each avatar, and the rotation of each physics body (i.e., a position and fifteen quaternions). The positions and rotations could then be set directly on the client, and if updated rapidly enough (100Hz), would avoid the application of any forces on the client machines. This method exploits the fact that several aspects of the game are known and/or fixed. For example, all avatars have the same bone and body hierarchy (e.g., the left arm consists of three separate physics bodies and is attached to the torso) and in this way only one position needs to be set. Since the bones are a known length or scale, positionings for each physics body can be cascaded down and then rotations applied. Further to this, we have a fast, reliable connection and a known hardware platform that the packets are being distributed to. Essentially the client machines in this version become networked renderers of the current physics scene that is calculated step by step in the server, with explicitly set positions and rotations for each physics body in the client's scenes.

Movement

The avatars move about the space via the application of forces to their physics bodies. Simply applying force to, say, the torso, produces a dragging effect, where limbs and other body parts move by virtue of their link to the torso. Forces that are applied during movement can be of two different types: impulse forces and continuous forces. Impulse forces are applied with all the force's energy which is imparted to the object instantaneously, whereas continuous forces are applied to the object over a period of time. (Note: in physics, impulse is a force applied over a

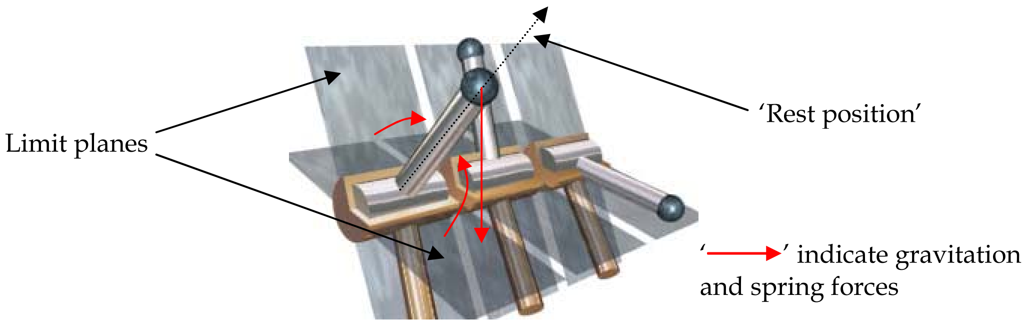


Figure 22. Limits for revolute (hinge) joints

period of time, i = f * t?). Torques can also be applied instead of the regular forces, so physics bodies can be given an angular momentum about a given axis rather than a linear momentum in a given direction.

The movement of avatars through space was envisioned in the proposal for Flypad as "floating" and later some elements of skydiving were discussed. The way that avatars float, fall, move became more detailed, however, as different styles of 'rest position' and motion through space were discussed.

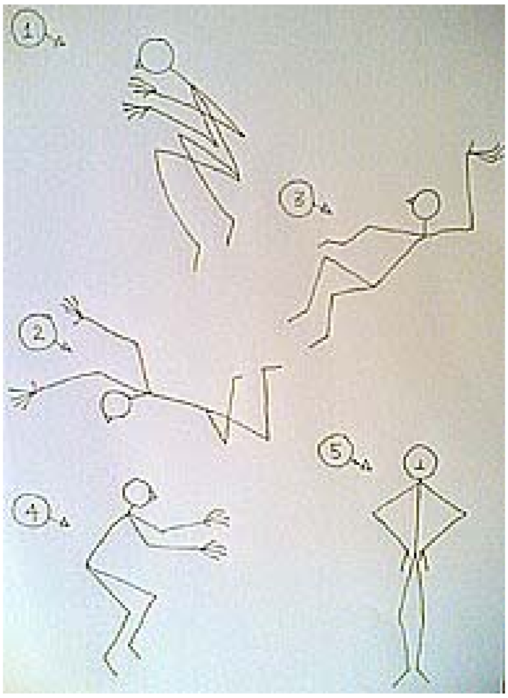


Figure 23. Avatar rest positions

Each avatar's rest position defines the way in which springs, dampers and limits are set for each joint in their body. These settings return physics bodies to particular angles in relation to one another when no force is being applied to the physics bodies of the avatar. For motion through the virtual space, gravitational force and the vectors of forces applied by player actions (e.g., flying to the left) can be set on a per-body basis, meaning that in addition to the characteristics of the joints, each physics body can have different of spring, damper, limit, and gravitation and force vectors. Figure 24 illustrates just part of this, showing limit planes and spring forces, a rest position for the bodies, and the gravitation force vector. The result is a complex interaction of all these attributes to produce a certain physical effect.

Figure 26 shows a document detailing five example rest positions for different avatars. For example, in (5), the rest position straightens the avatar, and bends the arms in the way shown. During movement, a force applied to the head and upper arms, for example, would make the avatar go head-first in the given direction, with its arms pulling up level to the shoulders.

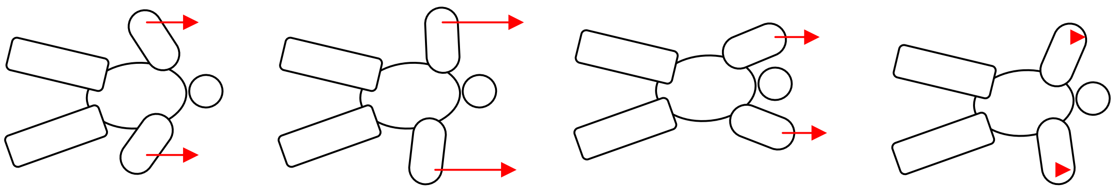


Figure 25. Cycles of forces

Avatar movement was initially conceived of as being animated, however in the transition from keyframed animation to becoming completely physics-driven, the possibility of crafting the way of moving was lost. Instead of applying simple forces to avatars, cycles of forces have been used to propel avatars across the space. Figure 27 shows a simple cycle of forces on the arms of the avatar, resulting in a 'swimming' motion.

Holds

As part of the game design, it was decided that colliding avatars should perform holds with one another. The aesthetics of these holds have been drawn from Mexican Wrestling in particular, and Figure 28 illustrates an example. Several discussions addressed the basic way in which holds might be achieved. Points at issue were:

1. How do avatars get drawn to one another before they are close enough to perform the hold? Are the players required to do all manoeuvring, or are there some attractive forces to help them?
2. How is it decided which hold to perform? Do players use some special footpad combination presses, or does the system decide based on some conditions (e.g., for avatars above and below one another)? What about multiple avatars in a hold?
3. How do players break away from a hold in progress without it appearing to be a failure to the system?
4. When and how does mutation occur?

