

# ZENITH

Version 2.0  
17th March 1995



DMA Design Ltd.

# Contents

Introduction.....	1
The Setting (Background).....	1
Flow of the Game.....	2
Game Elements.....	3
Worlds.....	3
Courses.....	3
Battle.....	3
Sprint.....	3
Endurance.....	3
General.....	3
Sections.....	3
Boundary Regions.....	4
Tiles.....	4
Tile Geometry.....	4
Tile Textures.....	4
Blocks.....	4
Block Geometry.....	5
Block Textures.....	5
Objects.....	5
Static Objects.....	5
Dynamic objects.....	5
Characters.....	6
Geometry.....	6
Texture.....	6
Ratings.....	6
Character Groups.....	6
Teams.....	7
Ratings.....	7
Actions.....	7
Climbing.....	7
Falling.....	7
Fighting.....	8
Statistics.....	8
Primary Stats.....	8
Secondary Stats.....	8
Character Development.....	8
Character Selection and Retiring.....	9
Game Types and Display Modes.....	10
Time Trial.....	10
One-on-One.....	10
Tag.....	10
Knockout.....	10
League.....	11
Multi-Player.....	11
The League.....	12
Divisions.....	12
Ratings, Points and Times.....	12
Extreme Zenith.....	13

User Interface (Front End) .....	14
Global Functions.....	14
Select.....	14
Exit.....	14
Abort .....	14
Change Selection.....	14
Menu Hierarchy .....	15
Fundamental Menu.....	15
Create Team Menu .....	15
Review Team Menu .....	15
Play Menu .....	15
Game Controls.....	16
Appendix A - Original Design Drawing	
Appendix B - Illustrations	
Appendix C - List of Character Animation Sequences	

## Introduction

As you may have guessed from the title, this is a game where the objective is to get to the top. The player controls a humanoid character which walks, runs, jumps, climbs and also has special moves for fighting and acrobatics. The character, level and any enemies are represented as 3D polygon models, allowing a wide variety of camera and lighting effects. The general look and feel of the game will be faintly cartoon-like and very dynamic. The objective is to make the game visually pleasing and responsive to play; realism is not an issue.

## The Setting (Background)

This game is set in the not so distant future when the greatest danger faced by humanity is boredom. Technology has answered the everyday needs of the population; food, clothing and shelter are available for everyone. Machines carry out all the mundane tasks (washing socks etc.) leaving people with much more free time. Some of the population obtain employment in the huge corporations which dominate the cities; this is for prestige and status rather than money. Others absorb their time in personal study or take up arts and crafts, while some are interested only in a life of leisure. Inevitably, there is also a sub-culture of individuals who don't fit into the new order. They pass their time with little regard for the rules or morals of society as a whole; many are part of the ubiquitous street gangs, or are otherwise affiliated to the criminal underworld.

Sport is enormously popular and the vast majority of people are either spectators, participants, or both. Many new sports and games have been devised, although most are simply variations on older ideas, made more entertaining by new technology. Sport (and life in general) is much safer than before; injuries are rare and can always be completely cured. There are, however, some people who enjoy the risk of injury or death and so deny themselves the benefits of technology. The result is extreme sport which, although officially discouraged, has a widespread following.

Zenith is one sport which is universally popular, having both tame and extreme versions. In the tame version the participants are equipped with harnesses (suspensor fields) that limit falling speed and cushion impacts. This means that falling a great distance will do no more than lose the player any height advantage and waste time. The extreme version offers no such benefits; a lengthy fall in this case is liable to result in the despatch of a cleaning drone equipped with mop and bucket.

There is a league table for the tame version of Zenith (henceforth League Zenith), this operates on a points scheme much like other sports. A comprehensive league table will be stored and maintained in non-volatile memory (assuming this is available). As well as the league table, the ten best times for each course will be stored.

League Zenith is covered by network television and takes place on huge brightly coloured structures designed for the purpose, each of which has a particular style or theme. These structures are described as worlds (think of theme parks) and are divided into courses of varying difficulty. Although there is no definite constraint on the layout of a world, the easy courses will tend to be on the lower reaches of the structure; in other words, difficulty will progress steadily with height.

The extreme version of Zenith takes place at night on any suitably challenging building. These are often corporate headquarters equipped with efficient security systems designed specifically to stop such incursions. The appearance of these buildings and their robotic guards is functional, dark and menacing. This sport has a huge cult following and amateur video footage is often featured on public access channels; network TV does not cover such events for legal reasons. Extreme climbs are arranged by underworld organisations who shield the climbers from prosecution (and persecution) in return for certain benefits: gambling, whether legally sanctioned or not, is as popular as ever.

Extreme Zenith cannot be selected by the player as an option (at least initially). Rather, the extreme courses are the bonus or "hidden" levels which will be offered to the player once they have achieved sufficient standing in the league. This offer will be made by an underworld



contact who approaches the character in person or by video mail and makes a hard to refuse offer. This can be shown as an animated sequence (rendered if appropriate) and hence offers immense scope for creative scripting and design. Once an initial offer is accepted, more regular offers will be made with suitable incentives to keep the player hooked.

## Flow of the Game

On power-up the game will go into the usual sequence of license and advertising screens followed by a brief introductory movie. This will be a 3D sequence rendered in real time and should depict an important set of events from the cameo story. It is expected that this movie will not last much longer than 30 seconds due to storage limitations; in any case the user will be able to bypass it and get on with the game.

The first thing a player must do is choose a character to play; either by taking one of the standard ones or by loading in a customised one (from battery backed RAM or similar). At this point the player can review the skills and abilities of the selected character while viewing an animated polygon model. This model will use the entire rendering capacity of the console (2000 textured, lit polygons) which will give a truly stunning effect. If desired, the player can flip to another screen which displays the characters league rating and times on different courses (a great many screens will be require for all courses).

Having selected a character, each player must select a game type (see later) and a course to play on. League games will require a particular sequence of courses however there will be a facility to play on almost any course for practice or two player head-to-head games. The first step in selecting a course is to choose one of the worlds. The world selection screen will present the worlds as small rendered images (all on a single screen); the user will move a cursor or highlight region to the desired world and select it. This brings the user to the world preview screen which shows the world as a moderately detailed polygon model (enough to be able to pick out the individual courses). Courses are then selected using a pointer/highlight scheme as for the world selection screen which brings the user to the course preview screen (phew, got there at last). Courses are previewed as polygon models (with the ability to zoom in and out) which will help the player in selecting the best route for their character.

Once everything (game modes, course and characters) has been selected the player starts the game at which point the view cuts to an aerial camera positioned above the course. A fairly fast (2 or 3 seconds) pan down to the start of the course takes place, accompanied by commentator voice-overs, crowds cheering and other atmosphere building effects. Play starts without further delay, accompanied by a huge roar of approval from the crowd.

During a climb the camera tracks the character, moving only slightly to simulate drift of an airborne vehicle. When the distance between the camera and the character exceeds some pre-set limit, a new camera position is selected. The camera travels to this new position in a short period of time with a smoothly damped motion. If the character is moving rapidly (such as during a fall) the camera will lock to them, maintaining viewing distance and angle (although not completely rigidly).

Play normally ends when the players have reached their goals or have been incapacitated by a severe impact (extreme version only). If a time limit expires then the player is allowed to complete the course for practice, or they can abort play. If a player aborts before the time limit expires, this will be considered an automatic defeat for league table purposes. Also, aborting just because you're not doing particularly well is rather unsporting; this will trigger jeering from the crowds and a suitable "one-liner" from the commentator (nobody likes a quitter).

# Game Elements

## Worlds

There will be a minimum of five worlds included in the game each of which will have an individual graphic style. The present short-list of styles is: Plastic, Oriental, Roman, Inca, and Egyptian. These are to be interpreted as guidelines only and should not restrict artistic creativity in any way. These worlds are used only for League Zenith, the extreme levels are played on more conventional structures resembling office blocks (albeit with a 1920's Gothic bias).

A world consists of many courses, all of which use the same set of blocks and textures (i.e. each world has blocks and textures unique to it). There will be a total of 256 textures per course and 256 block types per course. The textures may be either colour or intensity format and are limited to 16 colours or 16 intensity levels respectively. The colour textures must use one of 16 colour palettes of 16 colours each; these will typically be unique to that world.

*To be decided:-*

- *Approximate number of courses per world.*

## Courses

The majority of courses will consist of a single section, however, it is possible to design multi-section courses. Sections will normally be at right angles to one another with a boundary region between the two. Transit between sections is provided by bridges made out of blocks; there is no other way to cross a boundary region.

### Battle

This type of course encourages combat by restricting the players paths so that they come into contact regularly. This can be achieved by using lots of fixed and moveable blocks; a moveable block which obstructs progress has to be pushed out of the way by the first character to get there. This gives the other character time to catch up so that combat can take place.

### Sprint

This is a fast race-type course, best suited to characters with a high quickness stat. It will feature lots of open areas that allow the character to run quickly, also collapsing blocks and wide gaps to jump will keep play fast paced.

### Endurance

This type of course is designed to test the endurance of the player as well as the stamina of the character. It will contain many difficult sections such as overhangs and vertical areas; frustration value can be added by using areas that are likely to make the character fall off. These would result in a short fall that forces the character to go round again.

### General

This is a course for all-rounders and contains a balance of features found in the other three course types.

## Sections

A section is composed of large angled surfaces which are subdivided into square tiles. Characters can stand upon these tiles or climb up them, depending upon the type of the tile (some tiles may be slippery so the character will slide off them). To add variety, blocks will be attached to some tiles. Characters interact with blocks according to the block type; most blocks can be climbed or stood upon. Refer to the original design drawing for a better idea of how this works.

A section will be no larger than 32 blocks wide by 256 blocks high which corresponds roughly to 2 by 16 screens. This is expected to result in a playing time of between 3 and 6 minutes depending upon the skill of the player.

## **Boundary Regions**

These are areas of arbitrary polygon geometry which fill in the gaps between sections. Because they are not composed of tiles it is not possible to process collisions efficiently in boundary regions. Therefore characters are forbidden from crossing boundary regions except by using bridges made from blocks.

The difference between a boundary region and its neighbouring sections will be clearly indicated by the colour scheme of the boundary region. One possibility is to render the boundary region polygons in a colour which indicates danger (red being the obvious choice). Little need is seen for textures on boundary regions however these can be used if necessary; animation of colour and/or texture could give a good visual cue as to the difference between boundary regions and sections.

*To be decided:-*

- *What happens when characters attempt to cross a boundary region?*
- *Appearance of boundary regions.*

## **Tiles**

Tiles are square and form the basis of the course: a course could be composed solely of tiles, although this is unlikely to make for interesting game-play. Each tile has a type which determines its appearance and properties (rough or smooth and so on). Characters and moving objects (such as falling rocks) collide with tiles and behave in a realistic manner; the bounce velocity of an object can be determined from the angle of the tile and its surface properties.

### Tile Geometry

Tiles are simply square areas in 3D space which join perfectly with each other along their edges (i.e. they cannot intersect). Tiles are constrained to have a constant area which simplifies texturing but complicates collision detection (tiles do not exist within a regular 3D grid).

### Tile Textures

The appearance of a tile is determined by the intensity map applied to it and the material from which the course is made (the material and light source combination determine the shade colour). The tile intensity maps are part of the total texture budget for a world, the majority of which is required for blocks. A good rule of thumb would be to allocate 32 intensity maps for the tiles, leaving 224 colour textures for the blocks (a ratio of 1:7).

## **Blocks**

Blocks are placed upon the tiles of a course to add interest and variety to the game. Blocks, like tiles, have a particular appearance and a set of properties although these are generally more complex than those of tiles. Each block is attached to a particular tile but need not be directly adjacent to or intersecting that tile; an offset allows the block to be positioned with a fair degree of flexibility.

