SYSTEM FOR CONTINUOUS MONITORING OF PHYSICAL ACTIVITY DURING UNRESTRICTED MOVEMENT

A movement skills assessment system (10) without a confining field includes a wireless position tracker (14, 16) coupled to a personal computer (22) and viewing monitor (28) for the purpose of quantifying the ability of a player to move over sport specific distances and directions. The monitor displays a computer-generated virtual space (30) which is a graphic representation of a defined physical space in which the player moves and the current position of the player. Interactive software displays a target destination distinct from the current position of the player. The player moves as rapidly as possible to the target destination. As the movement sequence is repeated, performance-related parameters including quickness, heart rate activity as related to physical activity, consistency of maintaining a set position, and energy expenditure are measured. The system has applications in sports, commercial fitness and medical rehabilitation.
FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

<table>
<thead>
<tr>
<th>Code</th>
<th>Country</th>
</tr>
</thead>
<tbody>
<tr>
<td>AM</td>
<td>Armenia</td>
</tr>
<tr>
<td>AT</td>
<td>Austria</td>
</tr>
<tr>
<td>AU</td>
<td>Australia</td>
</tr>
<tr>
<td>BB</td>
<td>Barbados</td>
</tr>
<tr>
<td>BE</td>
<td>Belgium</td>
</tr>
<tr>
<td>BF</td>
<td>Burkina Faso</td>
</tr>
<tr>
<td>BG</td>
<td>Bulgaria</td>
</tr>
<tr>
<td>BJ</td>
<td>Benin</td>
</tr>
<tr>
<td>BR</td>
<td>Brazil</td>
</tr>
<tr>
<td>BY</td>
<td>Belarus</td>
</tr>
<tr>
<td>CA</td>
<td>Canada</td>
</tr>
<tr>
<td>CF</td>
<td>Central African Republic</td>
</tr>
<tr>
<td>CG</td>
<td>Congo</td>
</tr>
<tr>
<td>CH</td>
<td>Switzerland</td>
</tr>
<tr>
<td>CI</td>
<td>Côte d'Ivoire</td>
</tr>
<tr>
<td>CM</td>
<td>Cameroon</td>
</tr>
<tr>
<td>CN</td>
<td>China</td>
</tr>
<tr>
<td>CS</td>
<td>Czechoslovakia</td>
</tr>
<tr>
<td>CZ</td>
<td>Czech Republic</td>
</tr>
<tr>
<td>DE</td>
<td>Germany</td>
</tr>
<tr>
<td>DK</td>
<td>Denmark</td>
</tr>
<tr>
<td>EE</td>
<td>Estonia</td>
</tr>
<tr>
<td>ES</td>
<td>Spain</td>
</tr>
<tr>
<td>FI</td>
<td>Finland</td>
</tr>
<tr>
<td>FR</td>
<td>France</td>
</tr>
<tr>
<td>GA</td>
<td>Gabon</td>
</tr>
<tr>
<td>GB</td>
<td>United Kingdom</td>
</tr>
<tr>
<td>GE</td>
<td>Georgia</td>
</tr>
<tr>
<td>GN</td>
<td>Guinea</td>
</tr>
<tr>
<td>GR</td>
<td>Greece</td>
</tr>
<tr>
<td>HU</td>
<td>Hungary</td>
</tr>
<tr>
<td>IE</td>
<td>Ireland</td>
</tr>
<tr>
<td>IT</td>
<td>Italy</td>
</tr>
<tr>
<td>JP</td>
<td>Japan</td>
</tr>
<tr>
<td>KE</td>
<td>Kenya</td>
</tr>
<tr>
<td>KG</td>
<td>Kyrgyzstan</td>
</tr>
<tr>
<td>KP</td>
<td>Democratic People's Republic of Korea</td>
</tr>
<tr>
<td>KR</td>
<td>Republic of Korea</td>
</tr>
<tr>
<td>KZ</td>
<td>Kazakhstan</td>
</tr>
<tr>
<td>LI</td>
<td>Liechtenstein</td>
</tr>
<tr>
<td>LK</td>
<td>Sri Lanka</td>
</tr>
<tr>
<td>LR</td>
<td>Liberia</td>
</tr>
<tr>
<td>LT</td>
<td>Lithuania</td>
</tr>
<tr>
<td>LU</td>
<td>Luxembourg</td>
</tr>
<tr>
<td>LV</td>
<td>Latvia</td>
</tr>
<tr>
<td>MC</td>
<td>Monaco</td>
</tr>
<tr>
<td>MD</td>
<td>Republic of Moldova</td>
</tr>
<tr>
<td>MG</td>
<td>Madagascar</td>
</tr>
<tr>
<td>ML</td>
<td>Mali</td>
</tr>
<tr>
<td>MN</td>
<td>Mongolia</td>
</tr>
<tr>
<td>MR</td>
<td>Mauritania</td>
</tr>
<tr>
<td>MW</td>
<td>Malawi</td>
</tr>
<tr>
<td>MX</td>
<td>Mexico</td>
</tr>
<tr>
<td>NE</td>
<td>Niger</td>
</tr>
<tr>
<td>NL</td>
<td>Netherlands</td>
</tr>
<tr>
<td>NO</td>
<td>Norway</td>
</tr>
<tr>
<td>NZ</td>
<td>New Zealand</td>
</tr>
<tr>
<td>PL</td>
<td>Poland</td>
</tr>
<tr>
<td>PT</td>
<td>Portugal</td>
</tr>
<tr>
<td>RO</td>
<td>Romania</td>
</tr>
<tr>
<td>RU</td>
<td>Russian Federation</td>
</tr>
<tr>
<td>SD</td>
<td>Sudan</td>
</tr>
<tr>
<td>SE</td>
<td>Sweden</td>
</tr>
<tr>
<td>SG</td>
<td>Singapore</td>
</tr>
<tr>
<td>SI</td>
<td>Slovenia</td>
</tr>
<tr>
<td>SK</td>
<td>Slovakia</td>
</tr>
<tr>
<td>SN</td>
<td>Senegal</td>
</tr>
<tr>
<td>SZ</td>
<td>Swaziland</td>
</tr>
<tr>
<td>TD</td>
<td>Chad</td>
</tr>
<tr>
<td>TG</td>
<td>Togo</td>
</tr>
<tr>
<td>TJ</td>
<td>Tajikistan</td>
</tr>
<tr>
<td>TT</td>
<td>Trinidad and Tobago</td>
</tr>
<tr>
<td>UA</td>
<td>Ukraine</td>
</tr>
<tr>
<td>UG</td>
<td>Uganda</td>
</tr>
<tr>
<td>US</td>
<td>United States of America</td>
</tr>
<tr>
<td>UZ</td>
<td>Uzbekistan</td>
</tr>
<tr>
<td>VN</td>
<td>Viet Nam</td>
</tr>
</tbody>
</table>
SPECIFICATION

I. TITLE OF THE INVENTION

"System for Continuous Monitoring of Physical Activity During Unrestricted Movement"

II. IDENTIFICATION OF THE INVENTORS

Barry J. French
Kevin R. Ferguson

III. CROSS-REFERENCES


IV. GOVERNMENT RIGHTS

The present application pertains to an invention that was not performed under any federally sponsored research and development.

V. BACKGROUND

A. Field of the Invention

The present invention relates to a system for assessing movement and agility skills and, in particular to a wireless position tracker for continuously tracking and determining player position during movement in a defined physical space through player interaction with tasks displayed in a computer generated, specially translated
virtual space for the quantification of the player's movement and agility skills based on
time and distance traveled in the defined physical space.

B. The Related Art

Various instruments and systems have been proposed for assessing a person's
ability to move rapidly in one direction in response to either planned or random visual
or audio cueing. One such system is disclosed in French et al. United States Serial
No. 07/984,337, filed on December 2, 1992, entitled "Interactive Video Testing and
Training System", and assigned to the assignee of the present invention. Therein, a
floor is provided with a plurality of discretely positioned force measuring platforms. A
computer controlled video monitor displays a replica of the floor and audibly and
visually prompts the user to move between platforms in a pseudo-random manner.
The system assesses various performance parameters related to the user's
movements by measuring critical changes in loading associated with reaction time,
transit time, stability time and others. At the end of the protocol, the user is provided
with information related to weight-bearing capabilities including a bilateral comparison
of left-right, forward-backward movement skills. Such a system provides valuable
insight into user's movement abilities in a motivating, interactive environment.

Sensing islands or intercept positions in the form of digital switches or analog
sensors that respond to hand or foot contact when the player arrives at a designated
location have been proposed for providing a variety of movement paths for the user as
disclosed in United States Patent No. 4,627,620 to Yang. The measurement of transit
speeds has also been proposed using discrete optical light paths which are broken at
the designated locations as disclosed in United States Patent No. 4,645,458 to
Williams. However the inability to track the player's movement path continuously inhibits the development of truly interactive games and simulations. In these configurations, the actual position of the player between positions is unknown inasmuch as only the start and finish positions are determined. Most importantly, the requirement that the player move to designated locations is artificial and detracts from actual game simulation in that an athlete rarely undertakes such action, rather the athlete moves to a visually determined interception path for the particular sports purpose.

For valid testing of sports specific skills, many experts consider that, in addition to unplanned cueing, it is important that the distances and directions traveled by the player be representative of actual game play. It is thus desirable to have the capability to measure transit speeds over varying vector distances and directions such that the results can be of significant value to the coach, athletic trainer, athlete and clinician. It is also important to detect bilateral asymmetries in movement and agility so as to enable a clinician or coach to develop and assess the value of remedial training or rehabilitation programs. For example, a rehabilitating tennis player may move less effectively to the right than to the left due to a left knee injury, i.e. the "push off" leg. A quantitative awareness of this deficiency would assist the player in developing compensating playing strategies, as well as the clinician in developing an effective rehabilitation program.

In actual competition, a player does not move to a fixed location, rather the player moves to an intercept position determined visually for the purpose of either contacting a ball, making a tackle or like athletic movement. Under such conditions, it
will be appreciated that there are numerous intercept or avoidance paths available to the player. For example, a faster athlete can oftentimes undertake a more aggressive path whereas a slower athlete will take a more conservative route requiring a balancing of time and direction to make the interception. Successful athletes learn, based on experience, to select the optimum movement paths based on their speed, the speed of the object to be intercepted and its path of movement. Selecting the optimum movement path to intercept or avoid is critical to success in many sports, such as a shortstop in baseball fielding a ground ball, a tennis player returning a volley, or ball carrier avoiding a tackler.

None of the foregoing approaches spatially represents the instantaneous position of the player trying to intercept or avoid a target. One system for displaying the player in a game simulation is afforded in the Mandela Virtual World System available from The Vivid Group of Toronto, Ontario, Canada. One simulation is hockey related wherein the player is displayed on a monitor superimposed over an image of a professional hockey net using a technique called chroma-keying of the type used by television weather reporters. Live action players appear on the screen and take shots at the goal which the player seeks to block. The assessment provided by the system is merely an assessment of success, either the shot is blocked or, if missed, a goal is scored. This system uses a single camera and is accordingly unable to provide quantification of distance traveled, velocities or other time-vector movement information, i.e. physics-based information.