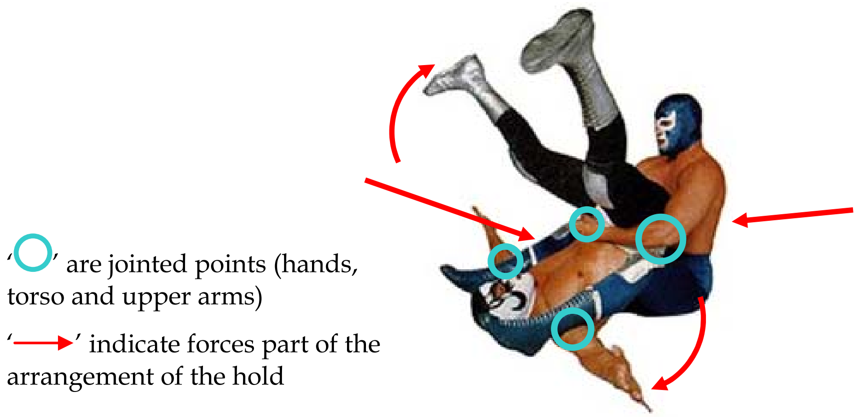


Figure 30. Example of the *katakana* hold

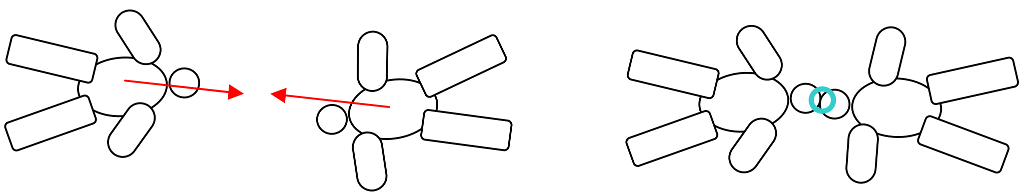


Figure 29. The *headbanger* hold

Going into a hold

Deconstructing the hold in Figure 31, we see that it contains several joint points and requires a relatively complex sequence of configurative action between the avatars before the hold is attempted. Firstly, the two participants need to be drawn together in some way that is beyond just the direct control of the player. Whilst game interactions centre around a player's ability to position themselves correctly, some amount of 'help' provided by forces may be appropriate in such a complex hold. In the katakana, this could involve small attraction and rotation forces (indicated as red arrows) in order to make alignment for the hold more feasible. Other more basic holds, on the other hand, may require little to no attractive/alignment forces because of their simplicity. The 'headbanger' hold, illustrated in Figure 32, is far simpler than the katakana and might require only she smallest of attractive forces.

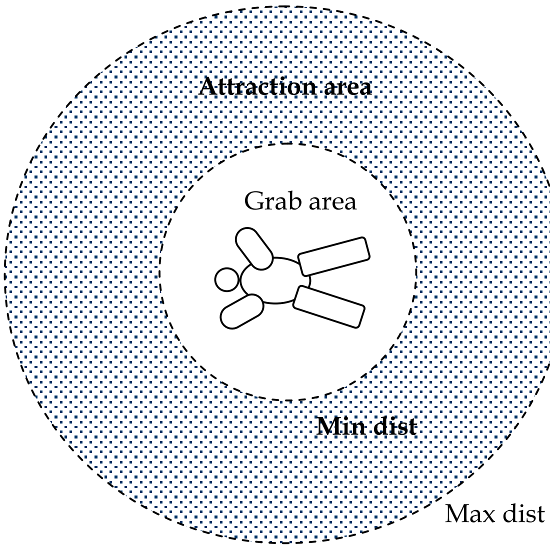


Figure 34. Hold boundaries for avatars

If attractive forces are present, avatars can have several configurable boundaries between them that might be determined on a per-hold basis. The maximum distance determines how far away any attraction between avatars may begin. Once inside the attraction area, avatars begin to be drawn together in some appropriate manner (as we saw in the katakana, rotation in addition to a general attraction). When inside the minimum distance, avatars enter the grab area, causing some sequence of grab attempts to be initiated. Quite how a particular hold is decided on is an open question that is addressed later.

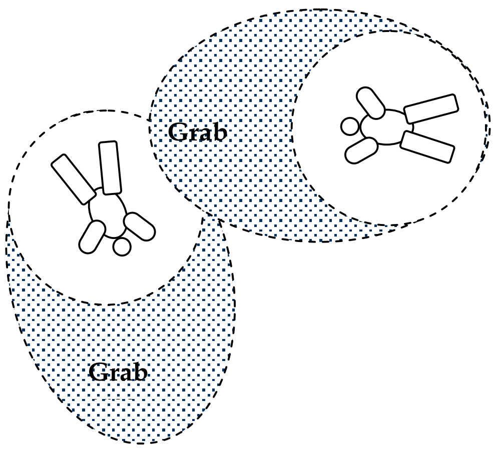


Figure 33. Mutable grabbing areas

Finding the closest hand, foot or other contact point may be constrained by a current state for the grab area. Grab areas may be mutable, and therefore the avatar may only need to search within a particular space, say, ahead of its current orientation. The previous diagram shows avatars with set areas to look only for grab-able limbs in front of them.

Deciding on holds

A hold is an attempt to get one bone into contact with another bone (which may or may not be on a different avatar), then if they make contact, form a temporary breakable joint. Forces that go to make up holds need to operate with the physics simulation rather than forcing physics bodies together. If, for example, an arm is forced into a position that goes against the shoulder joint, it has a tendency to snap off. For example, we might apply forces to a hand in order to move towards another hand, then after a given time, make the joint and notifying us of success, or give up and tell the system that this part of the hold has failed. In this way, the bone, physics body and joint model of the avatar can be complied with. For example, an avatar with long arms and flexible joints might be able to make the move, but another avatar with short stiff arms would not, and visibly so. Avatar physical build thus naturally determines which moves work and so help build complex and unpredictable holds from atomic grabs.

We define a 'grab' to be part of a hold that is an attempt to link the surface of two physics bodies together, regardless of avatar, which will fail if the avatar(s) physical arrangement doesn't allow it. We can create a palette of these grabs, which could be triggered automatically when we notice that two avatars are near each other in a certain way, or triggered by doing some sort of special sequence of pressings of the footpad (in a similar vein to the way beat-em-up, fighting games enable players to perform complex moves). Whilst both options were discussed, it was felt that the qualities of physically interacting with the footpad did not lend themselves to special sequences, and also that SOMETHING ELSE CAN'T REMEMBER.