### Block Geometry

Blocks may have free form geometry, however if the geometry is particularly complicated then collision detection will use an approximation. The block designer should bear this in mind to avoid visually obvious collision detection errors. This is not relevant to all types of block; some blocks, such as an air vent which directs a continuous blast of air upwards, or a clump of foliage, will require specially written collision code in any case.

As mentioned above, blocks are a property of a particular world; any world is limited to 256 different designs of block.

#### *To Be Decided:-*

- *Upper limit on number of vertices and surfaces per block*

### Block Textures

The majority of textures placed upon blocks will be drawn in 16 colours, although they could just as easily be intensity maps (which are used for tiles). These textures are allocated out of the total budget for a world: 256 textures for all tiles and blocks. As mentioned previously, a ratio of 1:7 for tile to block textures is thought to be about right (which means 32 textures for tiles and 224 for blocks).

Blocks can freely use any of the 256 textures allocated to a world; this is at the sole discretion of the block designer.

## **Objects**

Enemies and any other animating features such as moving platforms are classed as objects. An object may have animating geometry or texture maps (whereas a block is static); this is the basic distinction between a block and an object. Objects are relatively simple polygon models, certainly no more complex than one of the character models (200 polygons, 16 components). Objects are subdivided and classified according to certain properties:-

### Static Objects

Static objects are those which do not move by themselves but may be moved through character intervention. An example is a rock which could be pushed off a ledge, falling onto an opponent below. Refer to figures 13 to 16, 22 and 23 in Appendix B.

### Dynamic objects

Dynamic objects are capable of moving by themselves. Their movements may be determined procedurally or can simply be cyclic paths. Enemies fall into this category (procedurally determined motion) as do lifts and moving platforms (mostly cyclic paths). Refer to figures 17 to 21 in Appendix B.

## Characters

There will be 8 basic characters for the players to choose between when they first start playing the game. Each will have particular strengths, weaknesses and special abilities as well as different appearances, in order to provide the widest possible appeal. The breakdown of character types should be approximately 50% male, 25% female and 25% androgynous (robots or aliens). See figures 24 and 25 in Appendix B for character designs.

### Geometry

The character models are given an upper limit of 250 polygons to use; this is an absolute maximum and cannot be exceeded under any circumstances. The basic models are allocated roughly 200 of these polygons which leaves another 50 to add fine details and individualise the characters.

There will be a minimum of 8 of these models, however, if possible more models will be included. This is likely to mean that the same basic 200 polygons are used for many models, but each can use the remaining 50 polygons to provide some individuality.

### Texture

The texture maps for characters will be drawn in 16 colours. An area of 64 by 64 pixels is available for character texture maps; if this is inadequate then an additional 64 by 64 pixel texture palette will be available.

### Ratings

Each character has a rating determined by the number of points scored in all the types of game during that character's career (i.e. since they were selected as part of a team).

## Character Groups

To improve variety in character selection, the 8 basic models will each have several different sets of texture maps. It may also prove possible to vary the model geometry to a limited extent; the problem with this is that the models within each group must use the same animation's (this is due to the sheer volume of animation data). Consequently, any variations in geometry must be purely cosmetic; the proportions of body components must be constrained so that movement looks sensible.

This means that there are 8 groups of characters which use the same basic animation data (walking, running and so on). However each character is unique in terms of their balance of stats, selection of special moves and general appearance.

The characters within each group are graded from basic to elite according to their skills and abilities. Initially, a player may select only the basic character within each group but as they progress through the league and improve their rating, more advanced characters can be chosen. This provides an extra incentive to progress through the league and increases long term interest in the product.

The advanced characters should not be vastly superior to the basic ones; they should provide variety rather than an overwhelming advantage (the majority of any advantage should come from the skill of the player).

*To be decided:-*

- *How many different character designs: 32 , 40 , 48, 64?*
- *How many special moves per character : 4?*

## Teams

The player sets up a team selected from the available characters for use in the league matches. These characters are transferred to non-volatile storage so that they may be developed and customised by the player (see Character Development). A team consists of 8 characters so when a player starts of a new team they simply acquire copies of the basic characters from each group.

Each team is given a name defined by the player, subject to the usual restrictions (no obscenities). Teams are also identified by a bright and bold team logo (decal) which is chosen by the player from a menu (this makes each team easily recognisable). Each character displays the team decal on their back; this makes them easily recognisable from the rear. In order to improve recognition of characters from any viewing angle they can be given a colour-coded harness. If the harness were made from coloured polygons (rather than part of a texture) then it would be a simple matter to allocate a different colour to each player.

If a team member expires (extreme matches only) then a new team member must be selected. This might be one of the standard 8 characters provided that character has not already been selected. If the team has been doing particularly well then the replacement could be one of the more advanced (perhaps even elite) characters.

### Ratings

Each team is rated according to the performance of the individual teams members. This rating can be based on the top four characters in the team, or some other weighting could be applied.

*To be decided:-*

- *Number of characters per team (4?)*
- *Number of decals available (64?)*

## Actions

### Climbing

The player climbs by directing the character in the desired direction, which is normally either up, down, left or right. On occasions when the character cannot climb up or down because they are not adjacent to a wall, they will move into (further from the viewer) and out of (closer to the viewer) the scene respectively.

Depending upon which block type the character encounters, an appropriate animation will be played. Blocks will be categorised into groups having approximately the same geometry and a single animation used for the blocks in each group. This will keep the number of animation sequences within reasonable limits.

### Falling

The character can fall if they mis-judge a leap over a gap, or spend too long hanging from a ledge or clinging to a sheer rock face. Characters can also fall if dislodged by a competitor or enemy.

The main disadvantages of falling are that you end up back where you started, it wastes time and, in the extreme version of the sport, it can end your career. Characters can attempt to grab hold of surfaces and so stop a fall. The chances of success are determined by the reactions of the player, the skills of the character and the speed of the fall.

## Fighting

Combat is possible in most types of game so every character will have a few simple fighting moves. Some characters will specialise in combat (which makes them less proficient at climbing) and so will have more combat moves to choose from.

Because of animation limitations, complex fight sequences will be restricted to occurring on ledges or other flat areas. It is expected that simple punch or kick moves could be performed while hanging onto walls.

## **Statistics**

### Primary Stats

Primary stats are visible to players and determine the performance of a character during play.

<i>Strength</i>	Brute musculature and an appropriately sized frame to support it. This determines the damage given to an opponent in combat and the ability to push moveable blocks etc.
<i>Quickness</i>	Speed of movement and general agility. This determines top running speed, distance and height jumped, and the ability to dodge or parry an opponents attacks.
<i>Stamina</i>	Endurance, resistance to pain and fatigue. This determines the ability to maintain difficult positions such as hanging from a ledge.

Each of these stats will have values in the range 1-200, 100 being the normal human maximum (this can be exceeded by exceptional characters or special bonuses). Certain enemies may drain stats (by either proximity or touch) and there will be the occasional bonus object which restores or boosts stats. Stats will be returned to their normal level at the end of each stage.

*To be decided:-*

- *Display primary stats as bars on screen during play?*

### Secondary Stats

Secondary stats govern how the computer simulates each character and are thus only relevant to computer opponents. They are therefore not visible to the player but they can be discerned through playing the game (which, after all, is the whole point).

<i>Aggression</i>	How likely the character is to attack an opponent. This is basically the distance at which an attack sequence will be initiated. An extremely aggressive character will home in on an opponent, possibly ignoring opportunities to get to the goal.
<i>Attack</i>	Ferocity of an attack sequence; the time between attack moves.
<i>Defence</i>	How good the character is at blocking attacks; reaction time.

## **Character Development**

The game should be straightforward and easy to get into; we are looking for mass market appeal. Consequently, a sophisticated RPG style character development system is felt to be a bit over the top. We don't want to put anyone off playing because they have to spend 20 minutes generating a character in order to play. However, there is also the issue of long term interest; after a month or two of play the typical gamer will be fully conversant with the control method and features i.e. boredom is beginning to set in.



We really need to provide more depth in order to make this a top notch product, but in a carefully controlled manner. An attractive solution to this problem is to make use of the extreme events to provide the character development feature. Thus, the incentive to take part in an extreme event would be the award of skill bonuses or special abilities. This can be rationalised as the gift of cybernetic implants or modification of genotypes using retro-viruses. In most cultures this type of thing would be viewed with disgust and horror (particularly genetic alteration) and is thus likely to be illegal or at least strictly controlled. This would explain why the underworld agents could offer such services (illegal back street gene labs). This would undoubtedly be attractive to many members of the population; who can honestly say they would turn down the opportunity to become a demi-god?

This strategy also gets round the problem of arbitrary skill development which implies problems for play testing; in order to be absolutely certain that the game will always behave correctly we need to test all possible combinations - this is clearly not practical. The bonuses for each extreme match can be pre-determined which allows the permutations to be carefully tested. If some "randomness" is desired then bonuses could be selected from a limited set of possibilities based around some statistical distribution.

#### Examples of Bonuses

"Grip of Iron"	+20 points to Stamina
"Methuselah"	+10 points to each Stat
"Touch of Pain"	Double damage for each strike on an opponent

These bonuses are permanent i.e. they remain with the character until he or she dies or is retired. A character can only die if they make a big mistake during an extreme match and fall a great distance (level design can ensure that this doesn't happen too often). Retiring happens when the player is bored with their character and resets them to the default status, ready to start all over again. This is based on the assumption that when the player selects a character for customisation, the character data is copied to battery backed RAM and modified as necessary. The original character templates will always exist in ROM and can be re-instantiated as required.

### **Character Selection and Retiring**

When a player decides to start up a new team, that team starts off in the most basic division with the most basic characters; the team and individual ratings are all set to zero. The selection process for teams uses the general character selection method; this is yet to be decided but the options run from simple bitmap "point and click" to a virtual environment with animating models.

The line up of a players team may change during a league as a result of termination during extreme matches. The characters available at any point depend upon which division the team has reached. It is also possible to allow the team line-up to be changed at the discretion of the player i.e. characters could be "retired". The question is; should some limitations be imposed on this to prevent continual chopping and changing? This becomes more important when the advanced characters within each group become available as a result of teams progressing through divisions. We should avoid characters changing their entire line up as soon as some slightly better characters become available.

*To be decided:-*

- *Character selection method.*
- *Can we control/limit character selection (points scheme)?*



## Game Types and Display Modes

There are two types of display; full screen and split screen. The full screen is used when only a single human player is on-screen such as during a player versus computer match, thus computer controlled opponents do not always appear on screen. The split screen is used only for a match involving two human players and gives the direction of each player relative to the other (this is difficult to explain; refer to a diagram).

There are four types of game available to choose from; some are for only one or two players whereas others permit a very large number of players (but not simultaneously).

### Time Trial

This is the basic single player game and will be used for gaining practice on whichever courses the player has access to. The objective is to make it to the goal within the time limit and to beat your own personal best time and the course record. If any best times or records are beaten the player will be congratulated (play a short "glory" animation or movie) otherwise they will be encouraged to try again. In either case the player is offered the option to play the same course again or select a new one. This game type is restricted to the full screen display mode; refer to figure 3 in Appendix B.

### One-on-One

This can be played against a fellow human (which uses split screen mode) or against the computer (which uses full screen mode). Essentially, the first one to make it to the goal is the winner. There is no time limit, although best times are recorded as usual. This style of game has great potential for combat; stunning your opponent or pushing them off a ledge could gain a valuable advantage.

A variation on this game is tag. This involves the players taking the roles of evader and pursuer; the player with the best rating will start off as pursuer. The evader starts with a small time (or distance) advantage and has to make it to the next stage without being tagged by the pursuer (no time limit). When one player tags another, their roles as evader and pursuer are reversed. The new pursuer is stunned for a short time, giving the new evader a chance to escape.