Accordingly, it would be desirable to provide an assessment system in an environment representative of actual conditions for the assessment of relevant
movement skills that enable the player to view changes in his actual physical position in real-time, spatially correct, constantly changing interactive relationship with a challenge or task.

VI. SUMMARY OF THE INVENTION

The present invention overcomes the limitations of the aforementioned approaches by providing an assessment system wherein the player can execute movement paths without a confining field, i.e. fixed movement locations and while viewing progress toward completing a simulated task in a spatially correct relationship with the virtual objective being sought and have physics-based output information for undertakings.

The assessment system of the present invention provides an accurate measurement of movement and agility skills such that the results can be reported in absolute vectored and scalar units related to time and distance in a sport-specific simulation. Herein, the player is not required to move between fixed ground locations. Rather the player moves to intercept or avoid an object based on visual observations of his real-time constantly changing spatial relationship with the computer-generated object.

The present invention also provides a movement skills assessment system operable without a confining field that tracks the player's position continuously in real-time and not merely between a starting and finishing position. The system includes a wireless position tracker coupled to a personal computer. The computer is coupled to a viewing monitor that displays a computer generated virtual space in 4 dimension
space-time with a player icon representing the instantaneous position of the player in scaled translation to the position of the player in a defined physical space where the activity is undertaken. Interactive software displays a protagonist, defined as a moving or stationary object or entity, the task of the player being to intercept or avoid, collide or elude, the protagonist by movement along a path selected by the player, not a path mandated by hardware. The software defines and controls an interactive task and upon completion assesses the ability of the player to complete the task based on distance traveled and elapsed time in the defined physical space. As the movement sequence continues, velocity vectors are measured for each movement segment and processed to compare velocity related information in all directions as well as measurement of elapsed times or composite speeds.

In the preferred embodiment, the intensity of physical activity is quantified in that energy consumed (calories burned), acceleration, and other measurements are presented, based on user-supplied data such as weight.

The system has applications in sports, commercial fitness and medical rehabilitation wherein output and documentation of vectored, physics-based information is desired.

VII. BRIEF DESCRIPTION OF THE DRAWINGS

The above and other objects, advantages and features of the present invention will become apparent from the following description taken in conjunction with the accompanying drawings in which:
Figure 1 is a schematic view of a testing and training system in accordance with the invention;

Figure 2 is representative monitor display;

Figure 3 is a graphical representation of simulated movement skills protocol for the system of Figure 1;

Figure 4 is a graphical representation of a simulated agility skills protocol for the system of Figure 1;

Figure 5 is a graphical representation of a simulated task for the system; and

Figures 6 and 7 are software flow charts of a representative task for the system.

Figures 8 and 9 are software flow charts for the preferred embodiment.

VIII. DETAILED DESCRIPTION OF THE INVENTION

A. The Invention Generally

Referring to the drawings for the purposes of describing the invention embodiments, there is shown in Figure 1 an interactive, virtual reality testing and training system 10 for assessing movement and agility skills without a confining field. The system 10 comprises a three dimensionally defined physical space 12 in which the player moves, a pair of laterally spaced wireless optical sensors 14, 16 coupled to a processor 18 which comprises the wireless position tracking system. The processor 18 provides a signal along line 20 via the serial port to a personal computer 22 that, under the control of associated software 24, provides a signal to a large screen video monitor 28. The computer 22 is operatively connected to a printer 29, such as a
Hewlett Packard Desk Jet 540, for outputting data related to testing and training sessions.

Referring additionally to Figure 2, the monitor 28 displays a computer generated, defined virtual space 30 which is a scaled translation of the defined physical space 12. The position of the player in the physical space 12 is represented and correctly referenced in the virtual space 30 by a player icon 32 and interacts with a protagonist icon 34 in the performance of varying tasks or games to be described below.

The system 10 assesses and quantifies agility and movement skills by continuously tracking the player in the defined physical space 12 through continuous measurement of Cartesian coordinate positions. By scaling translation to the virtual space 30, the player icon 32 is represented in a spatially correct position and can interact with the protagonist icon 34 such that movement related to actual distance and time required by the player 36 to travel in the physical space 12 can be quantified.

The defined physical space 12 may be any available area, indoors or outdoors of sufficient size to allow the player to undertake the movements for assessing and quantifying distance and time measurements relevant to the player's conditioning, sport and ability. A typical physical space 12 may be an indoor facility such as a basketball or handball court where about a 20 foot by 20 foot area with about a 10 foot ceiling clearance can be dedicated for the training and testing. Inasmuch as the system is portable, the system may be transported to multiple sites for specific purposes. For relevant testing of sports skills on outdoor surfaces, such as football or baseball, where the player is most relevantly assessed under actual playing
conditions, i.e. on a grass surface and in athletic gear, the system may be transported to the actual playing field for use.

The optical sensors 14, 16 and processor 18 may take the form of commercially available tracking systems. Preferably the system 10 uses an optical sensing system available as a modification of the DynaSight system from Origin Instruments of Grand Prairie Texas. Such a system uses a pair of optical sensors, i.e. trackers, mounted about 30 inches apart on a support mast centered laterally with respect to the defined physical space 12 at a distance sufficiently outside the front boundary 40 to allow the sensors 14, 16 to track movement in the desired physical space. The processor 18 communicates position information to an application program in a host computer through a serial port. The host computer is provided with a driver program available from Origin which interfaces the DynaSight system with the application program. The sensors, operating in the near infrared frequency range, interact with passive or active reflector(s) worn by the player. The sensors report target positions in three dimensions relative to a fiducial mark midway between the sensors. The fiducial mark is the origin of the default coordinate system.

Another suitable system is the MacReflex Motion Measurement System from Qualisys. Any such system should provide an accurate determination of the players location in at least two coordinates and preferably three.

In the described embodiment, the player icon 32 is displayed on the monitor 28 in the corresponding width, lateral x axis, height, y axis and depth, or fore-aft z axis and over time t, to create a 4 dimensional space-time virtual world. For tasks involving vertical movement, tracking height, y axis, is required. The system 10 determines the
coordinates of the player 36 in the defined physical space 12 in essentially real time and updates current position without any perceived lag between actual change and displayed change in location in the virtual space 30, preferably at a sampling rate of about 20 to 100 Hz.

The monitor 28 should be sufficiently large to enable the player to view clearly virtual space 30. The virtual space 30 is a spatially correct representation of the physical space as generated by the computer 22. For a 20 foot by 20 foot working field, a 27 inch diagonal screen or larger allows the player to perceptively relate to the correlation between the physical and virtual spaces. An acceptable monitor is a Mitsubishi 27" Multiscan Monitor.

The computer 22 receives the signal for coordinates of the player's location in the physical space 12 from the detector 18 and transmits a signal to the monitor 28 for displaying the player icon in scaled relationship in the virtual space 30. An acceptable computer is a Compaq Pentium PC. In other words, the player icon 32 is always positioned in the computer-generated virtual space 30 at the x, y, z coordinates corresponding to the player's actual location in the physical space 12. As the player 36 changes location within the physical space 12, the players icon is repositioned accordingly in the virtual space 30.

To create tasks that induce the player 36 to undertake certain movements, a protagonist icon 34 is displayed in the computer-generated virtual space 30 by the computer software 24. The protagonist icon 34 serves to induce, prompt and lead the player 36 through various tasks, such as testing and training protocols in an interactive game-like format that allows the assessment and quantification of movement and
agility skills related to actual distance traveled and elapsed time in the physical space 12 to provide physics-based vectored and scalar information.

The protagonist icon 34 is interactive with the player 36 in that the task is completed when the player icon 32 and the protagonist icon 34 occupy the same location, i.e. interception, or attain predetermined separation, i.e. evasion. As used herein the protagonist icon is the graphic representation with which the player interacts, and defines the objective of the task. Other collision-based icons, such as obstacles, barriers, walls and the like may embellish the task, but are generally secondary to the objective being defined by the protagonist.

The protagonist icon 34 may have varying attributes. For example, the protagonist icon may be dynamic, rather than stationary, in that its location changes with time under the control of the software thereby requiring the player to determine an ever changing interception or evasion path to complete the task.

Further, the protagonist icon can be intelligent, programmed to be aware of the player's position in the computer-generated virtual space 30 and to intercept or evade according to the objectives of the task. Such intelligent protagonist icons are capable of making course correction changes in response to changes in the position of the player icon 32 in much the same manner as conventional video games wherein the targets are responsive to the icon under the player's control, the difference being that the player's icon does not correspond the player's actual position in a defined physical space.

The foregoing provides a system for assessing movement skills and agility skills. Movement skills are generally characterized in terms of the shortest time to
achieve the distance objective. They can be further characterized by direction of
movement with feedback, quantification and assessment being provided in
absolute units, i.e. distance/time unit, or as a game score indicative of the player's
movement capabilities related to physics-based information including speed, velocity,
acceleration, deceleration and displacement. Agility is generally characterized as the
ability to quickly and efficiently change body position and direction while undertaking
specific movement patterns. The results also are reported in absolute units, with
success determined by the elapsed time to complete the task.

The software flow chart for the foregoing tasks is shown in Figures 6 and 7. At
the start 80 of the assessment, the player is prompted to Define Protagonists 82. The
player may select the intelligence level, number, speed and size of the protagonists to
reside in the selected routine. Thereafter the player is prompted to Define Obstacles
84, i.e. static vs. dynamic, number, seed, size and shape. The player is then
prompted to Define Objectives 86, i.e. avoidance or interception, scoring parameters,
and goals, to complete the setup routine.

To start the task routine, the player is prompted to a starting position for the
task and upon reaching this position, the protagonist(s) and the obstacle(s) for the task
are generated on the display. The protagonist moves on the display, 90, in a
trajectory dependent on the setup definition. For an interception routine, the player
moves in a path which the player determines will result in the earliest interception point
with the protagonist in accordance with the player's ability. During player movement,
the player icon is generated, and continually updated, in scaled translation in the
virtual space to the player's instantaneous position in the defined physical space.
Movement continues until player contact, 92, and interception, 94, or until the protagonist contacts a boundary of the virtual space corresponding to the boundary of the defined physical space, 96. In the former case, if interception has occurred, a new protagonist appears on a new trajectory, 97. The player icon's position is recorded, 98, the velocity vectors calculated and recorded, and a score or assessment noted on the display. The system then determines if the task objectives have been met, 100, and for a single task, the final score is computed and displayed, 102, as well as information related to time and distance traveled in completing the task, and the session ends, 104.

In the event, the player does not intercept the protagonist icon prior to the later contacting a virtual space boundary corresponding to the boundary on the defined physical space, the direction of the protagonist is changed dependent on the setup definition, and the pursuit of the protagonist by the player continues as set forth above.