Different avatars might have different holds, or holds that they are more likely to perform. Because we know if a hold has failed or not, we can build up complex holds from a series of simple grabs. For example, if an avatar grabs a hand, then the other hand, then attaches its feet to a torso, the result would be a relatively complicated hold. We could define a chain of these grabbings by setting a first grab, then if that is successful, try and perform the next in line, then another and so forth. If one of them fails, then chain fails and perhaps some alternate route through the hold is decided upon. Here we have a tree of moves, briefly attempting StartGrab1 or 2 with a couple of alternate MainGrabs and FinishingGrabs.

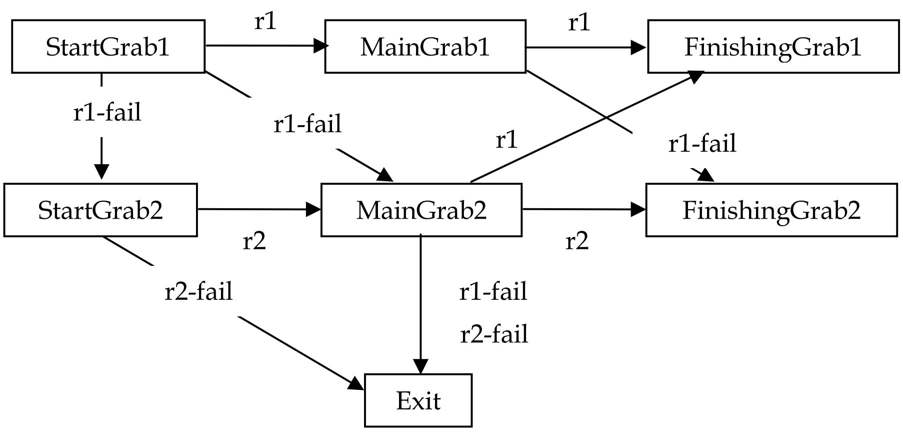


Figure 35. A tree of grabs that make up a complete hold

There are three suggested levels of complexity:

1. A grab sequence for the hold is chosen (somehow, maybe from orientation, limb positions etc.) from a palette of these sequences that the avatar has or come from a pool of shared sequences. A grab sequence has:
    1. A sequence of abstract grabs;
    2. Some kind of 'state,' e.g., what position it is in the cycle of grabs;
    3. A mechanism for following several paths of grabs/adapting to what grabs might have become available for different avatar types and avatar groupings;
    4. The sequence must then interpret this dynamic information into new grabs.
2. A grab is:
    1. Constructed from some lower-level atomic grabs;
    2. Some descriptive abstract language for specifying grabs in a high-level way, e.g., "left hand grab any thigh" or "nearest limb grab head"; and
    3. This level of grabs have to be interpreted from these descriptions into atomic grabs.
3. An atomic grab is:
    1. A basic attractive force and/or torque that manipulates one part of the body, e.g., one physics body that makes up a limb; and
    2. Able to decide whether the this atomic grab has failed in its action (i.e., and tell the grab it is part of this information).

How the initial chains of grabs are 'decided on' is the most difficult matter. A simple model, for example, would be to have a set number of 'complete' hold positions (e.g., as depicted in the diagrams), and decide which hold a pair of avatars should engage in given an approach direction; e.g., if avatars are approaching head-on, they would attempt the headbanger. This technique potentially would cause problems for holds that are only partially accomplished, since there is no defined way of recognising such partial holds. The exit strategy for a hold (next section) would become more important for players, as would enabling them to identify that the hold is partially completed. This is clearly difficult.

Another suggestion was to approach holds in a more modular way with less focus on the 'complete' moves being the only end-points for a particular hold, and with new holds being reported back to the player as they perform them. The intention behind this is to provide players with a sense of accomplishment even if they never reach the 'end state' of the hold. For example, during the course of a katakana, there are several component holds, such as getting the avatar's arms round the other, that can be reported back to the player as they do them. In this way, even holds that go 'wrong' can be rewarding experiences for the players, since they will still get a sense of building achievement.

Multiple avatars in a hold potentially cause problems. Holds between only two avatars are relatively straightforward, however directing several avatars towards a particular hold is far more difficult in terms of the trade-off between player control and hold 'coherence.' A possible solution is to use the less specified approach to holds in which, as mentioned before, complete holds are only one of many potential end-points.

Holds perhaps need to be authorable in some way. One approach might be to develop an XML schema to deal with the palettes of holds as described previous.

Leaving a hold

How a player exits a hold prematurely is problematic as well. How is an attempt to pull away from partially connected avatars differentiable from movement that is part of the player's attempts to perform the hold? Maybe there should be a special move on the footpad that breaks you from all holds? Part of the grabbing involves creating joints between physics bodies, as we have seen. Those joints can be temporary and breakable in order to ensure that the player could remove themselves with enough repulsive force applied. We have yet to see whether this technique is tenable for players and if it creates frustration. Here there will be an obvious trade-off between 'stickiness' and player control.

Mutation

Reaching the end of a 'chain' of moves, even if that chain is only one or two moves long, would result in attribute swapping, and the avatars mutate. Deciding when and how to perform a mutation is another problem tied up with the general holds problem; e.g., when do you mutate if there are no set detectable end-points to holds? Difficulties such as this can perhaps be overcome again by viewing 'complete' holds as only one of many end-points, and calls for a more flexible, dynamic conception of what constitutes a hold. Coupled to this is the issue of how to help signify and draw the attention of the player to a mutation that has taken place. Currently after a certain amount of time in a hold end-point, a mutation takes place, forcing the avatars away from each other and simultaneously performing the mutation. Again, the end-point potentially causes problems here since players may never achieve that end-point hold and instead get halfway there or perhaps have combined several different grabs.

For the actual mechanics of mutation, there are various parameters that go to make up an avatar that may be swapped or adjusted, such as:

- Size, length of bones (which in turn determines the dimensions of the physics bodies);
- Position of bones within a body;
- Mass/density of bones;
- Centre of mass within a shape (a cube might be heaviest in one corner, for example);
- Various joint parameters, limitations; and
- Where joints are in relation to bones force applied to each bone when the avatar moves.