### Tag

This is a variation on the One-on-One type of game and uses identical display modes. The game play differs in that the players (human or computer) take on the roles of evader and pursuer. The evaders objective is to make it to the next stage, while the pursuer has to tag the evader. When the evader is tagged the characters roles as evader and pursuer are reversed. The new pursuer is momentarily suspended (3 seconds should be sufficient) allowing the new evader an opportunity to escape.

### Knockout

This permits a large number of people to play against each other, although no more than two people may play at the same time. The display type (full or split screen) is determined in exactly the same way as for the One-on-One game. Three sizes of knockout game are available: 8, 16, and 32 player. These can accommodate up to those numbers of human players or, if there aren't enough humans, the numbers are made up with computer players.

Matches are played between pairs of players, the loser being knocked out and the winner progressing to the next round. In the case of the 32 player game this results in 16 matches in the first round, 8 in the second, 4 in the third, 2 in the fourth and one (the final) in the fifth round.

### League

This is the long term game type and the only way to progress to new sets of courses. It is suitable for a large number of players, although obviously only 2 may compete against each other at any one time. Like the other game types, human versus human uses the split screen display and human versus computer uses the full screen.

Like any other league system every competitor is required to play every other competitor. Clearly if the number of competitors is large then this results in a huge number of matches for the initial rounds. A desirable option to accelerate play is to allow matches from a single round to be played simultaneously on different consoles. This would be attractive to a group of, say, 8 friends who have access to 2 game consoles. The main problem with this is transferring information between machines i.e. some form of exchangeable storage device is necessary.

The league is divided into 10 divisions, division 1 being the elite and division 10 the amateur. Each division has 4 courses on which all matches are played. These courses will each emphasise a different aspect of the game; speed, strength, stamina and all-round ability.

*To be decided:-*

- *Maximum number of players in a league?*
- *Concurrent play of matches in a league (requires exchangeable storage device)?*

### Multi-Player

This allows up to four players to compete simultaneously using a full screen view. This clearly has problems with rendering load as four character models will be in view simultaneously, so sacrifices will have to be made in other areas. The obvious solution is to reduce the complexity of the map however this means having a different type of level i.e. different level format and game code to drive it as well as a different level editing process.

*To be decided:-*

- *Organisation of multi-player levels*

## The League

This is the most important game mode and provides the long term interest in the product. Only by competing in the league using a team can the player gain access to higher divisions. When a team has access to the higher divisions, the more advanced characters within each group may be chosen.

### Divisions

The league consists of 10 divisions, each of which has courses dedicated to it. The divisions are numbered in descending order i.e. division 10 is the easiest and division 1 is the hardest. Progress from one division to the next is determined by team performance; there is no way for a single character to explore the divisional courses.

There should be some variety in the courses within each division so as to provide for different balances of character skills. Ideally, each of the courses should emphasise a different aspect; speed, strength, stamina and all-round ability. This strategy lends itself to 4 (or multiples thereof) courses per division.

Each division has 4 characters competing in it at any time; if fewer than four (human) players are available then the numbers are made up using computer controlled characters. This means that 12 matches in total are required in order for every character to play against every other character. The results of matches where computer opponents play against computer opponents will be randomly determined (but weighted according to the characters involved).

Because there are 12 pairings between the characters and several (probably 4) courses to play on, it is impractical (and boring) to play each pairing on every course. Automatic selection of courses by either a "random" or a round-robin scheme is a possibility but this could result in a player being disadvantaged through no fault of their own. An honour scheme has a certain appeal; this would allow the player with the poorer rating to choose the course for each match.

A player may choose freely which of the characters in their team to use for any match, thus if they have chosen a well balanced team they should be able to compete fairly on any course.

*To be decided:-*

- *How many courses per division (4)?*
- *How are matches allocated to courses?*

### Ratings, Points and Times

Every character has a rating which is simply the sum total of points they have accumulated in their career (since being selected for a team). When this total exceeds certain thresholds, the character will receive offers of extreme matches.

Each team also has a rating which is calculated as the average of the members ratings. This means that the player has to use all the characters within the team in order to progress through divisions.

Points are allocated as follows:-

Getting to the goal within the time limit	1 Point
Getting to the goal before your opponent	1 Point
Setting a new best time for a course	2 Points

The time required to complete a course is obviously going to be vary somewhat from course to course. The ideal playing time will be determined by play testing; at present it is felt this should be between 3 and 6 minutes.

The league simply rates the characters in order of the points they have accumulated within that league (league points are set to zero at the start of each new league). If points tie between two or more characters at the end of a round, then the character who goes forward is the one with the best aggregate time over all courses played in the league so far. In the unlikely event that the aggregate times match then a playoff will be held.

## **Extreme Zenith**

The extreme matches differ from other types in that they use completely different styles of level and characters can be terminated if they aren't careful. These levels will be completely secret until a character achieves a high enough rating. Note that it is the character who is offered an extreme level rather than the team; the team rating is used only to progress (or regress) from division to division. This is a strictly "take it or leave it" offer so if the player doesn't wish to risk the character then they simply ignore the offer, missing the possibility of some bonuses (nothing ventured, nothing gained...)

The extreme levels use exactly the same map format as the regular levels however the design considerations are quite different. The extreme levels are meant to represent real buildings but in a gothic cartoon style (see Batman videos and comics, especially Frank Miller's The Dark Knight Returns).

The block designs for extreme levels are expected to be things like gargoyles or other ornate architectural features. As these will require considerable rendering capacity it is likely that extreme levels will use substantially fewer blocks. Refer to figure 4 in Appendix B for an example of how an extreme level might look.

## User Interface (Front End)

The user interface (menus and control pad functions) should be as uniform as possible to make the game easily accessible and intuitive. All the characters will be made to move (walk, run, jump etc.) using standard control inputs; special moves will require some thought to give logical and consistent button allocations (obviously the character designs and special moves must be finalised first).

Controller buttons (LEFT, RIGHT, A, B etc.) refer to the prototype Ultra64 Control Pad; it is assumed that any other controller will be entirely compatible with this in terms of number and location of buttons.

Every effort will be made to make the game as friendly as possible; advanced user interface techniques are under investigation. Features such as remembering a players last selections and offering them as defaults are easy to program and save players the tedium of selecting the same things over and over again. This type of 'intelligent' interface is easy to expand on; it is possible to remember a number of previous selections and provide these as options.

### Global Functions

These are functions which are available at every level of every menu and within the game. They must therefore be allocated to reserved buttons or sequences so that there is no ambiguity.

#### Select

This chooses the highlit menu option, character, world, course etc. This has no effect while a game is in progress as there is no need to select anything (at present anyway).

*To be decided:-*

- *Use START or A button?*

#### Exit

This exits from the current menu to the previous menu saving any changes to the state of the system. When in the fundamental mode this has no effect.

*To be decided:-*

- *Which button(s) to use?*

#### Abort

This exits from the current menu to the previous menu without making any changes to the state of the system. When in the fundamental mode this has no effect.

*To be decided:-*

- *Which button(s) to use?*

#### Change Selection

This navigates around menus or other selection panels within a single screen.

*To be decided:-*

- *Use UP, DOWN, LEFT, RIGHT buttons?*

## Menu Hierarchy

The functionality of the important modes, menus and other selection screens is defined here.

### Fundamental Menu

This is the menu entered by pressing the start button after game initialisation (which includes loading and licence screens). The menu is:-

- Create Team
- Review Team
- Play

### Create Team Menu

This goes into team selection mode (hopefully using a virtual environment walk-through) where the player can check out the available characters and choose the ones they like best. Once the team is chosen the player uses the exit function to return to the fundamental menu.

*To be finalised.*

### Review Team Menu

This allows the player to have a detailed look at the members of their team. Much more information is available, such as wins and losses, best times and so on. The player cannot make any changes within this menu.

*To be finalised.*

### Play Menu

The first choice here is to select the number of players and then the style of game.

*To be finalised.*

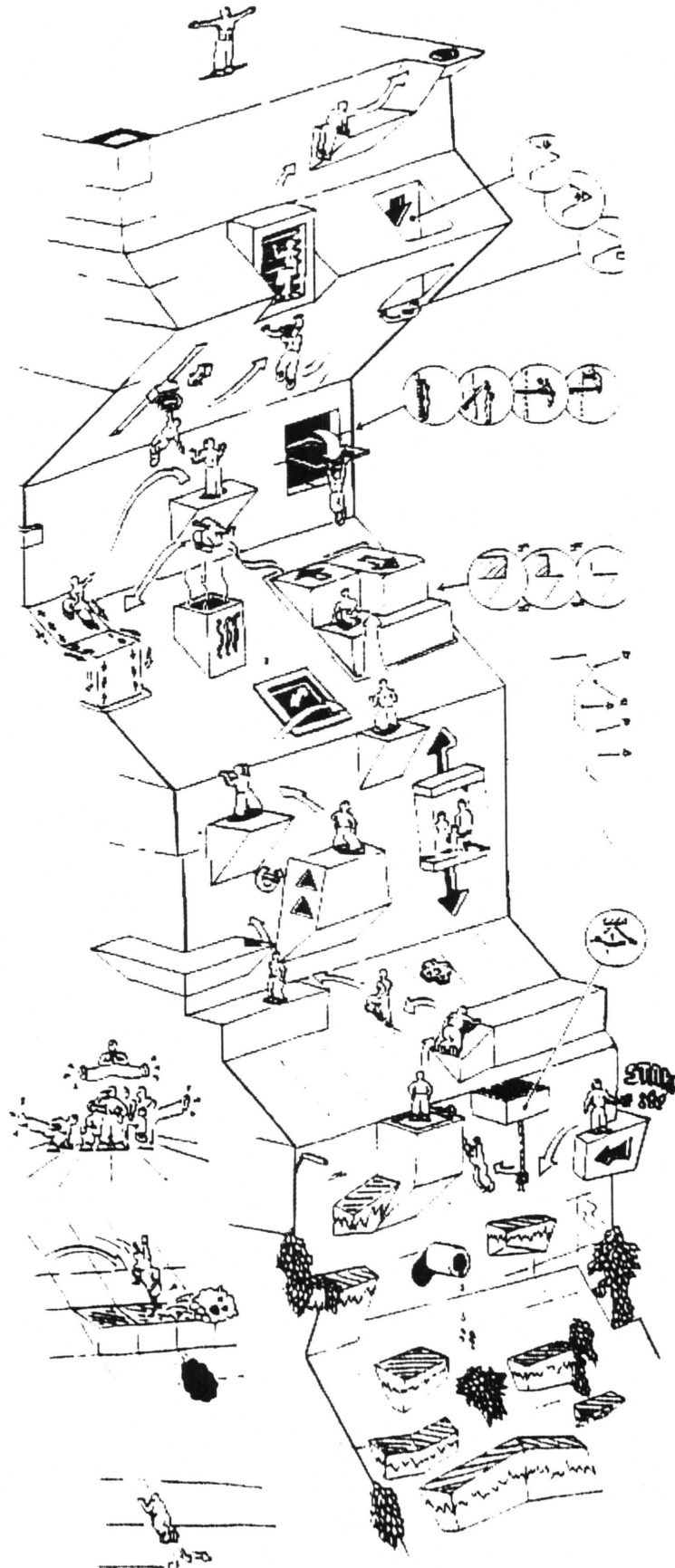
## Game Controls

These are the functions of the controller buttons while a game is in play. The naming of buttons is according to the prototype controller design received in December 1994.