Concurrently with the player pursuit, in the event that obstacles have been selected in the setup definition, the same are displayed, 110, and the player must undertake a movement path to avoid these obstacles. For a single segment task, if the player contacts the obstacle, 112, the obstacle is highlighted, 114, and the routine is completed and scored as described above. In the event a moving obstacle was selected in the setup definition, if the obstacle strikes a boundary, 116, the obstacle's direction is changed, 118, and the task continues.

For a multiple segment task, if the obstacle is contacted, the protagonist's direction changes and the movements continue. Similarly, upon interception for a multiple segment task, a new protagonist trajectory is initiated and the obstacles also
may be reoriented. The routine then continues until the objectives of the task have been met and the session completed.

The tasks are structured to require the player to move forward, backward, left and right, and optionally vertically. The player's movement is quantified as to distance and direction dependent on the sampling rate and the update rate of the system. For each sampling period, the change in position is calculated. At the end of the session, these samples are totaled and displayed for the various movement vectors.

For an avoidance task wherein the objective of the session is to avoid a protagonist seeking to intercept the player, the aforementioned is appropriately altered. Thus if the player is intercepted by the protagonist, the session ends for a single segment task and the time and distance related information is calculated and displayed. For multiple segment tasks, the protagonist trajectory has a new origin and the session continues for the defined task until completed or terminated.

An example of a functional movement skills test is illustrated in Figure 3 by reference to a standard three hop test. Therein the player 36 or patient stands on one leg and performs three consecutive hops as far as possible and lands on the same foot. In this instance the player icon 32 is displayed at the center of the rear portion of the computer-generated virtual space 30 a position in scaled translation to the position of the player 36 in the defined physical space 12. Three hoops 50, protagonist icons, appear on the display indicating the sequence of hops the player should execute. The space of the hoops may be arbitrarily spaced, or may be intelligent, based on standard percentile data for such tests, or on the best or average past performances of the player. In one embodiment, the player 36 is prompted to the starting position 52.
When the player reaches such position, the three hoops 50 appear representing the 50th percentile hop distances for the player's classification, and after a slight delay the first hoop is highlighted indicating the start of the test. The player then executes the first hop with the player's movement toward the first hoop being depicted in essentially real-time on the display. When the player lands after completion of the first hop, this position is noted and stored on the display until completion of the test and the second hoop and third hoop are sequentially highlighted as set forth above. At the end of the three hops, the player's distances will be displayed with reference to normative data.

A test for agility assessment is illustrated in Figure 4 for a SEMO Agility Test wherein the generated virtual space 30 is generally within the confines of a basketball free throw lane. Four cones 60, 62, 64, 66 are the protagonist icons. As in the movement skills test above, the player 36 is prompted to a starting position 68 at the lower right corner. When the player 36 reaches the starting position in the defined physical space the left lower cone 62 is highlighted and the player side steps leftward thereto while facing the display. After clearing the vicinity of cone 62, the fourth cone 66, diagonally across at the front of the virtual space 30 is highlighted and the player backpedals toward and circles around cone 66. Thereafter the player sprints toward the starting cone 60 and circles the same and then backpedals to a highlighted third virtual cone 64. After circling the cone 64, cone 66 is highlighted and the player sprints toward and circles the cone 66 and then side steps to the starting position 68 to complete the test. In the conventional test, the elapsed time from start to finish is used as the test score. With the present invention, however, each leg of the test can be
individually reported, as well as forward, backward and side to side movement capabilities.

As will be apparent from the above embodiment, the system provides a unique measurement of the play's visual observation and assesses skills in a sport simulation wherein the player is required to intercept or avoid the protagonist based on visual observation of the constantly changing spatial relationship with the protagonist. Additionally, excursions in the Y-plane can be quantified during movement as a measure of an optimal stance of the player.

The foregoing and other capabilities of the system are further illustrated by reference to Figure 5. Therein, the task is to intercept targets 70, 71 emanating from a source 72 and traveling in a straight line trajectories T1, T2. The generated virtual space 30 displays a plurality of obstacles 74 which the player must avoid in establishing an interception path with the target 70. The player assumes in the defined physical space a position which is represented on the generated virtual space as position P (x1, y1, z1) in accurately scaled translation therewith. As the target 70 proceeds along trajectory T1, the player moves along a personally determined path in the physical space which is indicated by the dashed lines in the virtual space to achieve an interception site coincident with the instantaneous coordinates of the target 70, signaling a successful completion of the first task. This achievement prompts the second target 71 to emanate from the source along trajectory T2. In order to achieve an intercept position for this task, the player is required to select a movement path which will avoid contact or collision with virtual obstacle 74. Thus, within the capabilities of the player, a path shown by the dashed lines is executed in the defined
physical space and continually updated and displayed in the virtual space as the player intercepts the protagonist target at position P (x3, y3, z3) signaling completion of the second task. The assessment continues in accordance with the parameters selected for the session, at the end of which the player receives feedback indicative of success, i.e. scores or critical assessment based on the distance, elapsed time for various vectors of movement.

Another protocol is a back and forth hop test. Therein, the task is to hope back and forth on one leg over a virtual barrier displayed in the computer-generated virtual space. The relevant information upon completion of the session would be the amplitude measured on each hop which indicates obtaining a height sufficient to clear the virtual barrier. Additionally, the magnitude of limb oscillations experienced upon landing could be assessed. In this regard, the protocol may only measure the vertical distance achieved in a single or multiple vertical jump.

The aforementioned system accurately, and in essentially real-time, measures the absolute three dimensional displacements over time of the body's center of gravity when the sensor marker is appropriately located on the player's mass center. Measuring absolute displacements in the vertical plane as well as the horizontal plane enables assessment of both movement skills and movement efficiency.

In many sports, it is considered desirable for the player to maintain a consistent elevation of his center of gravity above the playing surface. Observation of excursions of the player's body center of gravity in the fore-aft (Z) during execution of tests requiring solely lateral movements (X) would be considered inefficient. For example,
displacements in the player’s Y plane during horizontal movements that exceed certain preestablished parameters could be indicative of movement inefficiencies.

In a further protocol using this information, the protagonist icon functions as an aerobics instructor directing the player through a series of aerobic routines. The system can also serve as an objective physiological indicator of physical activity or work rate during free body movement in essentially real time. Such information provides three benefits: (1) enables interactive, computer modulation of the workout session by providing custom movement cues in response to the player’s current level of physical activity; (2) represents a valid and unique criteria to progress the player in his training program; and (3) provides immediate, objective feedback during training for motivation, safety and optimized training. Such immediate, objective feedback of physical activity is currently missing in all aerobics programs, particularly unsupervised home programs.

B. Specific Embodiments

In certain embodiments of the present invention, performance-related physical activity parameters, including calories burned, are monitored and quantified. The repetitive drudgery of conventional stationary exercise equipment that currently measures calories, heart rate, etc. is replaced by the excitement of three-dimensional movement in interactive response to virtual reality challenges presented on the monitor of the inventive system. Excitement is achieved in part by the scaling transformation achieved by the present invention, through which positional changes by the user moving in real space are represented in scaled relationship in the virtual world presented on the monitor.
One embodiment quantifies performance-related parameters including those related to:

(a) determining and training a user’s optimal dynamic posture;
(b) the relationship between heartrate and physical activity;
(c) quantifying quickness, i.e., acceleration and deceleration; and
(d) quantifying energy expenditure during free ranging activities.

It is especially significant that the user’s energy expenditure may be expressed as calories burned, inasmuch as this is a parameter of primary concern to many exercisers. The advantage of the inventive system is that a variety of environments in the virtual world displayed on the monitor can prompt any desired type and intensity of physical activity, achieving activity and energy expenditure goals in an ever-changing and challenging environment, so that the user looks forward to, rather than dreads, exercise, testing, or therapy sessions.

1. Definitions and Formulae Relating to Quantification of Intensity of Physical Activity

The following terms with the indicated meanings and formulae are used in respect of the inventive system.

Measurement of motion (movement in three planes) is used to quantify work and energy expenditure. Quantities such as force, acceleration and power, defined below, are dependent on the rate of change of more elementary quantities such as body position and velocity. The energy expenditure of an individual is related to the movement of the individual while performing the invention protocols.

a. Motion-Related Measurements
First, with the target (retro-reflector) placed at the center of gravity (CG) point near the midsection of an individual 36 under study, such individual being referred to herein as the subject, user, or player), an activity or protocol is delivered by the invention’s computer 22. For example it may be a simple repetitive motion performed at a uniform pace, it may be a rhythmic motion such as continuous jumping, or it could consist of a side-to-side motion; any representative movement is satisfactory.

In any case, each of these simple examples of an embodiment of the invention protocols consists of repetitive bilateral motion along a line. More complex examples of physical activities can be readily constructed by varying the tempo of a repetitive activity or by combining the up-down, side-to-side, and front-to-back motions of several simple activities into a general blended sequence of movements, either planned or unplanned.

The concept that a complex motion can be considered as a combination of simple bilateral movements in any of three directions is convenient since this approach allows focus on elementary movements with subsequent adding of the effects of these simple components. Such concept relates to the ability to monitor continuously the movement of the individual to measure the resultant energy expenditure.

The ability of this embodiment to accurately measure a subject's movement rests on being able to determine his or her position and velocity at arbitrary points of time. For a given point in time, a position is measured directly. The invention’s sampling rate is sufficiently fast to allow accurate measurements to be made at very
closely spaced intervals of time. By knowing an individual's position at arbitrary points along its path the velocity can be calculated.

In the present embodiment, positions can be used to determine velocity along a movement path: given the position of the individual at various instances of time, the embodiment can obtain the velocity in several ways. One method is to choose a point and calculate its velocity as being the result of dividing the distance between it and the next point by the time difference associated with those points. This is known as a finite difference approximation to the true velocity. For small spacing between points, it is highly accurate.

If \( D \) is the distance between consecutive points and \( T \) equal the time period to travel the distance \( D \), then the velocity \( V \) is given by the following rate of change formula

\[
V = \frac{D}{T},
\]

where \( V \) has the units of meters per second, m/s.

In three dimensional space, \( D \) is computed by taking the change in each of the separate bilateral directions into account. If \( dX, dY, dZ \) represents the positional changes between the successive bilateral directions, then the distance \( D \) is given by the following formula

\[
D = \sqrt{dX^2 + dY^2 + dZ^2},
\]

where "sqrt" represents the square root operation. The velocity can be labeled positive for one direction along a path and negative for the opposite direction. This is, of course, true for each of the bilateral directions separately.

This finite difference approximation procedure can also be used to calculate
the acceleration of the object along the path. This is accomplished by taking the change in velocity between two consecutive points and dividing by the time interval between points. This gives an approximation to the acceleration $A$ of the object which is expressed as a rate of change with respect to time as follows

$$A = \frac{dV}{dT},$$

where $dV$ is the change in velocity and $T$ is the time interval. Acceleration is expressed in terms of meters per second per second. The accuracy of this approximation to the acceleration is dependent on using sufficiently small intervals between points.