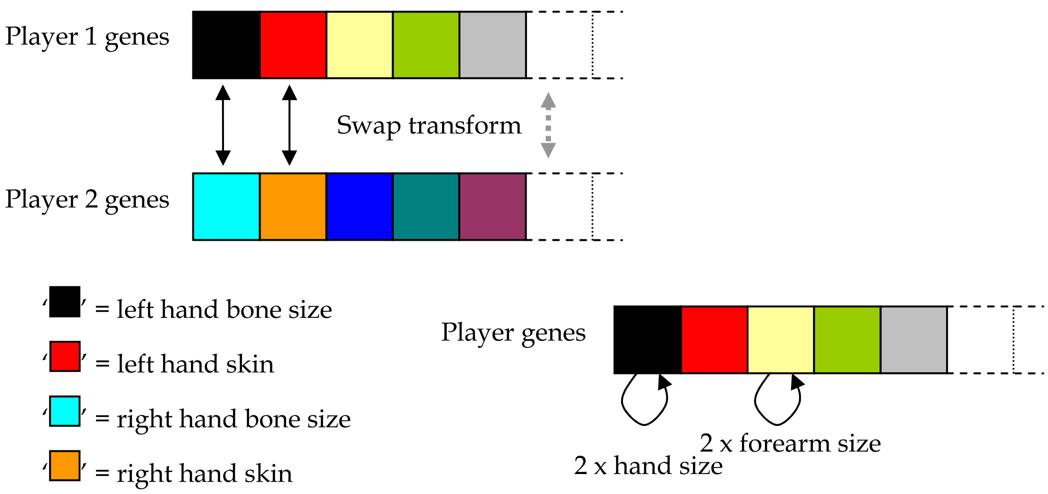


Figure 36. Simple example of gene swapping and mutation

The concept of 'avatar genes' was developed in order to talk about the way that mutation can be thought of as gene 'slots' being exchanged or modified in some way. Thus the conception treats all avatar characteristics as mutable parameters that develop with play, and interact with other player's avatars, resulting in a unique combination of initial personal preference (avatar as a generated object from the data body located in THEpUBLIC's database system) and in-game interactions.

Mutation may also occur without swapping, the special case being the growth of the avatar's torso in relation to the time which it has spent in the air. As detailed in the game proposal, avatars more easily fall to earth as they heavier and heavier, represented visually by their expanding girth.

Real and virtual cameras

The virtual camera's movement is instantaneous, and orientations set obviously have a negligible latency. In addition to this there are no restrictions on the path taken between start and end points. The real camera has particular physical attributes. It takes a significant amount of time to send instructions via RS232 to the camera (?s of a SECOND?), and the camera takes time to accelerate to achieve a constant speed, and then decelerate to zero speed (i.e., at its destination point):

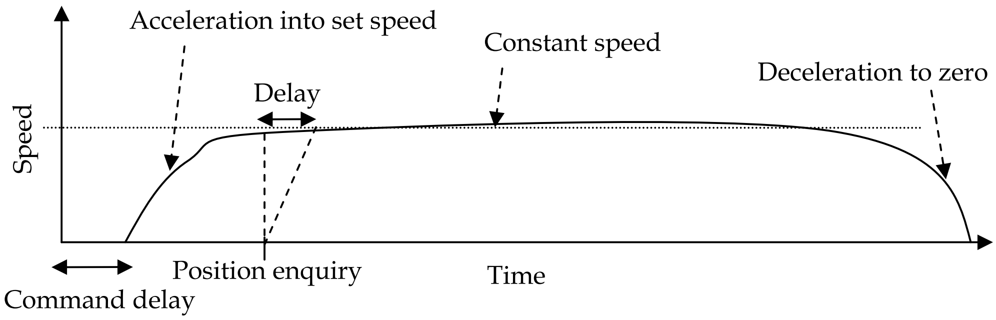


Figure 37. Reorientation of the camera time graph

Further to this, reorienting the camera to a particular point is a discrete affair involving a lower level of granularity than virtual camera reorientations. The Sony EVI D70 has 18 speed levels for pan and 17 for tilt. Settings for pan range are between `0xF725` and `0x08DB` with `0x0000` as centre (-2266 to +2267), giving a range of 4533 separate settings. Tilt range is between `0xFB70` to `0x04B0` again with `0x0000` as centre (-399 to +1200), giving a range of 1599 separate settings. Given that the pan range is ±170 degrees and the tilt range is -30 to +90 degrees, the resolution of movement is 0.075 degrees per step.