LEFT	Move left along the axis of the course, attempt to climb if adjacent to wall or block.
RIGHT	Move right along the axis of the course, attempt to climb if adjacent to wall or block.
UP	Move up the course, attempt to climb if adjacent to wall or block.
DOWN	Move down the course, attempt to climb if adjacent to wall or block.
L	Special move 1. Unique to the individual characters.
R	Special move 2. Unique to the individual characters.
START	Pause the game.
SELECT	Ignored.
A	Run.
B	Duck.
N	Kick.
S	Jump.
E	Push or Pull.
W	Punch.

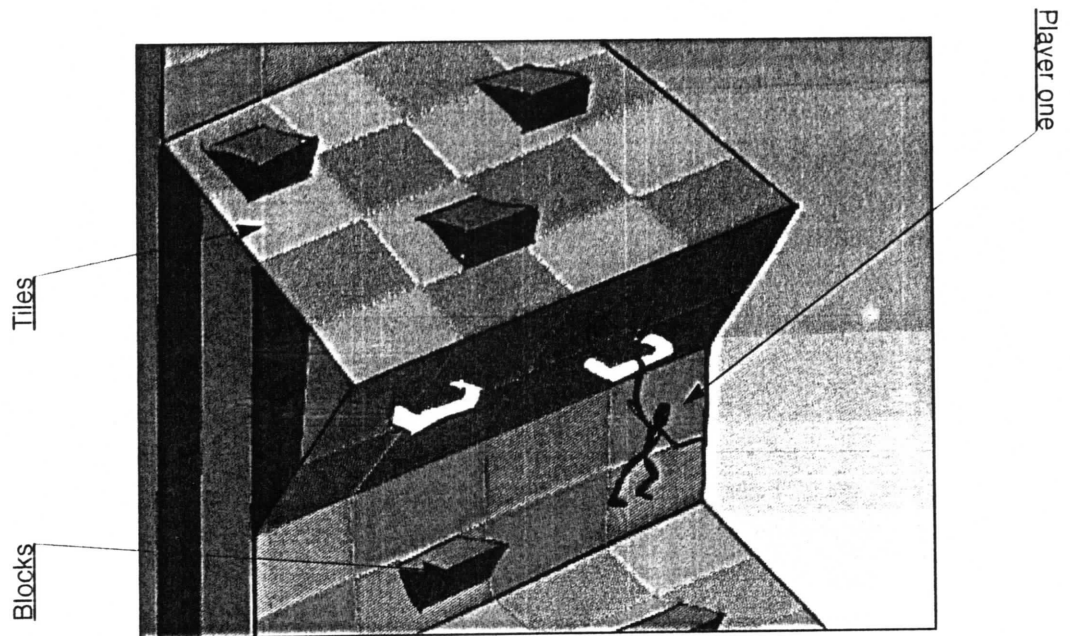
*To be finalised: Need more information on combat, special moves etc.*

## Appendix A - Original Design Drawing

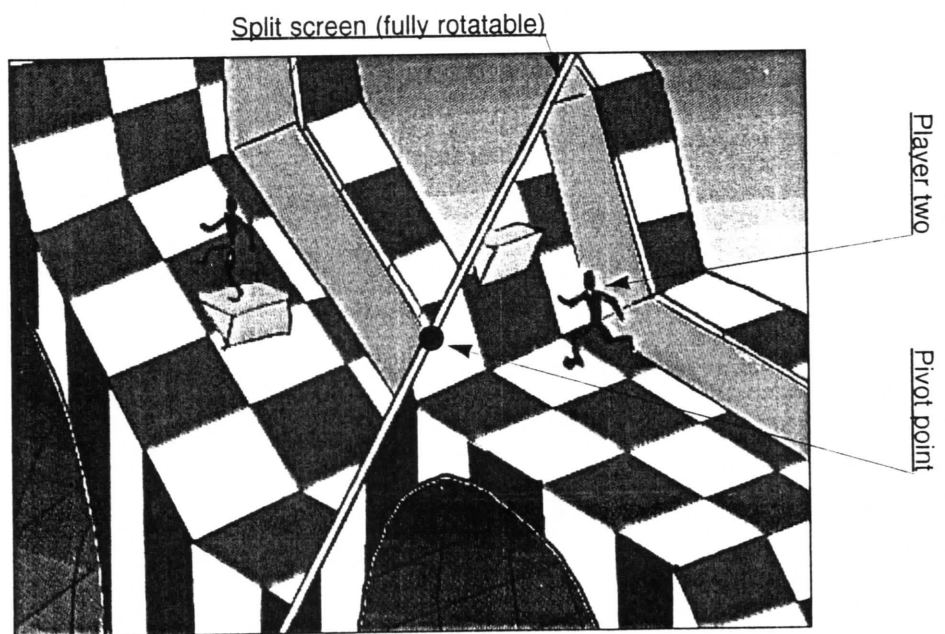




## Appendix B - Illustrations



**Fig 1 :** Typical screen shot of a one player game.



**Fig 2 :** Typical screen shot of a two player game.

Note the unique split screen which rotates from the centre through a full 360 degree spin. Here we see player one is slightly above player two so the split screen keeps adjusting to show their relative positions.

Fig 3 : Rendered screen of one player game.

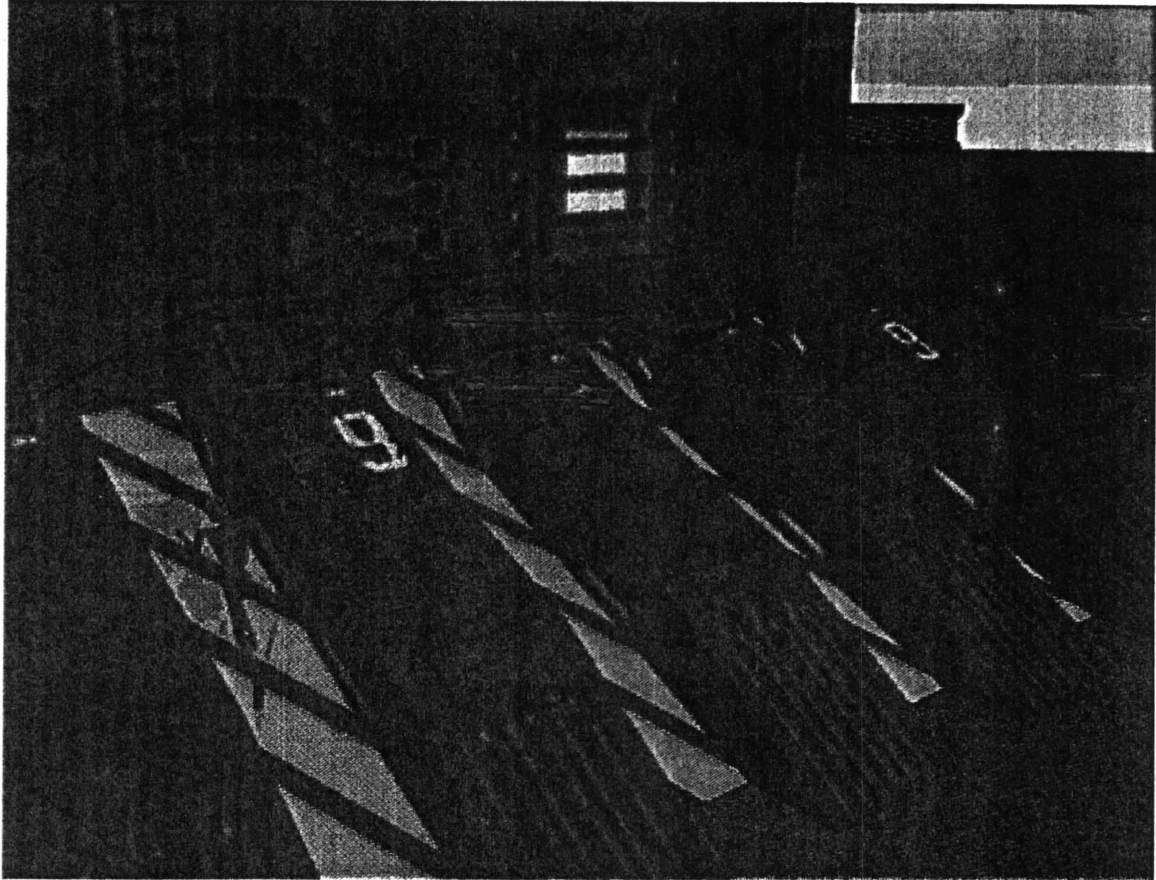
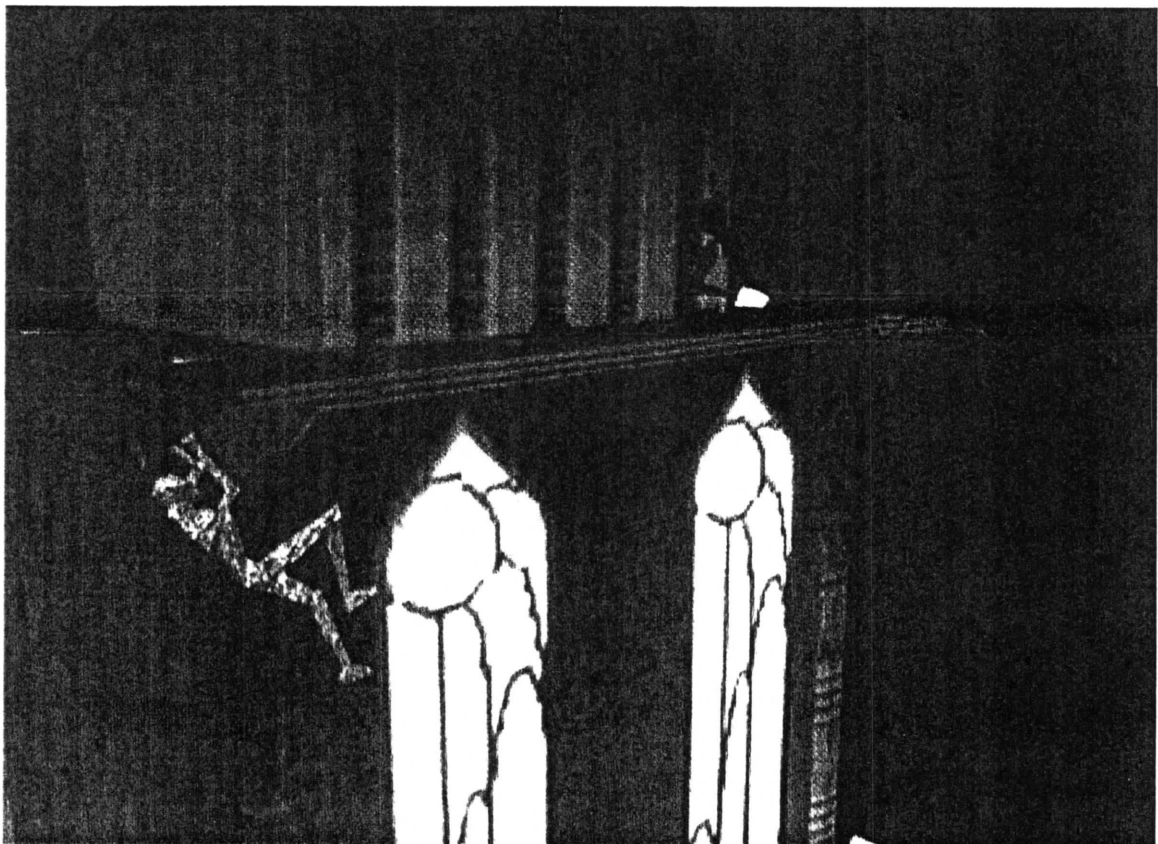
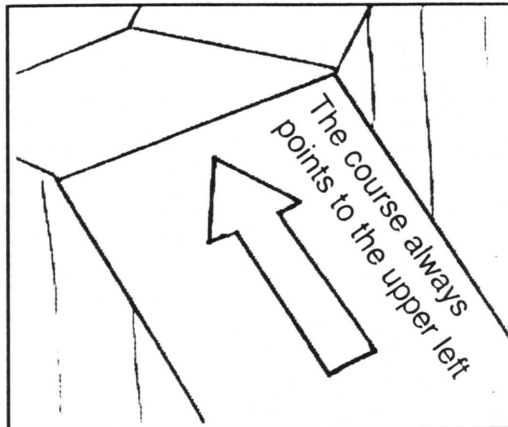