As an alternate to using smaller position increments to improve accuracy, more accurate finite difference procedures may be employed. This embodiment obtains positional data with accuracy within a few centimeters over time intervals of approximately .020 seconds, so that errors are assumed to be negligible.

In contrast to the finite difference approach, the positional data could be fitted by spline curves and treated as continuous curves. The velocity at any point would be related to the tangent to the individual's path using derivative procedures of standard calculus. This would give a continuous curve for the velocity from which a corresponding curve could be obtained for the acceleration of the individual.

In any case, the determination of the individual's acceleration provides a knowledge of the force $F$ it experiences. The force is related to the mass $M$, given in kilograms, and acceleration by the formula

$$F = MA.$$

This is a resultant formula combining all three components of force and acceleration,
one component for each of the three bilateral directions. The international standard of force is a newton which is equivalent to a kilogram mass undergoing an acceleration of one meter per second per second. This embodiment requires that the individual enter bodyweight (for MASS) prior to playing.

The effect of each component can be considered separately in analyzing an individual's movement. This is easily illustrated by recognizing that an individual moving horizontally will be accelerated downward due to gravity even as it is being decelerated horizontally by air drag. The effects of forces can be treated separately or as an aggregate. This allows one the option to isolate effects or lump effects together. This option provides flexibility in analysis.

b. Energy Expenditure Measurements

Energy and work may be measured by one embodiment. The energy expended by an individual in the inventive system can be derived from work. The mechanical work is calculated by multiplying the force acting on an individual by the distances that the individual moves while under the action of force.

Different individuals performing the same activity expend different amounts of heat due to differences in body mass, gender, and other factors. As indicated above, mechanical work done in an activity is determined in the present invention system by monitoring motion parameters associated with that activity. Total energy expenditure can be derived from known work-to-calories ratios.

2. Protocols And Their Use in the Preferred Embodiment

Four protocols used in embodiments of the inventive system enable quantification of performance-related parameters.
a. **Dynamic Posture**

"Dynamic Posture" means that athletic stance maintained during sport-specific activity that maximizes a player's readiness for a specific task. Examples are the slight crouches or "ready" position of a soccer goalie or a football linebacker.

Testing or training of dynamic posture is achieved by having the user initially assume the desired position and then tracking, in essentially real-time, displacements in the Y (vertical) plane during interactive protocols. Such Y plane displacements accurately reflect vertical fluctuations of that point on the body on which the reflective marker is placed, for example, the hipline, which is often referred to as the CG point.

In one embodiment, it is important both to determine, and train in, optimal dynamic posture. The optimal dynamic posture during sport-specific activities is determined as follows:

1. A retro-reflective marker is mounted at the athlete's CG point,
2. The invention's computer 22 measures in real-time displacements of the athlete's CG (Y-plane excursions) as he responds to interactive, sport-specific protocols.
3. The invention's computer 22 calculates in essentially real-time the athlete's movement velocities and/or accelerations during performance of sport-specific protocols.
4. The invention calculates the athlete's most efficient dynamic posture defined as that CG elevation that produces maximum velocities and/or accelerations/decelerations for the athlete.
5. The invention provides numerical and graphical feedback of results.
Once the optimal dynamic posture is determined, training optimal dynamic posture is achieved by:

(1) A retro-reflective marker is mounted at the athlete's CG point,

(2) The athlete 36 assumes the dynamic posture that he wishes to train,

(3) The invention is initialized for this CG position,

(4) The invention provides varying interactive movement challenges over sport-specific distances and directions, including unplanned movements,

(5) Y-plane excursions that exceed the pre-set threshold or window will generate real-time feedback of such violations for the user.

(6) The invention provides real-time feedback of compliance with the desired dynamic posture during performance of the protocols.

The invention uses unplanned, interactive game-like movement challenges requiring sport-specific responses. The participant will move most effectively during stopping, starting and cutting activities if he assumes and maintains his optimum Center of Gravity (CG) elevation. Additional movement efficiencies are achieved by the player by minimizing CG elevation excursions. The invention is capable of tracking in essentially real-time, the participant's CG elevation by monitoring Y plane displacements. During the training phase, the participant will be provided with real-time feedback of any Y plane excursions exceeding targeted ranges.

b. Heart Rate/Physical Activity Relationship

The relationship between heart rate and Physical Activity of the subject during performance of the protocols is quantified by the present invention. Heart rate is
measured by a commercially available wireless (telemetry) device (36A, Figure 2) in essentially real-time. Conventional cardiovascular exercise equipment attempts to predict caloric expenditure from exercise heart rate. Real time monitoring of heart rate is an attempt to infer the users' level of physical activity. But, heart rate is affected by factors other than physical activity such as stress, ambient temperature and type of muscular contraction, so the ratio or relationship between the two could be enlightening to the coach, athlete or clinician. For example, physical training lowers the heart rate at which tasks of a given energy cost are performed.

Prior art applications have attempted to measure these two parameters simultaneously in an attempt to validate one of the measurement constructs as a measure of physical activity. In all such cases though, such measurements were not in real-time; they were recorded over time and did not employ position tracking means nor involve interactive protocols used in the inventive system.

In another embodiment, simultaneous assessment and modulation of physical activity and heart rate is achieved as follows:

1. Subject 36 places a retro-reflective marker at his CG point.

2. A wireless heart-rate monitor (36A, Figure 2) is worn on the subject 36 which communicates in real-time with the invention's computer 22.

3. Subject 36 enters desired target heart-rate range. Entering desired target heart-rate range should be qualified as optional.

4. The invention provides interactive, functional planned and unplanned movement challenges over varying distances and directions.

5. The invention provides real-time feedback of compliance with selected heart-rate zone during performance of these protocols.

6. The invention provides a graphical summary of the relationship or correlation between heart-rate at each moment of time and free-body physical activity.
c. **Acceleration and Deceleration Quantification**

Assessment and quantification of *movement skills* during unplanned movement protocols over sport-specific distances is presented by the present invention. Movement skills are defined as the quantification of bi-lateral vector performance, i.e., how well a subject 36 moves left vs. right, etc. The present invention teaches the measurement of *accelerations/decelerations*, since it can sample positional changes approximately every 10 to 30 ms.

In still another embodiment, quantification of bi-lateral vector accelerations and decelerations are achieved as follows:

1. A retro-reflective marker is mounted at the athlete's CG point,
2. The invention tracks at sufficient sampling rate the athlete's movement in three-degrees-of-freedom during his performance of sport-specific protocols, including unplanned movements over various vector distances,
3. The invention calculates in essentially real-time the athlete's movement accelerations and decelerations,
4. The invention categorizes each movement leg to a particular vector,
5. The invention provides numerical and graphical feedback of bi-lateral performance.

D. **Energy Expenditure**

Quantification of the intensity of free-ranging physical activity as expressed in kilocalories per minute, and the total energy expended, is derived from movement data collected as the subject moves in response to prompts from the monitor, personal data such as weight inputted by the subject, and conventional conversion formulae.
During performance of the above protocols, the inventive system can measure the intensity, i.e., strenuousness or energy cost of physical activity during free ranging (functional) activities, expressed in calories per minute, distance traveled per unit of time.

Energy expenditure can be derived from the subject's movement data during performance of free-ranging activities. Well known laboratory instrumentation can be employed to ascertain the coefficient or conversion factor needed to convert work or power or distance derived from the movement data to calories expended. Oxygen uptake, expressed in milliliters per kilogram per minute can determine the caloric expenditure of physical activity and is considered the "gold standard" or reference when evaluating alternative measures of physical activity. The most precise laboratory means to determine oxygen uptake is through direct gas analysis, which would be performed on representative subject populations during their execution of the invention's protocols with a metabolic cart, which directly measures the amount of oxygen consumed. Such populations would be categorized based on age, gender and weight.

3. Software

The software flow chart for the tasks of an illustrative embodiment is shown in Figures 8 and 9. After the start 80 of the assessment, the user is prompted to DEFINE PLAYER ICON (81). This is where the player's body weight, sex, etc., other information necessary to calculate calories, is entered. The player is prompted to Define Protagonists 82. The player may select the intelligence level, number, speed and size of the protagonists to reside in the selected routine. Thereafter the
player is prompted to Define Obstacles 84, i.e., static vs. dynamic, number, speed, size and shape. The player is then prompted to Define Objectives 86, i.e., avoidance or interception, scoring parameters, and goals, to complete the setup routine. As part of DEFINE OBJECTIVES (86), the players 3-D path boundaries should be programmed, the reference frame of play, i.e., 1st person, 3rd person. The player is then prompted by PATH VIOLATION (86A). If yes then provide audio/visual cues alarms and record player’s icon change in position else just record player’s icon change in position. The OBJECTIVES MET decision block should point here if NO.

To start the task routine, the player is prompted to a starting position for the task and upon reaching this position, the protagonist(s) and the obstacle(s) for the task are generated on the display. The protagonist moves on the display, 90, in a trajectory dependent on the setup definition. For an interception routine, the player moves in a path which the player determines will result in the earliest interception point with the protagonist in accordance with the player’s ability. During player movement, the player icon is generated, and continually updated, in scaled translation in the virtual space to the player’s instantaneous position in the defined physical space. Movement continues until player contact, 92, and interception, 94, or until the protagonist contacts a boundary of the virtual space corresponding to the boundary of the defined physical space, 96. In the former case, if interception has occurred, a new protagonist appears on a new trajectory, 97. The player icon’s position is recorded, 98, the velocity vectors calculated and recorded, and a score of assessment noted on the display. The system then determines if the task objectives
have been met, 100, and for a single task, the final score is computed and displayed, 102, and calories burned in calculated, as well as information related to time and distance traveled in completing the task, and the session ends, 104.

In the event the player does not intercept the protagonist icon prior to the later contacting a virtual space boundary corresponding to the boundary on the defined physical space, the direction of the protagonist is changed dependent on the setup definition, and the pursuit of the protagonist by the player continues as set forth above.

Concurrently with the player pursuit, in the event that obstacles have been selected in the setup definition, the same are displayed, 110, and the player must undertake a movement path to avoid these obstacles. For a single segment task, if the player contacts the obstacle, 112, the obstacle is highlighted, 114, and the routine is completed and scored as described above. In the event a moving obstacle was selected in the setup definition, if the obstacle strikes a boundary, 116, the obstacle's direction is changed, 118, and the task continues.

For a multiple segment task, if the obstacle is contacted, the protagonist's direction changes and the movements continue. Similarly, upon interception for a multiple segment task, a new protagonist trajectory is initiated and the obstacles also may be reoriented. The routine then continues until the objectives of the task have been met, and the session completed.

The tasks are structured to require the player to move forward, backward, left and right, and optionally vertically. The player's movement is quantified as to distance and direction dependent on the sampling rate and the update rate of the
system. For each sampling period, the change in position is calculated. At the end of the session, these samples are totaled and displayed for the various movement vectors.