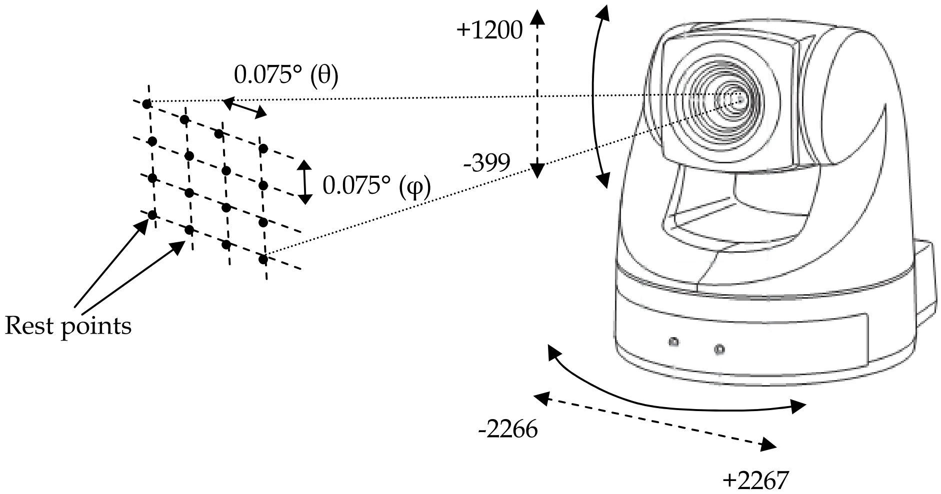


Figure 38. Sony EVI D70 motorised camera and the extents of its movement

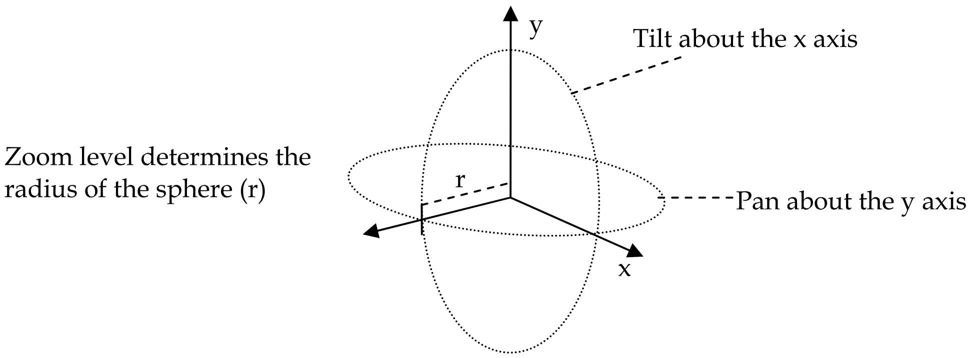


Figure 39. Pan, tilt and zoom

This stepping effectively creates a matrix of points that the camera can (theoretically) traverse between. There are limitations on how this matrix is traversed, and the camera can only move in eight separate directions. For those times when the $d\theta$ and $d\varphi$ of start and end (spherical) coordinates are equal (e.g., at an angle of 45 degrees), the camera will reach the end point evenly on both rotational axes. If $d\theta$ and $d\varphi$ are not equal, the smaller angle will be reached first. This basic motion restriction raises the issue of how create smooth or organic motions using the camera, or if it would even be possible at all.

The fixed speeds pose further problems since the speed at which an avatar moves inside the virtual environment is effectively a continuous value as opposed to the motorised camera's 18 discrete speed settings. Figure 40 (top right) shows an example avatar path matched against the camera's speed settings. It is clear that a level of mechanical smoothness is only possible to a limited degree. Further to this, there are conceivably regions where the avatar can move too quickly for the camera to catch up, or where the avatar moves too slowly for the camera to match an appropriate speed (i.e., it is either halted completely or moving at the slowest speed setting, which overtakes the avatar). It was thought that a possible solution might be to have the camera set to a higher speed than the speed required, then rapidly start and stop it to achieve the desired speed (Figure 41, bottom). Minimising the value of d would determine whether this was possible, however this approach also conflicts with some of the basic communication delay issues described.

The virtual and real cameras have vastly different capacities, however they are both being directed by the position of a completely virtual object (i.e., the avatar). These three elements (avatar, virtual and real cameras) are linked intricately; the avatar's movement essentially determines (or at lease heavily influences) the orientations of the real and virtual cameras.

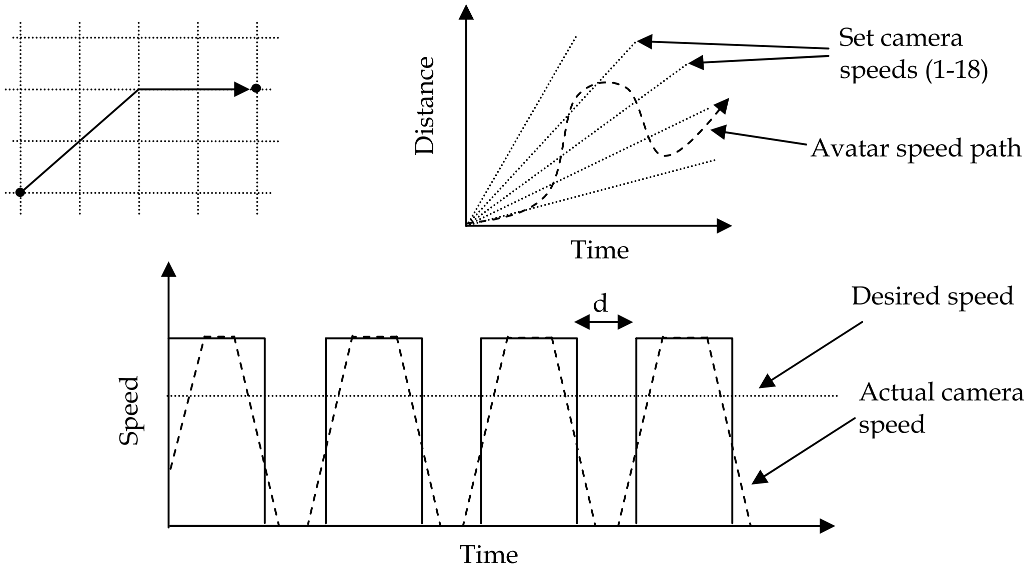


Figure 42. Camera motion from start to end (top left), the problem of discrete camera speeds (top right), and the emulation of arbitrary speeds

Relationships between cameras

The following diagram illustrates two different approaches to the way the avatar's motion may be associated with the motions of the cameras. The red arrows show avatar movements primarily determining the orientation of the virtual camera, which in turn instructs the real camera to match its movement as closely as possible. The black arrows, on the other hand, show avatar movements determining the orientation of the real camera, that then instructs the virtual camera as to its orientation. These are two extreme characterisations of ways of managing the both cameras. There are shortcomings for each method, however, which are the following.

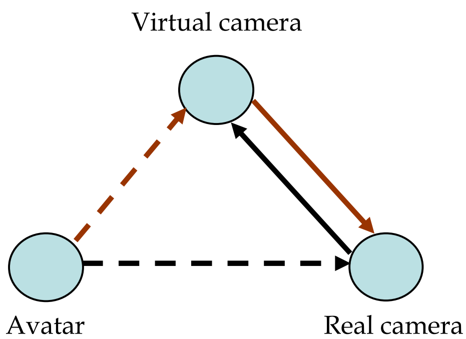


Figure 43. Relationships between avatar, real and virtual cameras

1. Avatar movement determining virtual camera orientation (red):

a. Orientation of the virtual camera is set according to some cinematographic logic based on the position and physical extent of the avatar. This position and logic might be based on a single central point of the avatar (e.g., focussing on the avatar's torso), or instead some point determined by several aspects (e.g., we might calculate the avatar's current height, projected onto the viewing plane). The virtual camera's orientation is therefore updated every frame according to this logic (this may be a static position, of course), and our simulation is typically running at around (? HOW MANY HERTZ? 50?).
b. The real camera is informed of each cycle and given a new orientation to move towards.
c. The virtual camera requests the current orientation of the real camera so as to remove any discrepancies between the two sets of orientations.

This approach is too reliant on sending information to and receiving orientation data from the motorised camera as though it can be polled at the same rate as the simulation. Whilst 'commands' for the virtual camera are executed instantaneously, testing has shown that the real camera becomes flooded with commands that it has not completed yet, and so lags far behind the motion of the virtual camera.