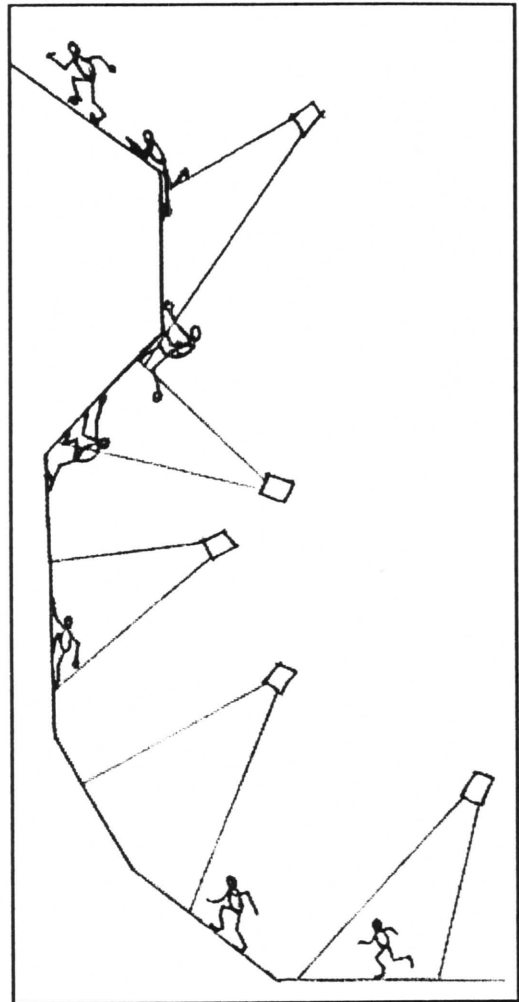
Fig 4 : Rendered screen showing extreme match.



## Camera Angles

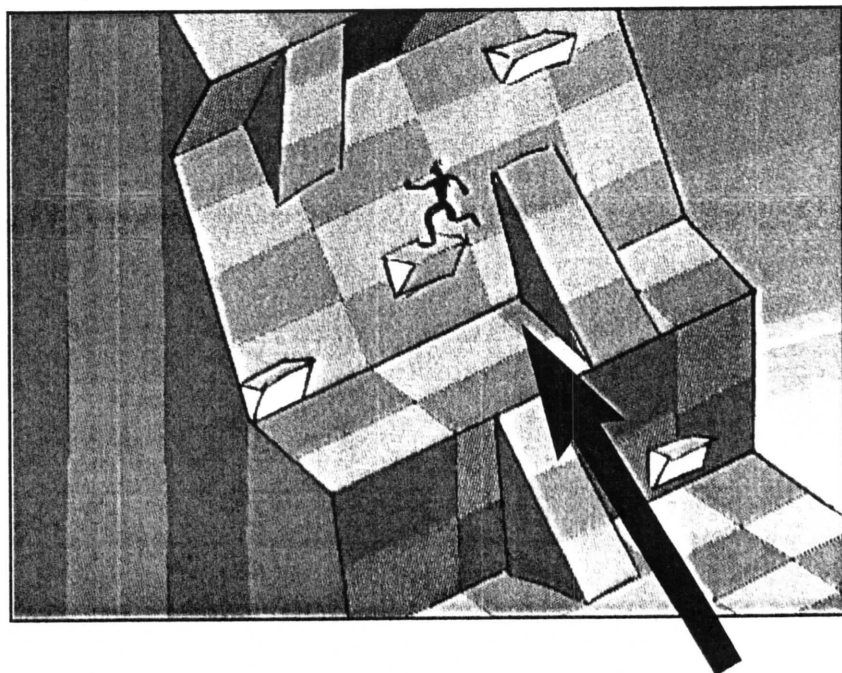


**Fig 5:** The view screen will always look at the landscape from the bottom right corner to the top left. If the course changes direction, the camera will rotate to accomodate the same view point of botom right to top left.



**Fig 6 :**  
As the camera reaches an overhang, the view will dip to show the area more clearly.

**Fig 7:** Always bottom right to top left.



## Example Course

Fig 8 : This would be a tough course to complete, containing many sections to tackle.

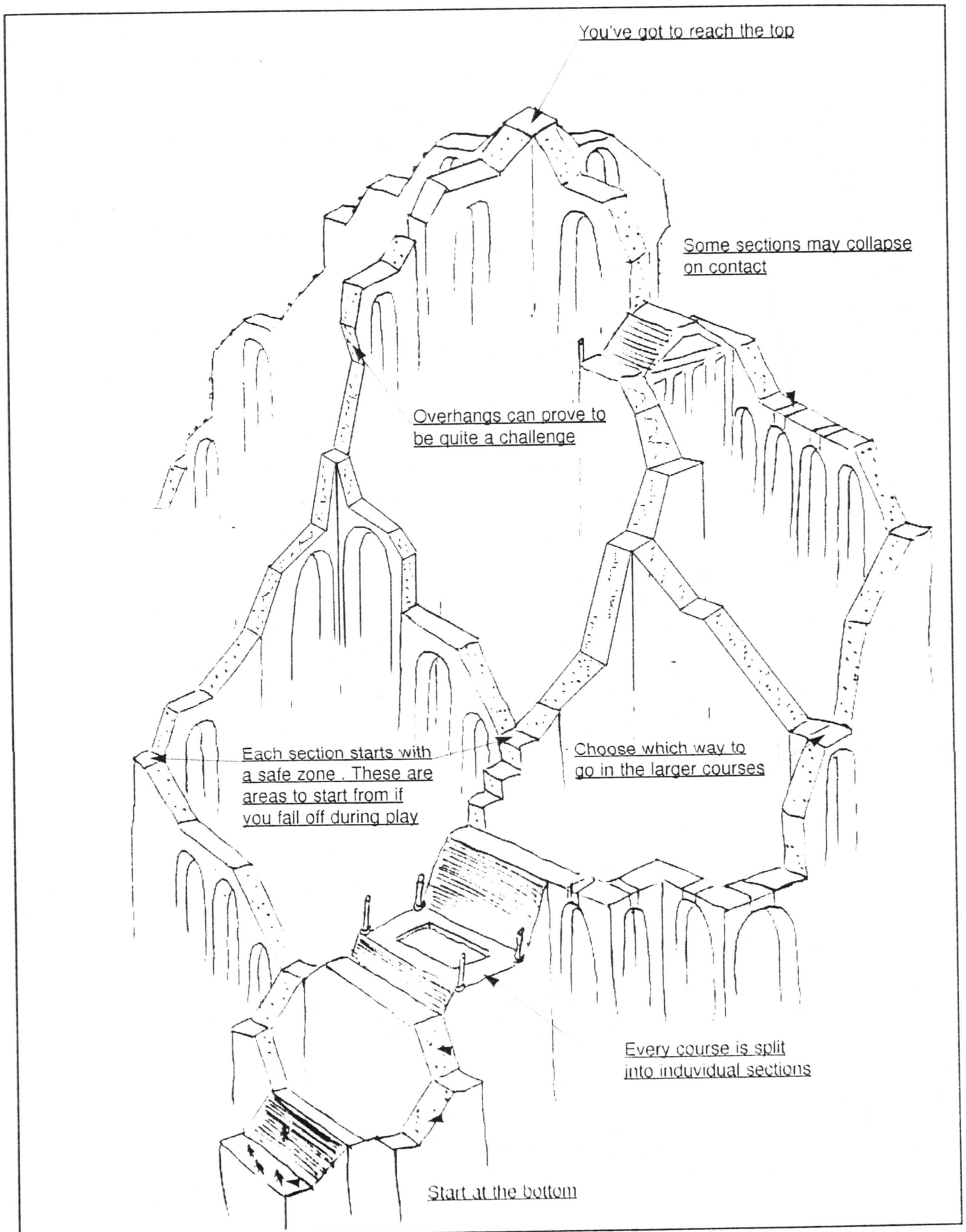


Fig 9 : Courses can take many forms...

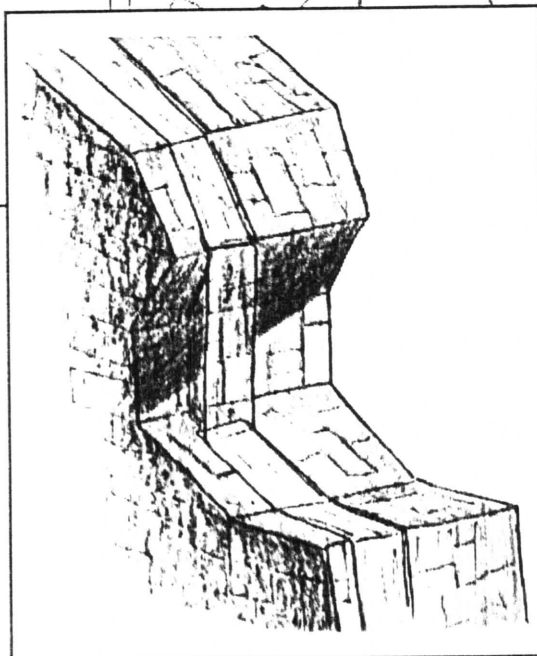
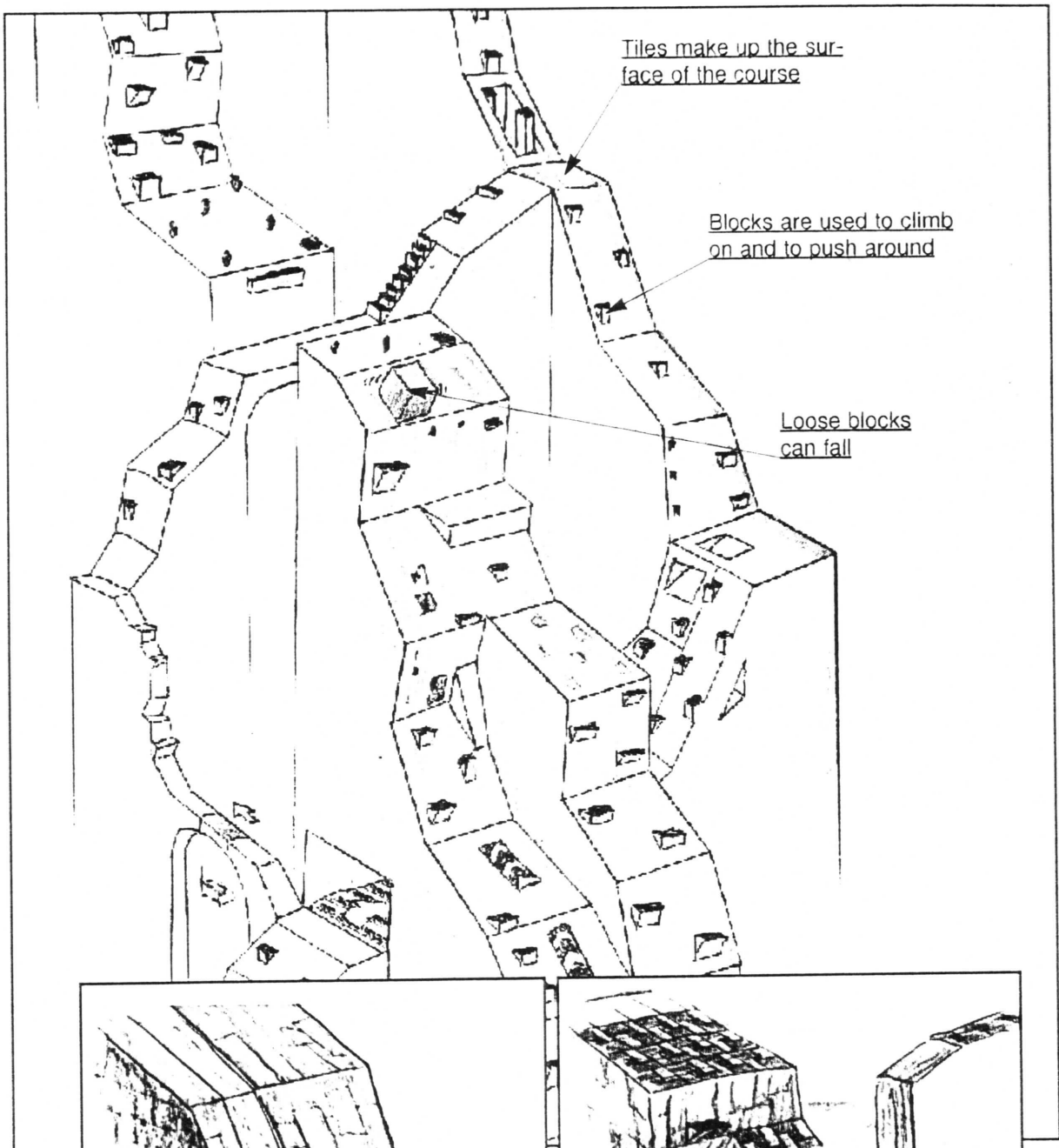