For an avoidance task wherein the objective of the session is to avoid a protagonist seeking to intercept the player, the aforementioned is appropriately altered. Thus if the player is intercepted by the protagonist, the session ends for a single segment task and the time and distance related information is calculated and displayed. For multiple segment tasks, the protagonist trajectory has a new origin and the session continues for the defined task until completed or terminated.

More particularly, the following source code generally enables the accomplishment of an illustrative embodiment of inventive systems.

ScaleHeight = 6945
ScaleWidth = 9645
Top = 285
Width = 9765
WindowState = 2 'Maximized
Begin B.OptionButton Option1
  BackColor = &H00FFFFFF&
  Caption = "Fixed Player"
BeginProperty Font
    name = "Arial"
    charset = 0
    weight = 700
    size = 12
    underline = 0 'False
    italic = 0 'False
    strikethrough = 0 'False
EndProperty
Height = 495
Index = 1
Left = 120
TabIndex = 11
Top = 4680
Width = 1815
End
Begin B.OptionButton Option1
BackColor = &H00FFFFFF&
Caption = "Free Player"
BeginProperty Font
   name = "Arial"
   charset = 0
   weight = 700
   size = 12
   underline = 0 'False
   italic = 0 'False
   strikethrough = 0 'False
EndProperty
Height = 495
Index = 0
Left = 120
TabIndex = 10
Top = 3720
Value = -1 'True
Width = 1695
End
Begin B.CheckBox Oponent_Visible
Appearance = 0 'Flat
BackColor = &H00FFFFFF&
Caption = "Enable Opponent"
BeginProperty Font
   name = "Arial"
   charset = 0
   weight = 700
   size = 12
   underline = 0 'False
   italic = 0 'False
   strikethrough = 0 'False
EndProperty
ForeColor = &H80000008&
Height = 495
Left = 120
TabIndex = 6
Top = 5880
Width = 2535
End
Begin B.PictureBox BackWall
Appearance = 0 'Flat
BackColor = &H00FFFF00&
BorderStyle = 0 'None
ClipControls = 0 'False
FillColor = &H00FFFF00&
FillStyle = 0 'Solid
ForeColor = &H00FFFFFF&
Height = 615
Left = 5280
ScaleHeight = 615
ScaleWidth = 615
TabIndex = 5
Top = 120
Visible = 0 'False
Width = 615
End

Begin B.PictureBox Oponent
Appearance = 0 'Flat
AutoSize = -1 'True
BackColor = &H00FFFFFF&
BorderStyle = 0 'None
Enabled = 0 'False
FillColor = &H0000000F&
FillStyle = 0 'Solid
BeginProperty Font
name = "MS Sans Serif"
charset = 0
weight = 400
size = 8.25
underline = 0 'False
italic = 0 'False
strikeThrough = 0 'False
EndProperty
ForeColor = &H00000000&
Height = 615
Index = 0
Left = 2760
ScaleHeight = 615
ScaleWidth = 615
TabIndex = 4
Top = 120
Visible = 0 'False
Width = 615
End

Begin B.PictureBox Target
Appearance = 0 'Flat
BackColor = &H00FFFFFF&
BorderStyle = 0 'None
FillColor = &H00FF80FF&
FillStyle = 0 'Solid
ForeColor = &H00000000&
Height = 645
Left = 3600
Picture = "ORIGIN.frx":0446
ScaleHeight = 645
ScaleWidth = 645
TabIndex = 3
Top = 120
Visible = 0 'False
Width = 645
End
Begin B.PictureBox Player_Icon
Appearance = 0 'Flat
BackColor = &H80000005&
BorderStyle = 0 'None
FillColor = &H00F0F0F0&
FillStyle = 0 'Solid
ForeColor = &H00F0F0F0&
Height = 615
Left = 4440
Picture = "ORIGIN.frx":088C
ScaleHeight = 615
ScaleWidth = 615
TabIndex = 2
Top = 120
Visible = 0 'False
Width = 615
End
Begin B.Timer Timer1
Interval = 500
Left = 8040
Top = 0
End
Begin B.Timer Player_Update
Enabled = 0 'False
Interval = 20
Left = 7560
Top = 0
End
Begin Threed.SSPanel Panel3D4
Height = 492
Left = 0
TabIndex = 8
Top = 0
Visible = 0 'False
Width = 732
_Version = 65536
_ExtentX = 1291
_ExtentY = 868
_StockProps = 15
ForeColor = 0
BackColor = 12632256
BeginProperty Font {0BE35203-8F91-11CE-9DE3-00AA004BB851}
    name = "MS Sans Serif"
    charset = 0
    weight = 700
    size = 8.2
    underline = 0 'False
    italic = -1 'True
    strikethrough = 0 'False
EndProperty
BevelWidth = 4
BevelOuter = 1
Alignment = 6
Enabled = 0 'False
Begin B.Label Etime_Disp
    Alignment = 2 'Center
    Appearance = 0 'Flat
    BackColor = &H00C0C0C0&
    Caption = "Etime"
    ForeColor = &H80000008&
    Height = 252
    Left = 120
    TabIndex = 9
    Top = 120
    Width = 492
End

Begin B.Label Label1
    Alignment = 2 'Center
    BackColor = &H00FFFFFF&
    Caption = "TRACKER DEMO"
BeginProperty Font
    name = "Arial"
    charset = 0
    weight = 700
    size = 48
    underline = 0 'False
    italic = 0 'False

35
End Property
ForeColor          = &H00008000&
Height             = 1215
Left               = 480
TabIndex           = 12
Top                = 600
Width              = 8655
End

Begin GraphLib.Graph Graph1
Height             = 1215
Left               = 6000
TabIndex           = 7
Top                = 0
Visible            = 0 'False
Width              = 1455
_Version           = 65536
_ExtentX           = 2566
_ExtentY           = 2143
_StockProps        = 96
BorderStyle        = 1
Enabled            = 0 'False
GraphType          = 10
RandomData         = 1
ColorData          = 0
ExtraData          = 0
ExtraData[]        = 0
FontFamily         = 4
FontSize           = 4
FontSize[0]        = 200
FontSize[1]        = 150
FontSize[2]        = 100
FontSize[3]        = 100
FontStyle          = 4
GraphData          = 0
GraphData[]        = 0
LabelText          = 0
LegendText         = 0
PatternData        = 0
SymbolData         = 0
XPosData           = 0
XPosData[]         = 0
End

Begin MSCommLib.MSComm XYZ_GRAB
Left               = 8520
Top                = 0
_Version           = 65536
_BeginExtentX  =  847
_BeginExtentY  =  847
_SockProps    =  0
_CDTTimeout   =  0
_CommPort     =  1
_CTSTimeout   =  0
_DSRSTimeout  =  0
_DTREnable    =  -1 'True
_Handshaking  =  0
_InBufferSize =  1024
_InputLen     =  0
_Interval     =  1000
_NullDiscard  =  0 'False
_OutBufferSize =  512
_ParityReplace =  "?"
_RThreshold   =  0
_RTSEnable    =  0 'False
_Settings     =  "19200,n,8,1"
_SThreshold   =  0

End
 Begin Threed.SSCommand Quit
  Height      =  735
  Left        =  8520
  TabIndex    =  1
  Top         =  6000
  Width       =  855
  _Version     =  65536
  _ExtentX     =  1508
  _ExtentY     =  1296
  _SockProps   =  78
  Caption      =  "QUIT"
  ForeColor    =  255
  BevelWidth   =  4
  Font3D       =  1
End
 Begin Threed.SSCommand Start
  Height      =  3615
  Left        =  3240
  TabIndex    =  0
  Top         =  3120
  Width       =  3855
  _Version     =  65536
  _ExtentX     =  6800
  _ExtentY     =  6376
  _SockProps   =  78
  Caption      =  "START"
  ForeColor    =  16711680
BeginProperty Font (0BE35203-8F91-11CE-9DE3-00AA004BB851)
    name = "MS Sans Serif"
    charset = 0
    weight = 700
    size = 24
    underline = 0 'False
    italic = 0 'False
    strikethrough = 0 'False
EndProperty
BevelWidth = 4
End
Attribute B_Name = "Form1"
Attribute B_CREATABLE = False
Attribute B_EXPOSED = False

Option Explicit

Private Sub Form_Click()
    Dim i As Integer
    Dim j As Long
    Dim temp As Integer
    Dim Sort As Integer
    Dim angle As String
    Static Right_Side As Integer
    Static Left_Side As Integer

    Player_Update.Enabled = False
    BackWall.Visible = False
    Start.Visible = True
    Start.Enabled = True
    Quit.Visible = True
    Quit.Enabled = True
    Option1(0).Visible = True
    Option1(1).Visible = True
    Option1(0).Enabled = True
    Option1(1).Enabled = True
    Opponent_Visible.Visibile = True
    Opponent_Visible.Enabled = True
    Player_Icon.Visible = False
    Opponent(0).Visible = False
    Target.Visible = False
    Label1.Enabled = True
    Label1.Visible = True
    Form1.Cls
Rem If Moves <> 0 Then
  Rem For i = 0 To Moves
    Rem DoEvents
    Rem Select Case A_Player2TargetDireccion(i)
      Rem Case "1"
        Rem Player2TargetAngle(i) = Abs((A_Delta_Target_Y(i) / (A_Delta_Target_X(i) + 1)) * 57.29578)
      Rem Case "2"
        Rem Player2TargetAngle(i) = 180 + (A_Delta_Target_Y(i) / (A_Delta_Target_X(i) + 1)) * 57.29578
      Rem Case "3"
        Rem Player2TargetAngle(i) = 180 + Abs((A_Delta_Target_Y(i) / (A_Delta_Target_X(i) + 1)) * 57.29578)
      Rem Case "4"
        Rem Player2TargetAngle(i) = 360 + (A_Delta_Target_Y(i) / (A_Delta_Target_X(i) + 1)) * 57.29578
          Rem If Player2TargetAngle(i) = 360 Then
              Rem Player2TargetAngle(i) = 0
            Rem End If
    Rem End Select
    Rem Player2Target_Distance(i) = Sqr((A_Delta_Target_X(i) * A_Delta_Target_X(i) + (A_Delta_Target_Y(i) * A_Delta_Target_Y(i))))
    Rem Player_Transit_Speed(i) = Player2Target_Distance(i) / ((Transit_Time(i) + 1) * 50&)
      Rem Next i
    Rem Player2TargetAngle(Moves + 1) = 361
    Rem Do
      Rem Sort = False
      Rem For i = 0 To Moves
        Rem DoEvents
        Rem If Player2TargetAngle(i) > Player2TargetAngle(i + 1) Then
          Rem temp = Player2TargetAngle(i)
          Rem Player2TargetAngle(i) = Player2TargetAngle(i + 1)
          Rem Player2TargetAngle(i + 1) = temp
          Rem temp = Player_Transit_Speed(i)
          Rem Player_Transit_Speed(i) = Player_Transit_Speed(i + 1)
          Rem Player_Transit_Speed(i + 1) = temp
          Rem Sort = True
        Rem End If
        Rem Next i
      Rem Loop Until Sort = False
    Rem Graph1.Width = Field_Width_Center
    Rem Graph1.Top = 0
    Rem Graph1.Left = 0