2. Avatar movement determining real camera orientation (black):
    a. The orientation of the real camera is set periodically (i.e., the camera is not flooded with commands). In order to do this, the position of the avatar cannot be reported constantly, so some notion of frame constraints must be introduced. When the avatar's position moves outside the frame, only then is the position and therefore orientation updated.
    b. The real camera moves towards the new orientations.
    c. The virtual camera requests the current orientation of the real camera, however, given that we cannot flood the camera with commands, this update must too be infrequent.

A problem is introduced this time when the virtual camera requests orientations from the real camera. As the real camera is reorienting, and as mentioned before, it takes time to request the current orientations during this time, and flooding the camera's buffer with requests for such orientation data also causes problems.

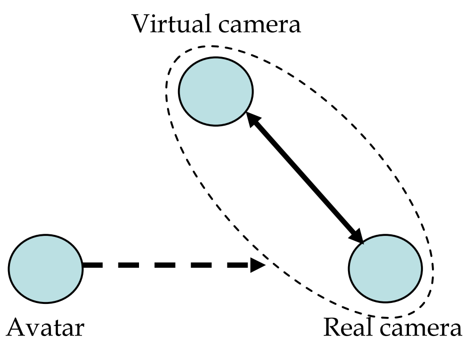


Figure 44. Relationships between avatar, real and virtual cameras

In order to overcome the problems associated with either method, a hybrid approach is required that treats virtual and real cameras as a tethered system with varying levels of tightness in the coupling. Two main issues, then, are interpolating and modelling motion, and managing camera updates.

1. Interpolation and modelling motion:

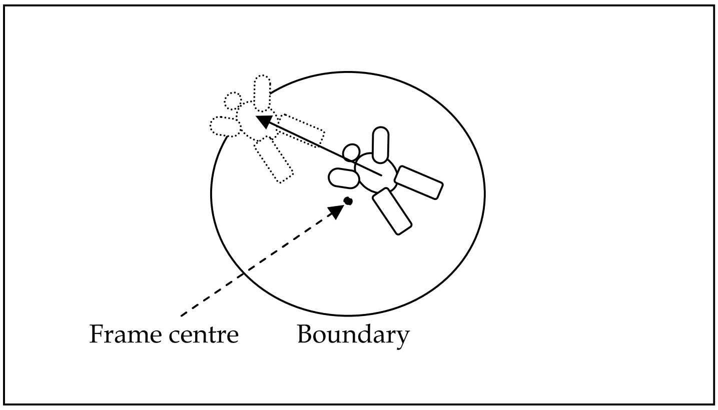
    a. Since the virtual camera cannot request orientation data frequently and yet needs constant updates so as to guard against skipping, the real camera control interface must be able to provide some interpolated response. This response would need to predict the current location of the camera between its start and end orientations.

    b. Beyond this is the modelling of acceleration, deceleration and command delay times. It is unclear currently whether assuming a basic plateauing function would be enough to model acceleration and its inverse, deceleration. In addition, it is also unclear whether command delays are predictable in time span and whether a simple dead reckoning measure based on a known delay would be flexible and reliable enough to compensate for this.

2. Managing camera updates:

    a. Reducing the number of updates and requests sent to the camera involves reducing the number of *required* updates for the virtual camera. If we reduce this number of required updates, the virtual camera simply must move less frequently.

    b. Using a basic 'cinematographic' logic like frame constrainers can reduce the required updates between real and virtual cameras. Currently at this stage, we have a simple boundary mechanism for determining when requests are made of the camera system. This is essentially a within-frame boundary as shown below.



Frame

Figure 45. Basic frame constraints

    c. The frame centre is set to the avatar's initial position. The position of the avatar may be anywhere within that boundary region, however when the edge is reached, the constraints system requests that the camera system move the frame centre to the new position of the avatar:

    d. The constraints system's requests are sent to the real camera each time a constraint is 'broken' (e.g., the boundary constraint being 'broken' by the avatar moving outside that boundary). These requests are essentially new directions that we need to make the camera point to (new 'waypoints'). The virtual camera of course has no physical attributes and so can get to the new spot to point to instantaneously. Given, however, that the real camera has some physical properties (such as initial starting time acceleration, time it takes to get to this point, etc.), the virtual camera needs to be frequently updated with the real camera's progress to that new point.

e. In this sense there is a tethering between the motions of the virtual and the real:

   i. There are infrequent updates are sent by the constraint system (i.e., the 'waypoints').

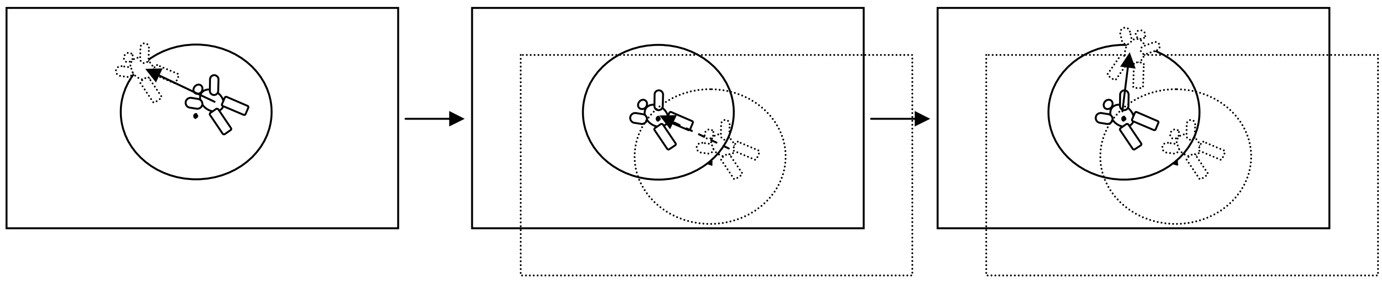


Figure 46. Frame constraints in action

   ii. The real camera subsequently provides frequent updates to the virtual camera on its current position, how far it is away from the new point, and so forth. This information then can be used to move the virtual camera in synchrony with the real camera's progress.

Even using a simple constraint system, the requirements on the real camera can be cut down drastically. The previous discussion neglects the zoom of the camera, which is a complicating factor in the constraints system.

It is also worth noting at this point that the movements of the avatars cannot easily be predicted, and thus a well-defined pathway for the cameras to follow is hard to construct. The frame constraint technique illustrated reduces complex avatar motion into a series of waypoints, shown at the top of Figure 48. The limitations of the camera mean that at best, this kind of waypointed motion is the most fine-grained that can be achieved, whereas the bottom half of the diagram shows a movement that is unobtainable.