Fig 10 : World style example (Egyptian)

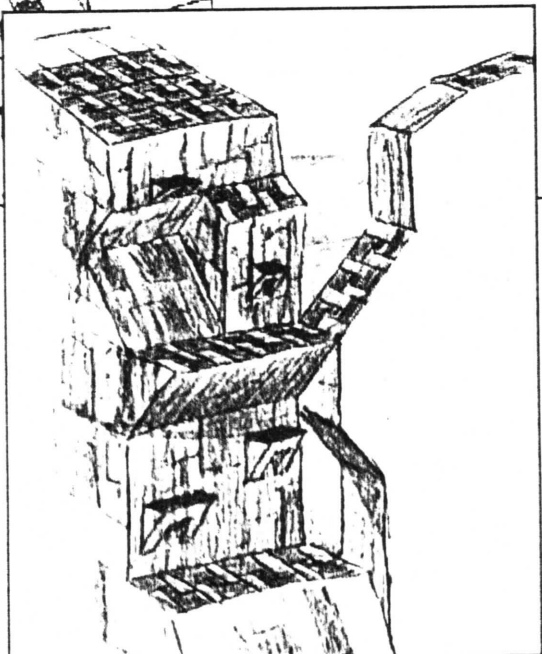
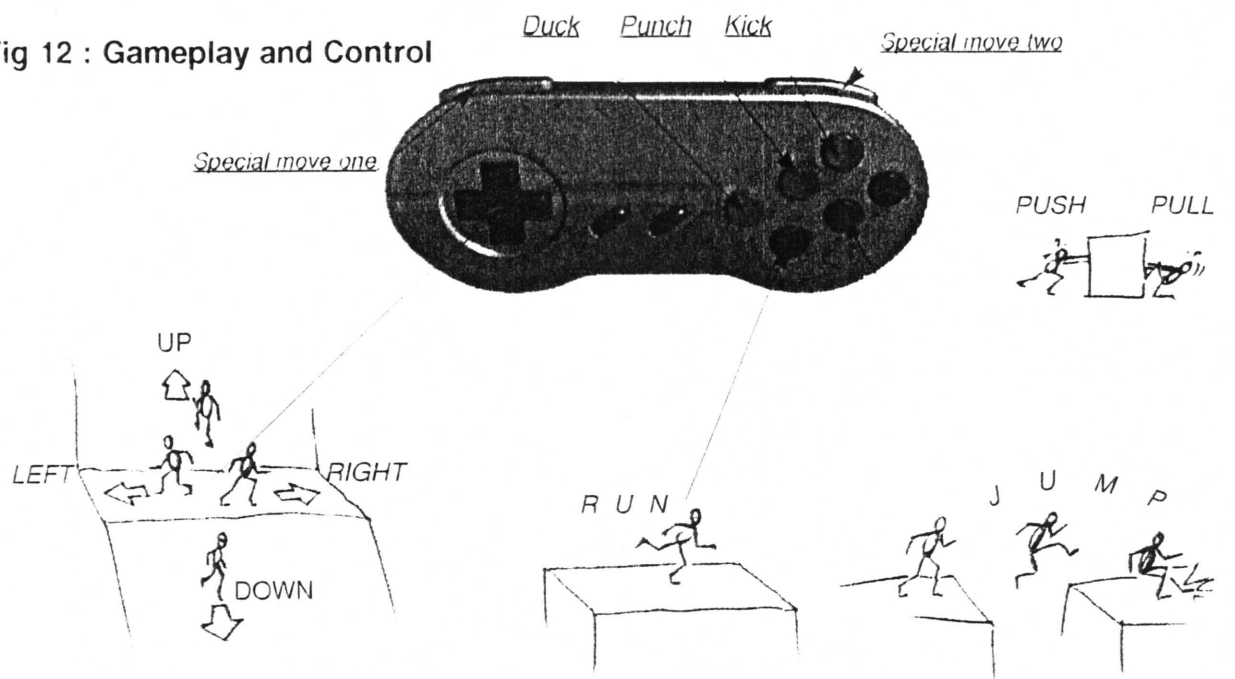
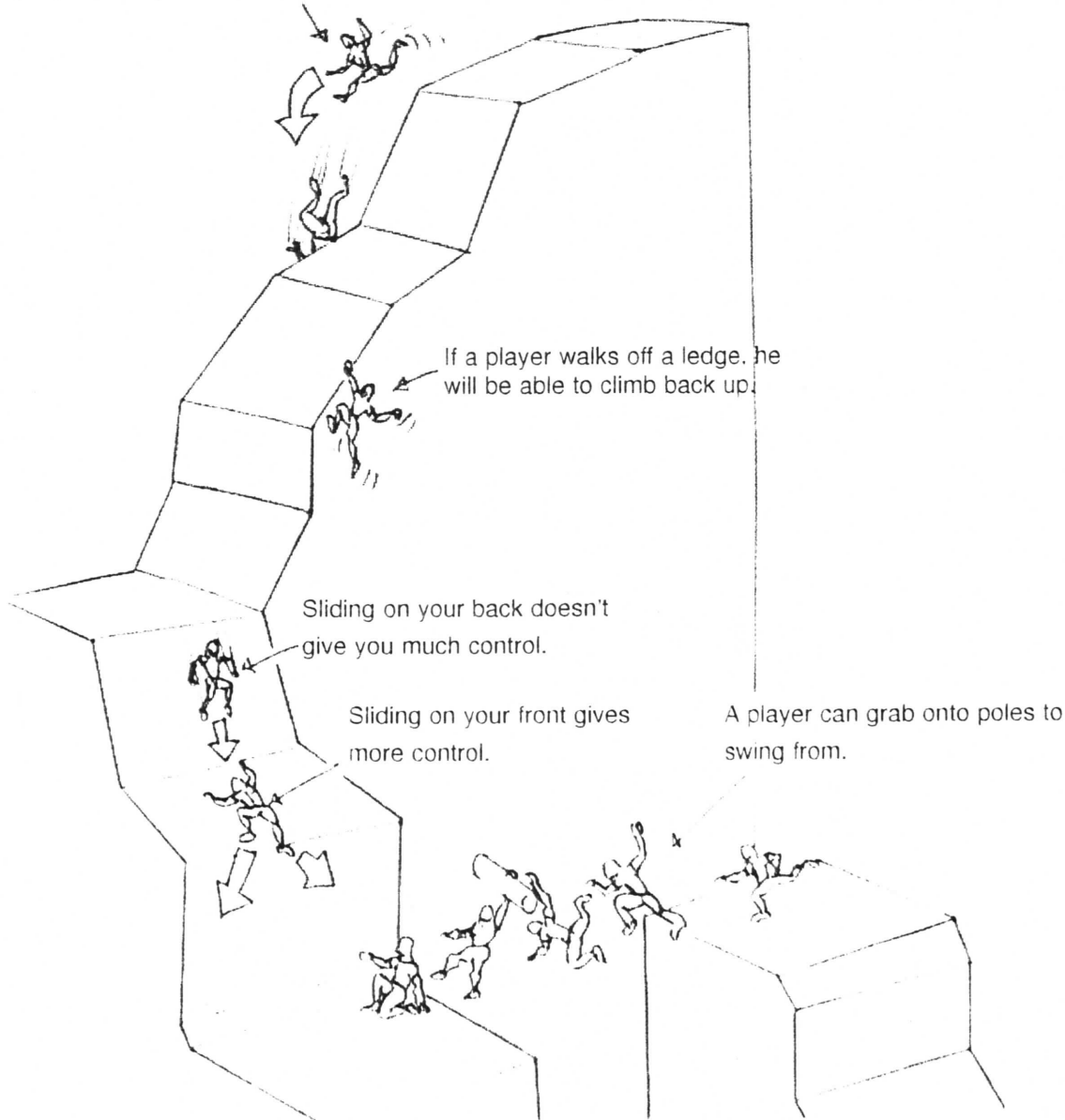


Fig 11 : World style example (Inca)

Fig 12 : Gameplay and Control



If a player jumps off the course, he will fall.