39
Rem Graph1.Height = FieldHeight
Rem Graph1.DataReset = 1
Rem For i = 0 To Moves
    Rem Graph1.ThisPoint = i + 1
    Rem Graph1.XPosData = Player2Target_Angle(i)
    Rem Graph1.GraphData = Player_Transit_Speed(i)
Rem Next i
Rem Graph1.DrawMode = 2
Rem Graph1.Refresh
Rem Graph1.Visible = True
Rem XYZ_Grab.InBufferCount = 0
Rem End If
Rem End If
End Sub

Private Sub Form_Load()
    XYZ_Grab.InputLen = 1
    XYZ_Grab.PortOpen = True
    XYZ_Grab.State = False
    XYZ_Update_State = False

    Rem Etime_sec = 0
    Rem eminute = 0
    Rem eisecond = 0
    Rem Etime_Enabled = False

    OponentY_Delta = 50
    OponentX_Delta = 50

    Rem Etime_Disp.Caption = Format$(eminute, ".") & ":" & Format$(eisecond, "00")

    Rem Determine size of playing field
    Form1.WindowState = 2
    Form1.Show
    Field_Width = Form1.Width
    Field_Height = Form1.Height
    Field_Width_Center = Field_Width \\ 2
    Horizon = Field_Height \ 3

    PlayerIcon_X_Offset = 0
    PlayerIcon_Y_Offset = 0
    PlayerIcon_Z_Offset = 0

    Icon_Width_Max = 500
    Icon_Height_Max = 500
    Icon_Dim_Comp = 40
Private Sub Oponent_Paint(Index As Integer)
    Form1.FillColor = &HFFFFFF
    Form1.Circle ((Oponent(0).Left + Old_Oponent_Width_Half), (Oponent(0).Top + (Old_Oponent_Height / 1.5))), Abs(Old_Oponent_Width_Half - Icon_Dim_Comp), &HFFFFFF, , 0.6
    Oponent(0).Cls
    Oponent(0).Move (Oponent_X_Position - (New_Player_Icon_Left_Delta / Oponent_Lateral_Scale)), (Oponent_Y_Position - Oponent_Depth_Scale), Oponent_Width, Oponent_Height
    Form1.FillColor = &HF0F0F0
    Form1.Circle ((Oponent(0).Left + Oponent_Width_Half), (Oponent(0).Top + (Oponent_Height / 1.5))), (Oponent_Width_Half - Icon_Dim_Comp), &HF0F0F0, , 0.6
    Oponent(0).Circle (Oponent_Width_Half, Oponent_Height_Half), Abs(Oponent_Width_Half - Icon_Dim_Comp), , 0.6
End Sub

Private Sub Player_Icon_Paint()
    BackWall.Refresh
    If GameType Then
        Form1.FillColor = &HFFFFFF
        Form1.Circle ((Player_Icon.Left + Icon_Width_Max_Half), (Field_Height - Icon_Height_Max - Old_Player_Icon_Elev_Delta)), (Icon_Width_Max_Half - Icon_Dim_Comp), &HFFFFFF, , 0.6
        Player_Icon.Cls
        Player_Icon.Move New_Player_Icon_Left, (Field_Height - Icon_Height_Max), Icon_Width_Max, Icon_Height_Max
        Player_Icon.Circle (Icon_Width_Max_Half, Icon_Height_Max_Half), (Icon_Width_Max_Half - Icon_Dim_Comp), , 0.6
        Form1.FillColor = &HFF00&
        Form1.Circle ((Player_Icon.Left + Icon_Width_Max_Half), (Field_Height - Icon_Height_Max - New_Player_Icon_Elev_Delta)), (Icon_Width_Max_Half - Icon_Dim_Comp), &H0&, , 0.6
    Else
        Form1.FillColor = &HFFFFFF
        Form1.Circle ((Player_Icon.Left + Old_Player_Icon_Width_Half), (Player_Icon.Top - Old_Player_Icon_Elev_Delta)), (Old_Player_Icon_Width_Half - Icon_Dim_Comp), &HFFFFFF, , 0.6
        If New_Player_Icon_Top >= GHStart Then
            Player_Icon.Cls
            Player_Icon.Move New_Player_Icon_Left, New_Player_Icon_Top, New_Player_Icon_Width, New_Player_Icon_Height
        End If
    End If
End Sub
Player_Icon.Circle (New_Player_Icon_Width_Half, New_Player_Icon_Height_Half, (New_Player_Icon_Width_Half - Icon_Dim_Comp), 
, , , 0.6
Form1.FillColor = &HFF00&
Form1.Circle ((Player_Icon.Left + New_Player_Icon_Width_Half), (Player_Icon.Top - New_Player_Icon_Elev_Delta), (New_Player_Icon_Width_Half - Icon_Dim_Comp), &H0&, , , 0.6
End If
End If

End Sub

Private Sub Player_Update_Timer()

Rem Adjust movement timer
Movement_Time = Movement_Time + 1

Old_Player_Icon_Width_Half = New_Player_Icon_Width_Half
Old_Player_Icon_Height = New_Player_Icon_Height
Old_Target_Width_Half = Target_Width_Half
Old_Target_Height = Target_Height
Old_Oponent_Width_Half = Opponent_Width_Half
Old_Oponent_Height = Opponent_Height
Old_Player_Icon_Elev_Delta = New_Player_Icon_Elev_Delta

Rem Execute if hardware tracking valid
If XYZ_Grab_State And ((Origin_Data_Packet.Track_Status And &H3) >= &H2) Then

Rem Y (height) coordinates
New_Player_Icon_Elev_Delta = (Origin_Data_Packet.Y.Coordinate - Player_icon_Y_Offset) / Field_Scale.ZDiv

Rem X(lateral), Z(depth) coordinates offset compensation and new player position calculation
New_Player_Icon_Top_Delta = (Origin_Data_Packet.Z.coordinate - Player_icon_Z_Offset) / Field_Scale.YDiv
New_Player_Icon_Left_Delta = (Origin_Data_Packet.X.Coordinate - Player_icon_X_Offset) / Field_Scale_XDiv
New_Player_Icon_Left = New_Player_Icon_Left_Delta + Player_Init_X
New_Player_Icon_Top = New_Player_Icon_Top_Delta + Player_Init_Y

If Not (GameType) Then
New_Player_Icon_Depth_Scale = New_Player_Icon_Top_Delta / (4 * Abs(Player_ICON.Top / GHScale))
New_Player_Icon_Width = Icon_Width_Max * (1 + ((-New_Player_Icon_Depth_Scale + New_Player_Icon_Top_Delta) * 1.5 / Field_Height))

If (New_Player_Icon_Width < (2 * Icon_Dim_Comp)) Then
    New_Player_Icon_Width = 2 * Icon_Dim_Comp
End If

New_Player_Icon_Height = (New_Player_Icon_Width * 3) / 4
New_Player_Icon_Width_Half = New_Player_Icon_Width / 2
New_Player_Icon_Height_Half = New_Player_Icon_Height / 2
New_Player_Icon_Lateral_Scale = Abs(Player_Icon_Top / Horizon)
New_Player_Icon_Elev_Delta = New_Player_Icon_Elev_Delta * (1# - (Horizon / Player_Icon_Top))
End If

If New_Player_Icon_Elev_Delta < -New_Player_Icon_Height_Half Then
    New_Player_Icon_Elev_Delta = -New_Player_Icon_Height_Half
End If

Rem Consider special case of first Target appearance
If Target_Moved Then

    Movement_Time = 1

Rem Save starting reference position
Old_Player_Icon_Left = New_Player_Icon_Left
Old_Player_Icon_Top = New_Player_Icon_Top

Rem Calculate delta and sign between new player position and current target position
Rem Delta_Target_X = Target.Left - Player_Icon.Left
Rem Delta_Target_Y = Target.Top - Player_Icon.Top
Rem Sgn_Delta_Target_X = Sgn(Delta_Target_X)
Rem Sgn_Delta_Target_Y = Sgn(Delta_Target_Y)
Rem If (Sgn_Delta_Target_X + Sgn_Delta_Target_Y) = -2 Then
    Rem Player2Target_Direction = "2"
Rem Elself (Sgn_Delta_Target_X + Sgn_Delta_Target_Y) = 2 Then
    Rem Player2Target_Direction = "4"
Rem Elself Sgn_Delta_Target_X >= 0 Then
    Rem Player2Target_Direction = "1"
Rem Else
    Rem Player2Target_Direction = "3"
Rem End If

Rem Calculate direction and delta arrays between player and target
Rem A_Player2Target_Direction(Moves) = Player2Target_Direction
Rem A_Delta_Target_X(Moves) = Delta_Target_X
Rem A_Delta_Target_Y(Moves) = Delta_Target_Y
Target_Moved = False
End If

Rem Calculate delta and sign between current player position and last player position
Delta_Player_New_X = New_Player ICON Left - Old_Player ICON Left
Sgn_Delta_Player_New_X = Sgn(Delta_Player_New_X)
Delta_Player_New_Y = New_Player ICON Top - Old_Player ICON Top
Sgn_Delta_Player_New_Y = Sgn(Delta_Player_New_Y)

Rem Save last player icon position
Old_Player ICON Left = New_Player ICON Left
Old_Player ICON Top = New_Player ICON Top

XYZ_Grab State = False
XYZ_Update State = True

End If

Rem Update grid when player icon has moved
If XYZ_Update State Then

GridColor = &H0&

Rem Enable grid scrolling if player icon in close proximity of starting position
If (New_Player ICON Top_Delta > 0) Or Not (GameType) Then
    GridStop = True
Else
    GridStop = False
End If

Rem Initialize horizontal grid scrolling parameters
Old_GHDiv = GHDiv
Old_GHStart = GHStart
Old_GWBStart = GWBStart
Old_GWTStart = GWTStart
If Not (GridStop) Then
    GHStart = GHStartInit - (New_Player ICON Top_Delta / (4 * Abs(Player ICON Top / GHStart)))
    GHDiv = GHStepInc - (New_Player ICON Top_Delta / (GHScale * Abs(Player ICON Top / (GHStart * 1.5)))))
End If
GHStep = 0
Old_GHStep = 0
GHCOUNT = 0

Rem Draw horizontal scrolling grid lines
Do Until Old_GHStep > Field_Height
    If GHCount > 1 Then
        GridY = Old_GHStart + Old_GHStep
        Form1.Line (0, GridY)-(Field_Width, GridY), &HFFFFFF
        If (GridStop And (GHCount > 0)) Then GridColor = &HFF&
        GridY = GHStart + GHStep
        Form1.Line (0, GridY)-(Field_Width, GridY), GridColor
    End If
    GHStep = GHStep + (GHDiv * (GHCount + 1) * (GHCount + 1))
    Old_GHStep = Old_GHStep + (Old_GHDiv * (GHCount + 1) * (GHCount + 1))
    GHCount = GHCount + 1
Loop