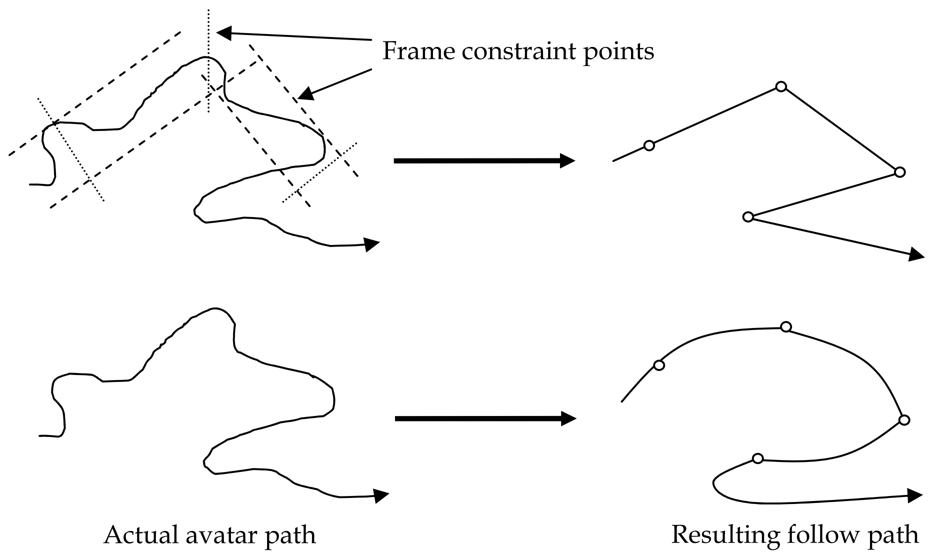


Figure 47. Constructing camera movement paths

Overview

Figure 50 provides an overview of the cycle of interaction, frame constraints, virtual and real camera tethering and the sequence/process by which each of these flows into the others.

The footpad provides some input to the physics engine. The physics engine then determines the

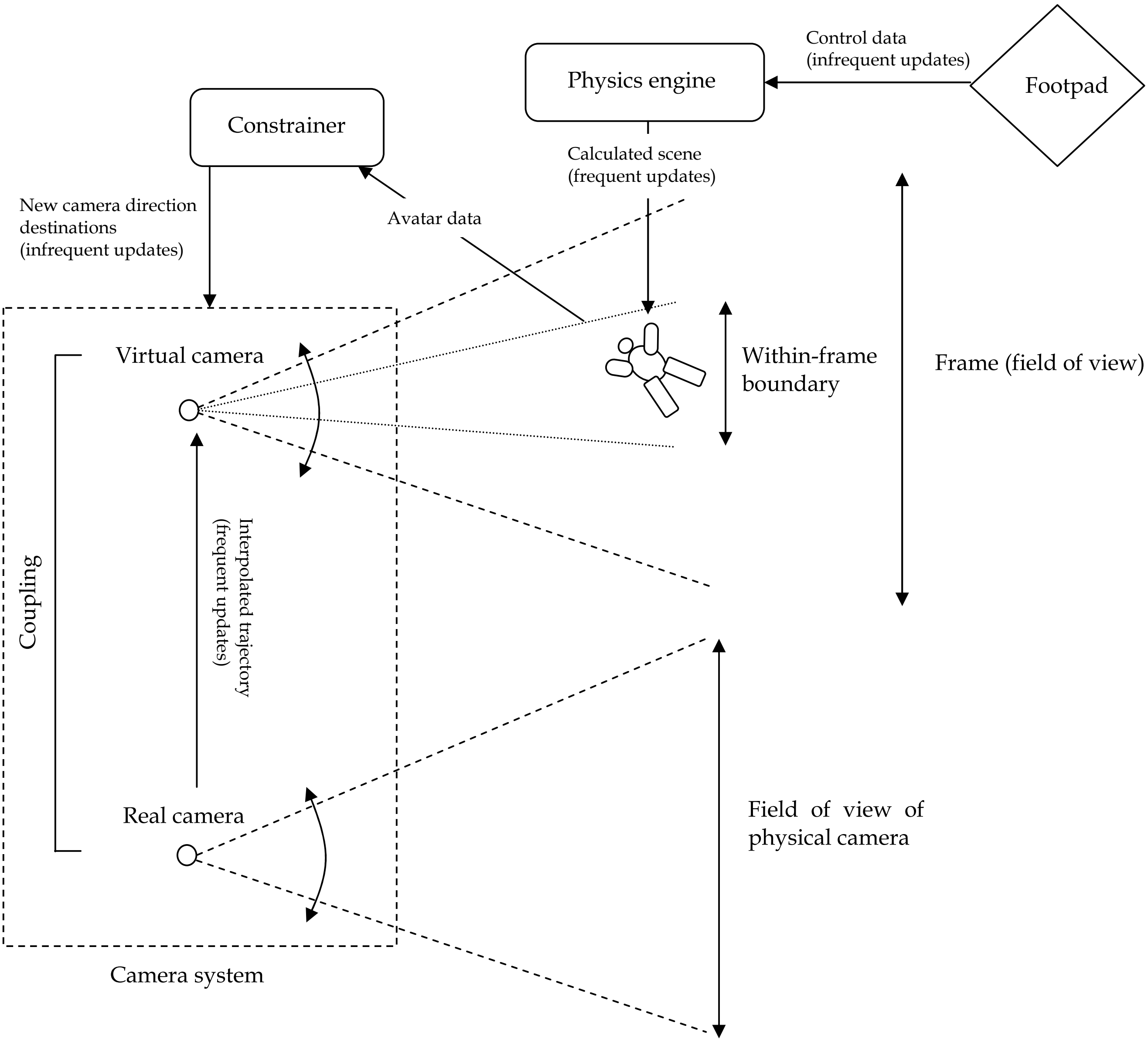


Figure 49. Schematic of the relationships between cameras and avatar

scene, and thus the position of the avatar(s). The cinematographic constraints on these positionings determine what requests are made to the camera system.

Further cinematographic issues

We have only considered the simplest of constraints. There are several examples of automated constraint and virtual cinematography systems, and whilst most of the literature seems to focus on purely virtual cameras with no restrictions upon their movement, some issues are relevant to the limited scenario in Flypad.