Game Elements

Fig 13 : Collapsing paths

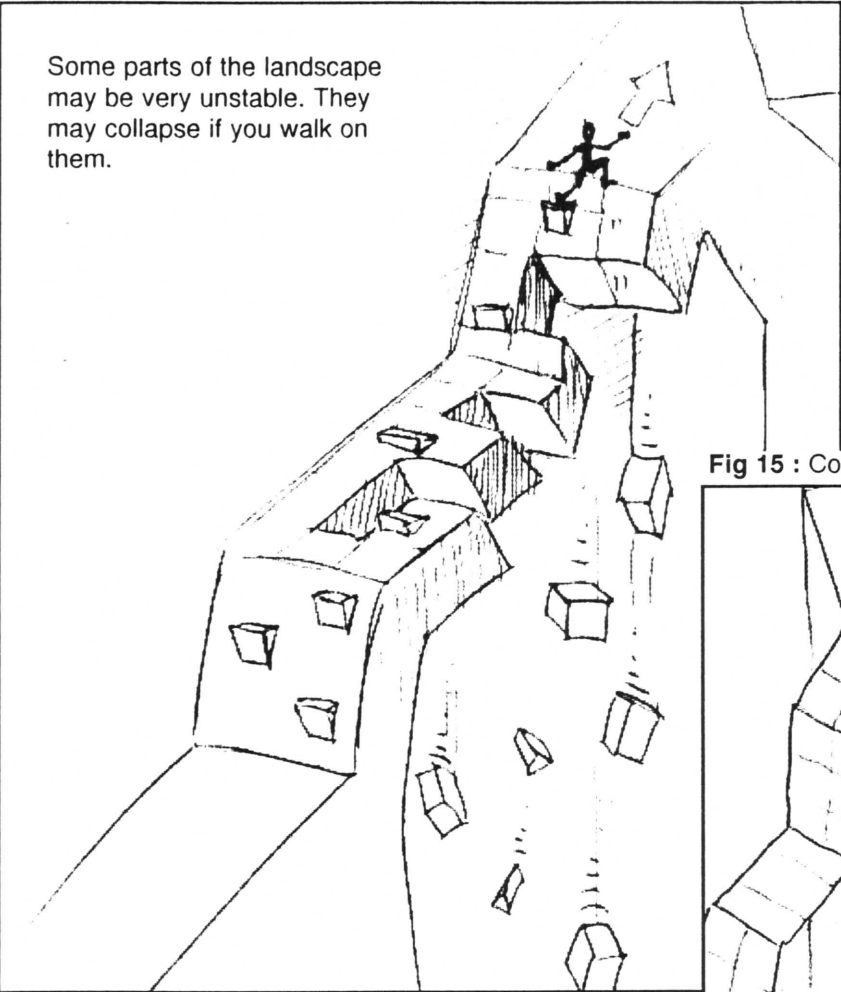


Fig 14 : Blocks

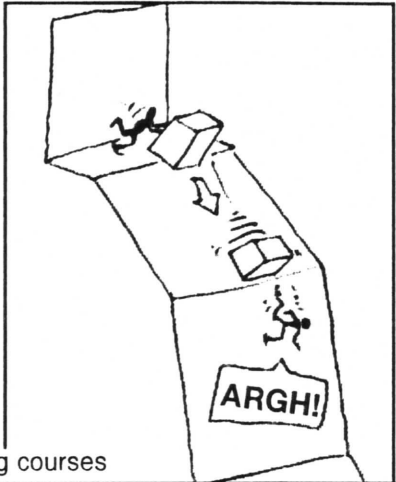


Fig 15 : Collapsing courses

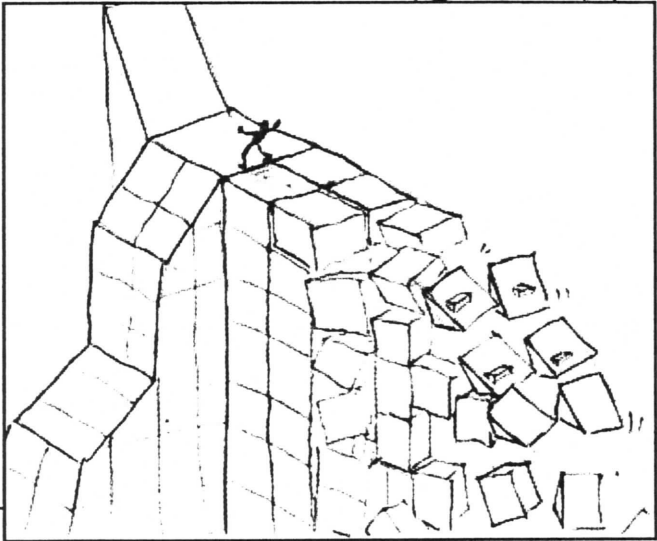


Fig 16 : Rotating path

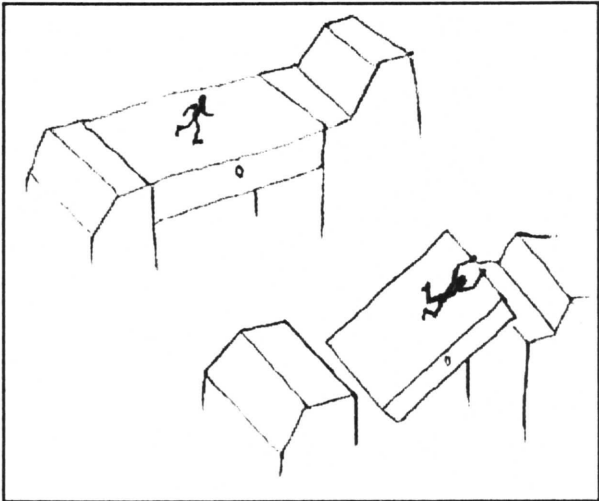
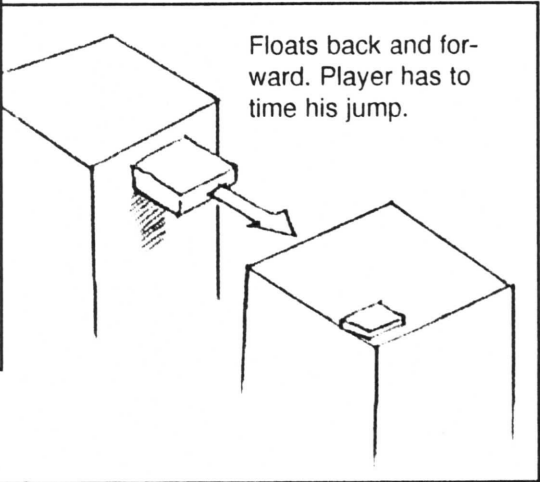


Fig 17 : Floating bridge



## Game Elements

Fig 18 : Rotating circle

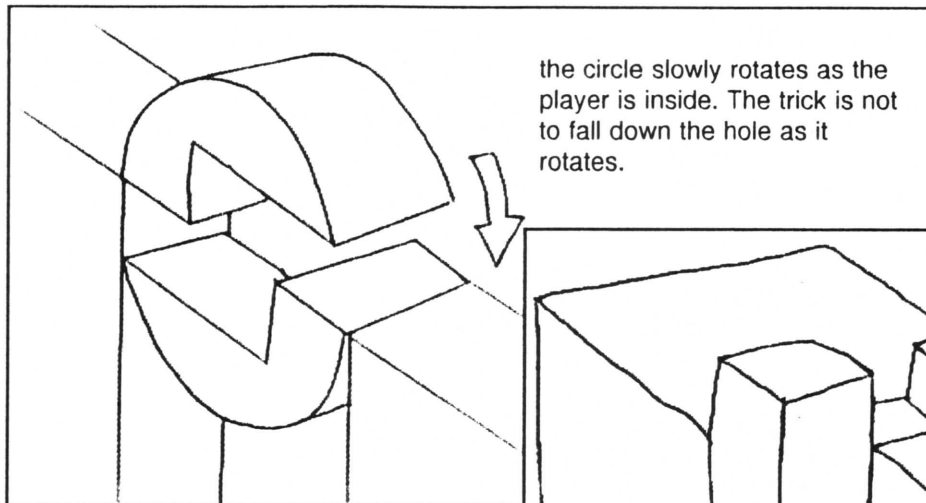


Fig 19 : Moving blocks

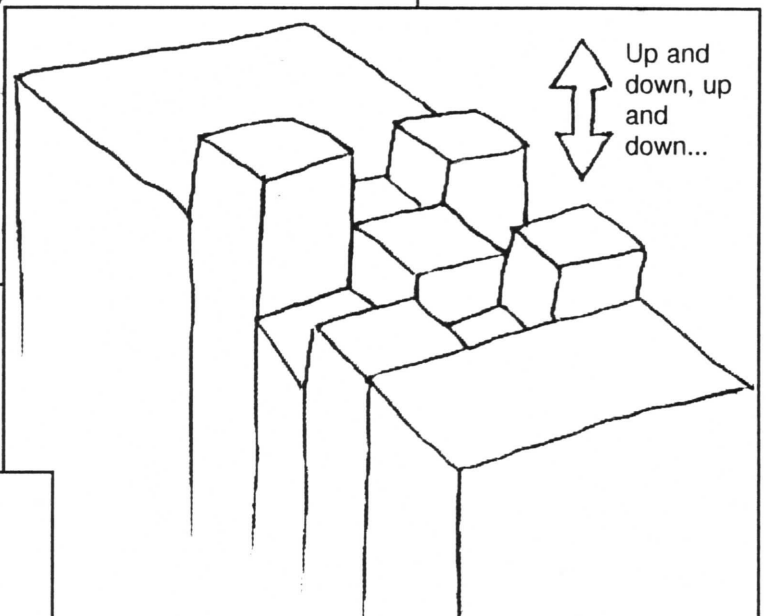


Fig 20: Escalators

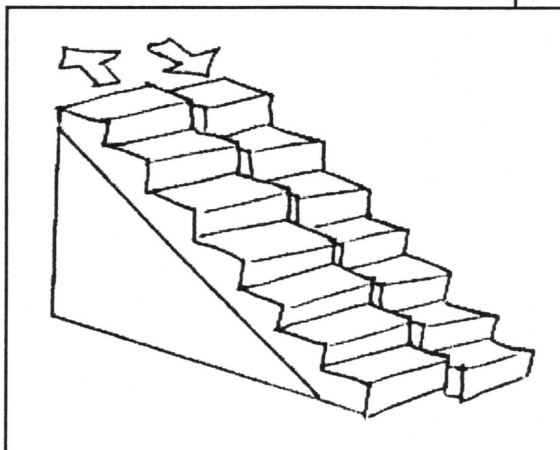


Fig 21: Wind and air vents.

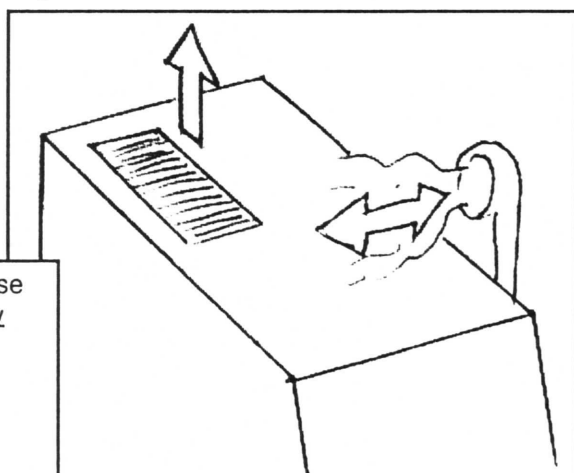


Fig 22: Bridges

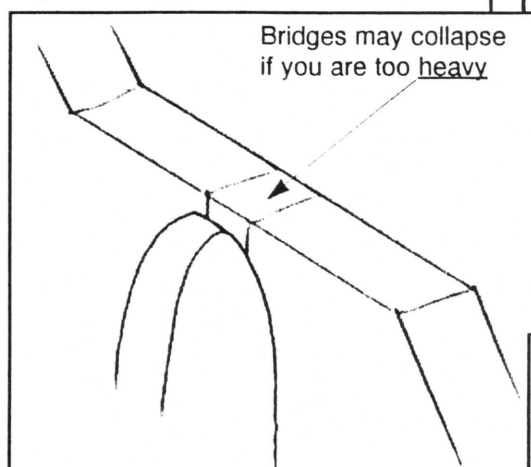
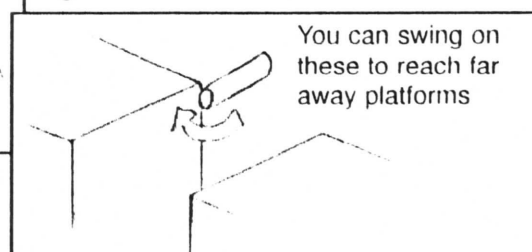