Rem Initialize vertical radial scrolling grid lines
If Not (GridStop) Then
    GWTStart = New_Player.Icon_Top_Delta * GWTScale
End If
GWBStart = -GWTStart - (New_Player.Icon_Left_Delta * GWBScale)

Rem Draw vertical radial scrolling grid lines
For GRadialNum = 0 To 15
    Old_GTRadial(GRadialNum) = GTRadial(GRadialNum)
    If Not (GridStop) Then
        GTRadial(GRadialNum) = GRadialNum * ((Field_Width - (2 * GWTStart)) / 15)
    End If
    Form1.Line ((Old_GWTStart + Old_GTRadial(GRadialNum)), (Old_GHStart + Old_GHDiv))-(Old_GWBS + GBRadial(GRadialNum)), Field_Height), &HFFFFFF
    If GridStop Then GridColor = &HFF&
    Form1.Line ((GWTStart + GTRadial(GRadialNum)), (GHStart + GHDiv))-((GWBStart + GBRadial(GRadialNum)), Field_Height), GridColor
Next GRadialNum

Rem Draw background rectangles to simulate horizon perspective change
If (Sgn_Delta_Player_New_Y >= 0) Then
    Form1.Line (0, (GHStart + GHDiv))-(Field_Width, (Old_GHStart + Old_GHDiv)), &HFFFFFF, BF
End If
If (Sgn_Delta_Player_New_Y <= 0) Then
    Form1.Line (0, (Old_GHStart + Old_GHDiv))-(Field_Width, (GHStart + GHDiv)), &HFFFF00, BF
End If
End If
End If

Rem Wait for player to return to starting Y position
If Target_Found And (New_Player_ICON_Top_Delta >= 0) Then
  Rem Erase target shadow
  Form1.FillColor = &HFFFFFF
  Form1.Circle ((Target.Left + Old_Target_Width_Half), (Target.Top +
  (Old_Target_Height / 1.5)), Abs(Old_Target_Width_Half - Icon_Dim_Comp),
  &HFFFFFF, , , 0.6
  Target.Enabled = True
  Target_Top = (GHStart + (4 * GHDiv)) + (Rnd * (Field_Height - GHStartInit - (2 *
  Icon_Height_Max)))
  Target_Top = Target_Top
  Target_Top_Delta = Target_Top - (Field_Height - Icon_Height_Max)
  Target_Left = (Rnd * (Field_Width - Target_Width)) + Abs(Player_ICON.Top /
  Target_Top)
  Target.Visible = True
  Target_Moved = True
  Target_Found = False
  Target_Init = True
  Rem Etime Enabled = True
  End If

Rem Modulate target position, size and redraw
If ((New_Player_ICON_Top_Delta <= 0) And GameType) Or Target_Init Then
  Target_Init = False
  Target_Density_Scale = New_Player_ICON_Top_Delta / (4 *
  Abs(Player_ICON.Top / Target_Top))
  Target_Width = Icon_Width_Max * (1 + ((-Target_Density_Scale +
  Target_Top_Delta) / (Field_Height - Horizon)))
  Target_Height = (Target_Width * 3) / 4
  Target_Width_Half = Target_Width / 2
  Target_Height_Half = Target_Height / 2
  End If

Target_Lateral_Scale = Abs(Player_ICON.Top / Target_Top)
If Target_Lateral_Scale < 1 Then Target_Lateral_Scale = 1

Rem Monitor player/target boundaries interception
Player_ICON_Intercept_X = Player_ICON.Left + New_Player_ICON_Width
Player_ICON_Intercept_Y = Player_ICON.Top + New_Player_ICON_Height
If (Player_ICON_Intercept_X >= Target.Left) And (Player_ICON.Left <= (Target.Left
  + Target_Width)) And ((Target_Top + Target_Height) >= Player_ICON.Top) And
  (Target_Top <= Player_ICON_Intercept_Y) Then
  If Not (Target_Found) Then
    Target_Found = True
    Target.Visible = False
    Target.Enabled = False
    Rem Transit_Time(Moves) = Movement_Time
    Moves = Moves + 1
  End If
Display_Update = True
Beep
End If
End If

Rem Redraw oponent icon
If Oponent_Visible.Value And XYZ_Update_State Then
    Rem Monitor player/oponent boudaries interception
    Rem Disable appropriate timers
    Rem If (Player_Icon_Intercept_X >= Oponent(0).Left) And (Player_Icon.Left <=
    (Oponent(0).Left + Oponent_Width)) And ((Oponent(0).Top + Oponent_Height) >=
    Player_Icon.Top) And (Oponent(0).Top <= Player_Icon_Intercept_Y) Then
        Rem Game = False
        Rem Player_Update.Enabled = False
        Rem Beep
        Rem Else

        Oponent_Y_Step = Oponent_Y_Delta / (Abs((Player_Icon.Top /
        Oponent(0).Top) - 1.1) + 0.01)
        Oponent_X_Step = Oponent_X_Delta

Select Case Oponent_Trajectory
    Case 0
        Oponent_X_Position = Oponent_X_Position + Oponent_X_Step
        Oponent_Y_Position = Oponent_Y_Position + Oponent_Y_Step
    Case 1
        Oponent_X_Position = Oponent_X_Position - Oponent_X_Step
        Oponent_Y_Position = Oponent_Y_Position + Oponent_Y_Step
End Select

If (Oponent(0).Top < (GHStart + GHDiv)) Then
    Rem Or (Not (Target_Found) And ((Oponent(0).Left + Oponent(0).Width) >=
    Target.Left) And (Oponent(0).Left <= (Target.Left + Target_Width)) And
    ((Oponent(0).Top + Oponent(0).Height) >= Target.Top) And (Oponent(0).Top <=
    (Target.Top + Target.Height))) Then
        Oponent(0).Visible = False
    Else
        Oponent(0).Visible = True
End If

If ((Oponent(0).Left + Oponent_Width) <= 0) Then
    Oponent_Trajectory = 0
    Oponent_Trajectory_Change = True
ElseIf (Oponent(0).Left >= Field_Width) Then
    Oponent_Trajectory = 1
    Oponent_Trajectory_Change = True
Elseif (Oponent(0).Top >= Field_Height) Then
    Oponent_Trajectory = Not (Oponent_Trajectory) And &H1
    Oponent_Trajectory_Change = True
End If

If Oponent_Trajectory_Change Then
    Oponent_Trajectory_Change = False
    Oponent_Depth_Scale = New_Player_Icon_Top_Delta / (4 * Abs(Player_Icon.Top / (GHStart + GHDiv)))
    Oponent_Lateral_Scale = Abs(Player_Icon.Top / (GHStart + GHDiv))
    Oponent_Y_Position = GHStart + (2 * GHDiv) + Oponent_Depth_Scale + Oponent_X_Position = Rnd * (Field_Width - Oponent_Width)
Else
    Oponent_Depth_Scale = New_Player_Icon_Top_Delta / (4 * Abs(Player_Icon.Top / Oponent(0).Top))
    Oponent_Lateral_Scale = Abs(Player_Icon.Top / Oponent(0).Top)
End If

Oponent_Top = Oponent_Y_Position
Oponent_Left = Oponent_X_Position
Oponent_Top_Delta = Oponent_Top - (Field_Height - Icon_Height_Max)
Oponent_Width = Icon_Width_Max * (1 + (Oponent_Depth_Scale + Oponent_Top_Delta) / (Field_Height - Horizon)))
Oponent_Height = (Oponent_Width * 3) / 4
Oponent_Width_Half = Oponent_Width / 2
Oponent_Height_Half = Oponent_Height / 2

Oponent_Paint(0)

End If

Rem Update target position
Target_Paint

Rem Update player position
Player_Icon_Paint

XYZ_Update_State = False

Rem Adjust real time clock
Rem If Etme_Enabled Then
    Rem Etme_sec = Etme_sec + 1
    Rem If Etme_sec >= Einterval Then
        Rem Etme_sec = 0
        Rem esecond = esecond + 1
        Rem If esecond = 60 Then
            Rem esecond = 0

48
    Rem eminute = eminute + 1
    Rem End If
    Rem Update_Etime = True
    Rem End If
    Rem End If

End Sub

Private Sub Quit_Click()
    End
End Sub

Private Sub Start_Click()

    Dim j As Long

    Target_Delay_MSecond = 2000
    Target_Delay_Value = Target_Delay_MSecond / Player_Update.Interval

    Target_Color_Step = &H100 & / Target_Delay_Value
    Target_Color_Step = (Target_Color_Step * &H10000) + ((Target_Color_Step) * 
    &H100 &)

    Graph1.Visible = False
    GameType = Option1(1).Value
    If GameType Then
        Field_Scale_XDiv = 3#
        Field_Scale_YDiv = 3#
        Field_Scale_ZDiv = 3#
    Else
        Field_Scale_XDiv = 10#
        Field_Scale_YDiv = 10#
        Field_Scale_ZDiv = 4#
    End If

    Rem Initialize elapsed time
    Rem essecond = 0
    Rem eminute = 0
    Rem Etime_sec = 0
    Rem Einterval = 1000 \ Player_Update.Interval
    Rem Etime_Disp.Caption = Format$(eminute, "#") & ":" & Format$(essecond, "00")
    Rem Etime_Enable = False

    Rem Clear controls' display
    Oponent_Visible.Visible = False
    Start.Visible = False
Quit.Visible = False
Option1(0).Enabled = False
Option1(1).Enabled = False
Option1(0).Visible = False
Option1(1).Visible = False
Oponent_Visible.Enabled = False
Start.ENABLED = False
Quit.ENABLED = False
Label1.ENABLED = False
Label1.VisiBLE = False

Rem Initialize size and position of player icon
Rem Determine default sizes of other icons at the same coordinates

Player Icon.Visible = True
Icon_Width_Max = 1000 'twips
Icon_Width_Max_Half = Icon_Width_Max / 2
Icon_Height_Max = (Icon_Width_Max * 3) / 4
Icon_Height_Max_Half = Icon_Height_Max / 2
New_Player_Icon_Top_Delta = 0
Player_Init_X = Field_Width_Center - Icon_Width_Max_Half
Player_Init_Y = Field_Height - Icon_Height_Max
Player_Icon.Top = Player_Init_Y
Player_Icon.Left = Player_Init_X
New_Player_Icon_Left = Player_Init_X
New_Player_Icon_Top = Player_Init_Y
New_Player_Icon_Elev_Delta = 0
Old_Player_Icon_Elev_Delta = 0
New_Player_Icon_Depth_Scale = New_Player_Icon_Top_Delta / (4 * Abs(Player_Icon.Top / Horizon))
New_Player_Icon_Width = Icon_Width_Max * (1 + ((New_Player_Icon_Depth_Scale + New_Player_Icon_Top_Delta) / (Field_Height - Horizon)))
New_Player_Icon_Height = (New_Player_Icon_Width * 3) / 4
Old_Player_Icon_Height = New_Player_Icon_Height
New_Player_Icon_Width_Half = New_Player_Icon_Width / 2
Old_Player_Icon_Width_Half = New_Player_Icon_Width_Half
New_Player_Icon_Height_Half = New_Player_Icon_Height / 2
New_Player_Icon_Lateral_Scale = Abs(Player_Icon.Top / Horizon)
Rem Player_Icon.Refresh