One system provides authoring tools for a virtual camera shot constraint solver (Bares, 2000). Particular requirements for shots are authored, and subsequently processed to produce a solved path/position for the camera. Some of these constraints were:

- Prescribed maximum/minimum size of actors in-shot;
- Tolerable/desirable levels of occlusion between actors;
- Permitted positions for the camera (e.g., solutions to camera shots inside a room have to stay within the volume of the room);
- Which actors should be in-shot and which should be out-of-shot;
- Camera field of view; and
- Type of camera movement (e.g., pan, translation, etc.).

(Liu, 2001) on the other hand features a system automating the cinematography of a lecture room environment. The system therefore has no virtual aspect, yet has elements of motorised camera management, covering the framing of the lecturer, editing the shots, durations, cuts between cameras and provides some guidelines for tracking the speaker (such as not moving the camera with every movement of the speaker, trying to reduce movement, giving lead room of gaze direction or head orientation, and leaving at least half a head of room above target).

(He, 1996) also covers some camera placement issues, constraints, shot sequences and so forth:

- Camera placements include issues about the "line of interest" (i.e., a line between actors), internal, external and apex shot positionings:

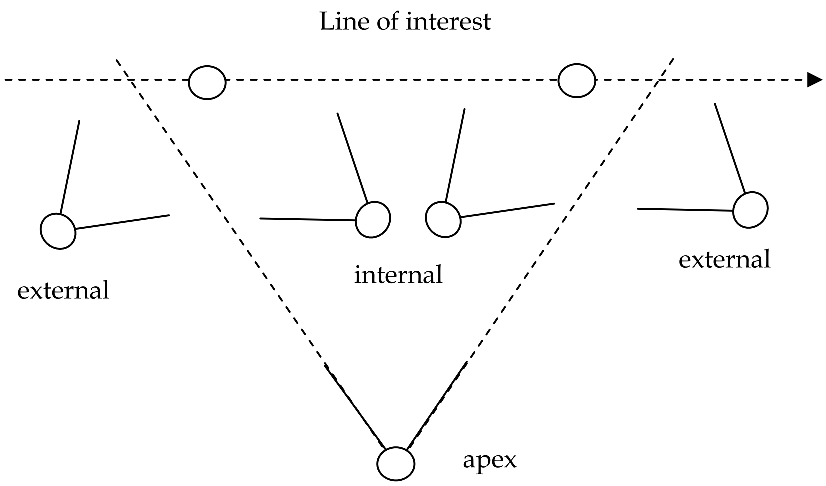


Figure 51. Line of interest and shot positioning

- Cinematographic heuristics and constraints include things like:
  - Don't cross the line of interest;
  - Avoid jumps in cuts between shots (making a marked difference between size, view or number of actors in cuts between shots);

- Let the actor lead (the actor initiating all movement of the camera and the camera coming to rest before the actor finishes movement).

Also discussed are types of shots (e.g., track, pan and follow). It might be useful to consider, given this research, the implications and the affect upon play of having only one kind of shot in which the camera can only be panned.

With these concepts in mind, the Flypad camera system has some of its degrees of freedom removed; i.e., the motorised camera has pan and tilt rotations and a zoom function. In addition to this, we have a problem of matching physical movements with virtual. The set of cinematographic requirements is a special case of some of these other systems, in that the camera has a fixed position (i.e., looking into the space from the mount) plus a zoom function. The problem is novel in that a mutual tethering of virtual and real exists in which information to and from each camera influences the movement of the other. Typically cinematographic requirements are only made exclusively of a real camera, or a virtual one.

1. Occlusion presents us with problems in that occlusion in the real world has no mapping to the virtual world. We have (currently) defined a volume of "safe" space in which avatars may exist; avatars cannot move outside those bounds since such movement might break virtual/real mappings because a virtual object cannot (currently) be occluded by a real object in the space such as a pillar. As such the solution to this mismatch between the virtual and real, is the restriction of the virtual to "safe" space.

2. The focal length of the camera may be an issue. Since our camera has two degrees of freedom, and zoom will create distortion, we need to

Currently we have considered constraining real camera movements to virtual camera movements (and vice versa), but we also might need to constrain avatar motion by the motion of the camera. For example, the camera will perhaps not behave smoothly if avatars collide with any violent impacts, even if we place constraints on framings etc. In this sense the avatar motions might need to be slowed down to a certain "impact speed" before their collisions.

We could use a constraint-based approach that specifies a set of restrictions and then solves the camera movement for those constraints. Constraints could be, for example:

- Target must stay wholly within a particular area of the frame
- Target must stay wholly within the frame
- At least one nominated part (in film, often the head/upper body) of the target must stay within an area of the frame or the entire frame

Essentially we are dealing with two types of constraints:

1. Framing constraints; and
2. Motion constraints.

In order to make camera motion more organic, we might like to use a movement algorithm that features some spring and damper effects. Obviously we are quite constrained by the physical camera in this case, however we might broadly be able to manage any rapid changes in direction with this technique. It all depends upon the way in which the avatars move.

- Tethering real and virtual cameras
  - apply cinematographic principles to compensate for physical camera limitations
  - The avatar, the real camera and the virtual:
    - Fundamental problems with linking real and virtual, and the nature of avatar movement

        - Coupling the real and virtual tightly
        - Loosely coupling real and virtual

Graphics

Meshes can be 'draped' over the armature of the physics bodies best as possible to correlate the volumes of the physics bodies with the structure of the mesh.

- Gloss without looking dated
    - skinning
    - use visual effects to imply depth (dof, shadows)
    - HDR effects

References

William Bares, Scott McDermott, Christina Boudreaux, and Somying Thainimit. Virtual 3D camera composition from frame constraints. In *MULTIMEDIA '00: Proceedings of the eighth ACM international conference on Multimedia*, pages 177-186. ACM Press, 2000.

Qiong Liu, Yong Rui, Anoop Gupta, and J. J. Cadiz. Automating camera management for lecture room environments. In *CHI '01: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 442-449. ACM Press, 2001.

Li wei He, Michael F. Cohen, and David H. Salesin. The virtual cinematographer: a paradigm for automatic real-time camera control and directing. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 217-224. ACM Press, 1996.

Sandy Ressler, Brian Antonishek, Qiming Wang, Afzal Godil, and Keith Stouffer. When worlds collide - interactions between the virtual and the real. In *Proceedings of 15th Twente Workshop on Language Technology: Interactions in Virtual Worlds*, May 1999.

Stuart Reeves, Steve Benford, Claire O'Malley, and Mike Fraser. Designing the spectator experience. In *Proceedings of SIGCHI Conference on Human Factors in Computing Systems (CHI)*, April 2005.