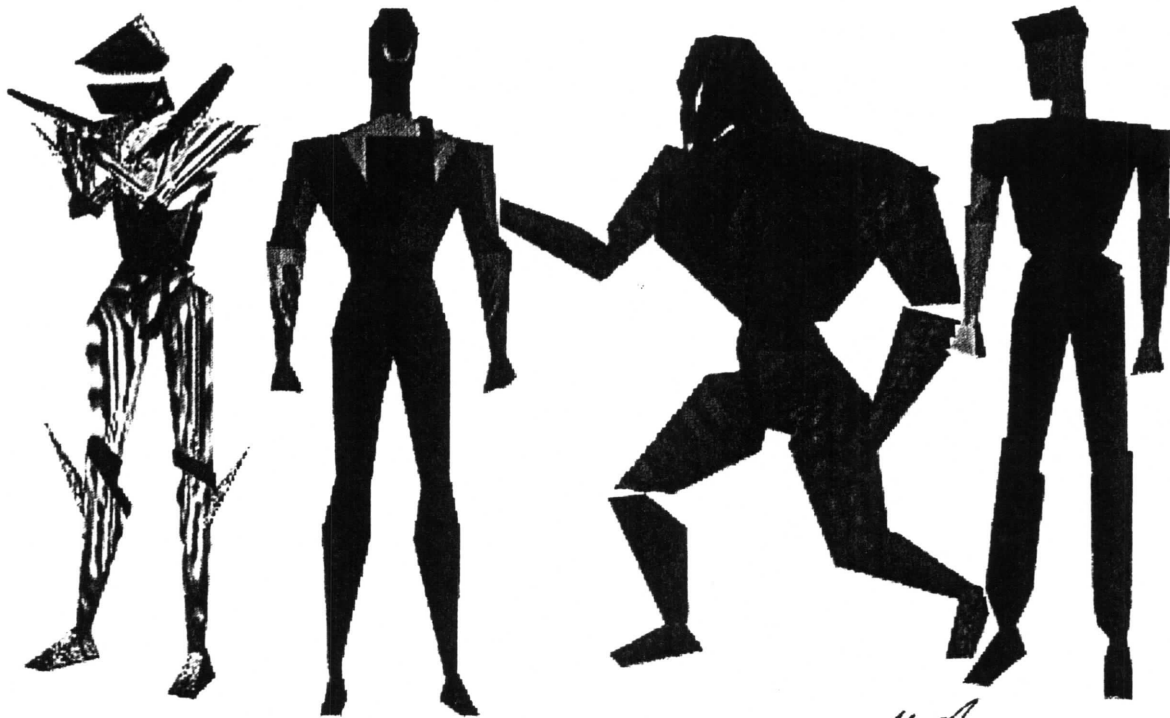
Fig 23: Poles





## Characters

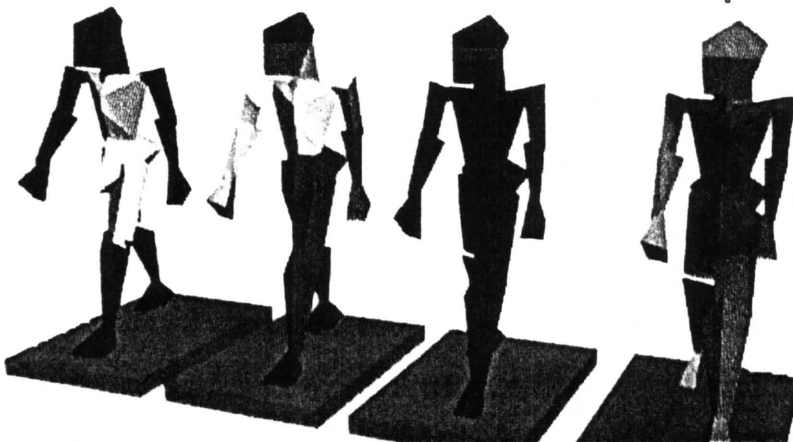
Fig 24: Models and characters



There will be eight basic models within the game. These will cover a range of about 30 characters.

Each team will wear a common recognisable harness with the team decal clearly visible.

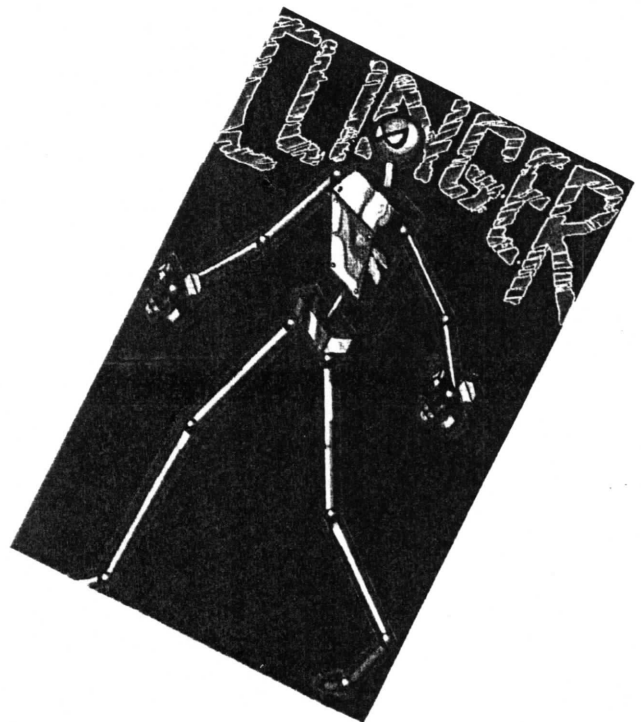
By changing the texture maps and altering a small amount of 'cosmetic' polygons (hair, clothing etc), the characters can be made to look unique when still using the same basic model.

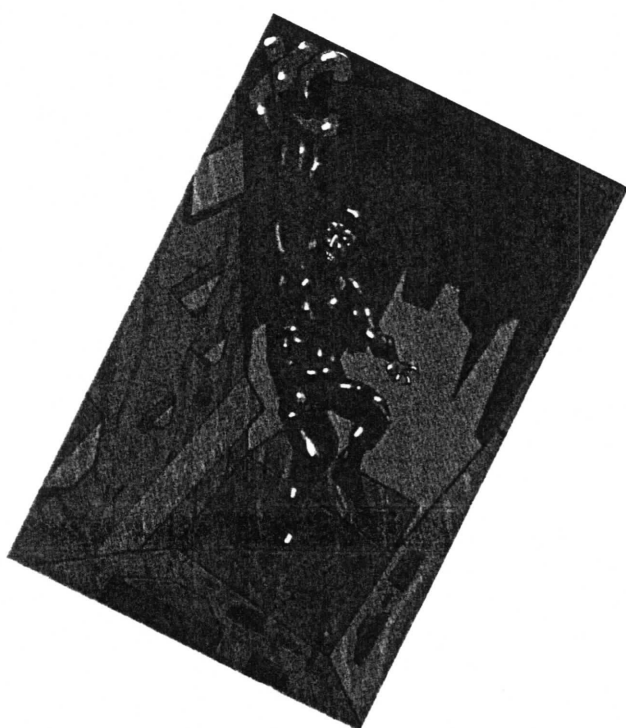
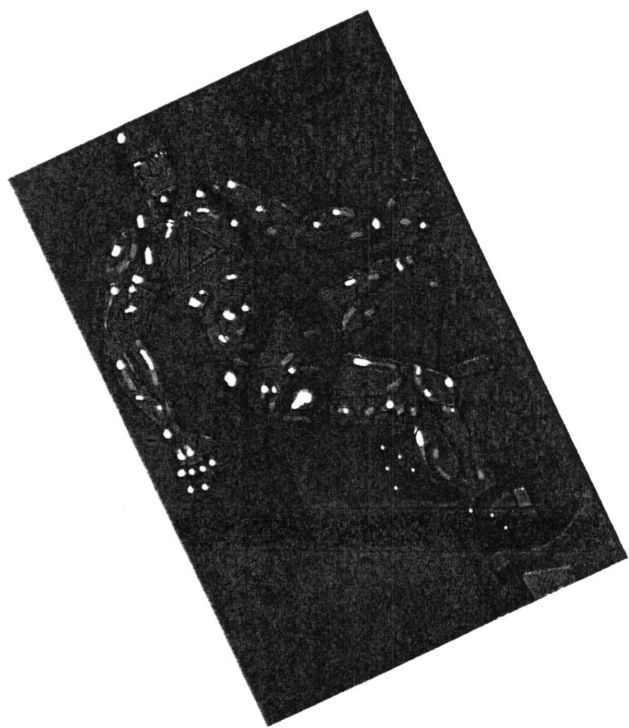
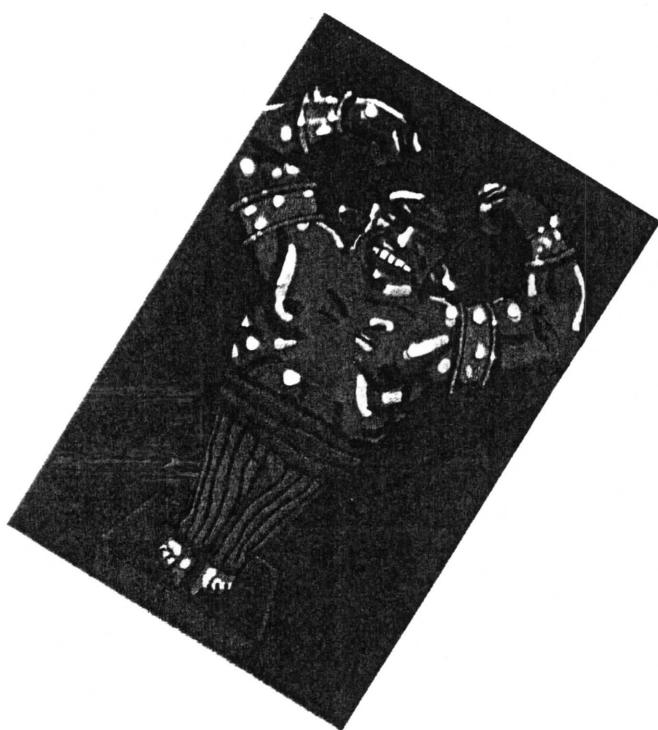


## Characters



Fig 25: Team members will consist of a variety of characters.





## Appendix C - List of Character Animation Sequences

Below is a list of the basic animation sequences used for each character. Each character model may require it's own version of each of these in order to produce aesthetically pleasing results. However, it is hoped that as many as possible of these animation's can be shared between models in order to reduce production time and ROM storage.

Each character will also have some special moves unique to it; these are not detailed here.

- 1) Standing on flat ground
- 2) Standing on slope facing up
- 3) Standing on slope facing down
- 4) Standing on slope facing left
- 5) Standing on slope facing right
- 6) Turning on the spot (clockwise)
- 7) Turning on the spot (anti-clockwise)
- 8) Walking on flat ground
- 9) Walking down slope
- 10) Walking up slope
- 11) Walking left along slope
- 12) Walking right along slope
- 13) Taking a step forward
- 14) Taking a step back
- 15) Side Step left
- 16) Side Step right
- 17) Running on flat
- 18) Running up slope
- 19) Running down slope
- 20) Running left along slope
- 21) Running right along slope
- 22) Jumping from a standing position
- 23) Jumping from a walk
- 24) Jumping from a run
- 25) Jumping down to a lower platform
- 26) Landing
- 27) Hanging from a platform ledge (two handed)
- 28) Changing from 2 to 1 hands (getting tired)
- 29) Hanging by 1 hand
- 30) Letting go
- 31) Lower down from ledge
- 32) Scramble up slope
- 33) Climb flat wall
- 34) Pull up to ledge
- 35) Climb up rope
- 36) Ascending ladder
- Slide down slope (on your front):
- 37) Leaning left
- 38) Leaning right
- Slide down slope (on your back):
- 39) Leaning left
- 40) Leaning right
- 41) Slide from your front to your back
- 42) Duck
- 43) Punch

- 44) Kick
- 45) Hit by a force (rock, opponent)
- 46) Getting crushed by a falling object(s)
  
- 47) Push
- 48) Pull
- 49) Picking things up (tools etc.)
- 50) Throw things (rocks etc.)
  
- 51) Swinging on bars (2 handed)
- 52) Swinging on bars (1 handed)
- 53) Shimmy (from bar to bar)
  
- 54) Slide on ice
- 55) Slipping and falling on ice
- 56) Getting back up
  
- 57) Crawl
  
- 58) Floating in the air
  
- 59) Victory actions
  
- Special moves:**
- 60 - 68) Individual fight and acrobatic moves