Rem Initialize grid parameters
GHStartInit = Horizon
GHStart = GHStartInit
GRAdialNum = 15
GHSteplnc = 25
GHDiv = GHSteplnc
GWStart = 32000
GWBWidth = 32000
GHScale = 64
GWTScale = 2
If GameType Then
  GWBScale = 2
Else
  GWBScale = 10
End If
GWTStart = 0
GWBSStart = 0

Rem Create initial grid radial endpoints
For j = 0 To GRRadialNum
  GBRadial(j) = j * ((Field_Width + (2 * GWBWidth)) / GRRadialNum)
  GTRadial(j) = j * (Field_Width / GRRadialNum)
Next j

Rem Create initial background wall dimensions
BackWall.Left = 0
BackWall.Width = Field_Width
BackWall.Top = 0
BackWall.Height = GHStart + GHDiv
BackWall.Visible = True

Rem Pause to allow player to arrive at initial position
For j = 0 To &H8FFFFF
  Next j

Rem Initialize flags and serial communications
Initial_Offset = True
XYZ_Grab_State = False
XYZ_Update_State = False
XYZ_Grab.RThreshold = 1
XYZ_Grab.InBufferCount = 0
Game = True
Moves = 0
Oponent_Trajectory = 0
Oponent_Trajectory_Change = True
Oponent_Y_Position = GHStartInit + GHDiv
Oponent_X_Position = Rnd * (Field_Width - Oponent_Width)
Oponent_Top = Oponent_Y_Position
Oponent_Left = Oponent_X_Position
Oponent_Depth_Scale = New_Player Icon_Top_Delta / (4 * Abs(Player Icon.Top / Oponent(0).Top))
Oponent_Top_Delta = Oponent_Top - (Field_Height - Icon_Height_Max)
Oponent_Width = Icon_Width_Max * (1 + ((Oponent_Depth_Scale + Oponent_Top_Delta) / (Field_Height - Horizon)))
Oponent_Height = (Oponent_Width * 3) / 4
Oponent_Width_Half = Oponent_Width / 2
Oponent_Height_Half = Oponent_Height / 2
Oponent_Lateral_Scale = Abs(Oponent(0).Top / Target.Top)
Rem Oponent(0).Refresh

Target_Found = True
Target_Moved = False
Player_Update. Enabled = True

Rem Determine opponent visibility
If Oponent_Visible.Value Then
   Oponent_Move_Enabled = True
   Oponent(0). Enabled = True
   Oponent(0). Visible = True
Else
   Oponent_Move_Enabled = False
   Oponent(0). Visible = False
   Oponent(0). Enabled = False
End If

End Sub

Private Sub Target_Delay_Click()

End Sub

Private Sub Target_Paint()
If GameType Then
   Form1.FillColor = &H00FF00
   Form1.Circle ((Target.Left + Old_Target_Width_Half), (Target.Top + (Old_Target_Height / 1.5)), Abs(Old_Target_Width_Half - Icon_Dim_Comp), &H00FF00, , 0.6
   Target.Cls
   Target.Move (Target_Left - (New_Player Icon_Left_Delta /
Target_Lateral_Scale), (Target_Top - Target_Depth_Scale), Target_Width, Target_Height
Form1.FillColor = &H0F0F0
Form1.Circle ((Target.Left + Target_Width_Half), (Target.Top + (Target_Height / 1.5)), (Target_Width_Half - Icon_Dim.Comp), &H0F0F0, , 0.6
Target.Circle (Target_Width_Half, Target_Height_Half), (Target_Width_Half - Icon_Dim.Comp), , , 0.6
Else
Form1.FillColor = &HFFFFFF
Form1.Circle ((Target.Left + Old_Target_Width_Half), (Target.Top + (Old_Target_Height / 1.5)), Abs(Old_Target_Width_Half - Icon_Dim.Comp), &HFFFFFFF, , , 0.6
Target.Cls
Target.Move (Target.Left - (New_Player_Icon_Left_Delta / Target_Lateral_Scale)), Target_Top, Target_Width, Target_Height
Form1.FillColor = &H0F0F0
Form1.Circle ((Target.Left + Target_Width_Half), (Target.Top + (Target_Height / 1.5)), (Target_Width_Half - Icon_Dim.Comp), &H0F0F0, , , 0.6
Target.Circle (Target_Width_Half, Target_Height_Half), (Target_Width_Half - Icon_Dim.Comp), , , 0.6
End If
End Sub

Private Sub Timer1_Timer()
If Game Then
Rem If Update_Etime Then
Rem Etime Disp.Caption = Format$(eminate, "#") & ":" & Format$(esecond, "00")
Rem Update_Etime = False
Rem End If
If ((Origin_Data_Packet.Track_Status And &H3) < &H3) Then
XYZ_Grab.RThreshold = 1
End If
End If
End Sub

Private Sub XYZ_Grab_OnComm()
Dim dummy1 As Long
Dim dummy2 As Long

Dim Scale_Shift As Long
If (XYZ_Grab.CommEvent = 2) And (XYZ_Grab.InBufferCount >= 16) Then
Origin_Data_Packet.XYZ_Scaling = Asc(XYZ_Grab.Input)
If (Origin_Data_Packet.XYZ_Scaling And &HF0) = &H80 Then
Select Case (Origin_Data_Packet.XYZ_Scaling And &H3)
Case 0
Scale_Shift = 1
End If
End If
Case 1
    Scale_Shift = 2
Case 2
    Scale_Shift = 4
Case 3
    Scale_Shift = 8
End Select
Origin_Data_Packet.Track_Status = Asc(XYZ_Grab.Input)
If (Origin_Data_Packet.Track_Status And &HF0) = &H80 Then
    dummy1 = Asc(XYZ_Grab.Input) * &H100&
    dummy2 = Asc(XYZ_Grab.Input)
    Origin_Data_Packet.X_Coordinate = (dummy1 + dummy2) * Scale_Shift
    If ((Origin_Data_Packet.X_Coordinate And &H10000) = &H10000) Then
        Origin_Data_Packet.X_Coordinate = Origin_Data_Packet.X_Coordinate
    Or &HFFFFFF0000
    End If
    dummy1 = Asc(XYZ_Grab.Input) * &H100&
    dummy2 = Asc(XYZ_Grab.Input)
    Origin_Data_Packet.Y_Coordinate = (dummy1 + dummy2) * Scale_Shift
    If ((Origin_Data_Packet.Y_Coordinate And &H10000) = &H10000) Then
    Or &HFFFFFF0000
    End If
    dummy1 = Asc(XYZ_Grab.Input) * &H100&
    dummy2 = Asc(XYZ_Grab.Input)
    Origin_Data_Packet.Z_coordinate = (dummy1 + dummy2) * Scale_Shift
    Rem If ((Origin_Data_Packet.Z_coordinate And &H8000&) = &H8000&)
Then
    Rem Origin_Data_Packet.Z_coordinate =
    Origin_Data_Packet.Z_coordinate Or &HFFFFFF0000
End If
End If
XYZ_Grab_State = True
If Initial_Offset Then
    Player_Lcon_X_Offset = Origin_Data_Packet.X_Coordinate
    Player_Lcon_Y_Offset = Origin_Data_Packet.Y_Coordinate
    Player_Lcon_Z_Offset = Origin_Data_Packet.Z_coordinate
    Initial_Offset = False
End If
XYZ_Grab.InBufferCount = 0
XYZ_Grab.RThreshold = 16
End If
End If
End Sub
While the invention has been described in connection with certain embodiments, it is to be understood that the invention is not to be limited to the disclosed embodiments, but on the contrary, is intended to cover various modifications and equivalent arrangements included within the spirit and scope of the foregoing specification and of the following claims.
CLAIMS

We claim:

1. A testing and training system for assessing the ability of a player to complete a task, comprising:
   - providing a defined physical space within which the player moves to undertake the task;
   - tracking means for determining the position of the player within said defined physical space based on at least two Cartesian coordinates;
   - display means operatively coupled to said tracking means for displaying in a virtual space a player icon representing the instantaneous position of the player therein in scaled translation to the position of the player in said defined physical space;
   - means operatively coupled to said display means for depicting in said virtual space a protagonist;
   - means for defining an interactive task between the position of the player and the position of the protagonist icon in said virtual space;
   - means for assessing the ability of the player in completing said task based on quantities of distance and time.

2. The testing and training system as recited in Claim 1 wherein said task is interception of said protagonist by said player icon at a common position in said virtual space.
3. The testing and training system as recited in Claim 1 wherein said task is evasion of said protagonist by said player icon avoiding a common position with said protagonist in said virtual space.

4. The testing and training system as recited in Claim 1 wherein calculating means determines information relating to distance traveled by said player in said defined physical space and the elapsed time for said player to complete said task and providing said information on said display means.

5. The testing and training system as recited in Claim 1 wherein said task comprises a plurality of segments requiring sufficient movement of said player in said defined physical space to provide quantification of bilateral vector performance of said player in completing said task.

6. A system as in Claim 1 further comprising:
   measuring in essentially real time Y-plane excursion displacements of the user's center of gravity as the user responds to interactive protocols;
   calculating the user's movement velocities and/or accelerations during performance of said protocols;
   determining a user's most efficient dynamic posture; and
   providing numerical and graphical results of said measuring, calculating, and determining.
7. A system as in Claim 1, further comprising:
   calibrating the system for a dynamic posture that a user wishes to train; selected by the user;
   providing varying interactive movement challenges over distances and directions;
   providing real-time feedback of a measurement of compliance with the desired dynamic posture during performance of the protocols, and
   providing results of the user’s performance.

8. A system as set forth in Claim 1, further comprising:
   providing a wireless heart-rate monitor for the user to wear, said monitor coupled to said computer;
   providing means for a user to enter desired target heart-rate range;
   providing interactive, movements over varying distances and directions;
   providing instructions to a user comparing in real time the actual versus the desired heart-rate to determine compliance with a selected heart-rate zone during a user’s performance;
   monitoring in real time physical activity and heart rate so that a physical activity to heart rate ratio can be ascertained; and
   presenting results of a user’s performance.

9. A system as in Claim 1 further comprising:
tracking at sufficient sampling rate the user's movement in three-degrees-of-freedom during his performance of protocols, including unplanned movements over various vector distances;
calculating in essentially real-time the user's movement accelerations and decelerations;
categorizing each movement leg to a particular vector; and
displaying feedback of bilateral performance.

10. A system of claims 1-9 further comprising:
providing a means for a user to input data unique to such user relating to energy expenditure and
calculating a user's energy expenditure during physical activity.

11. A testing and training system comprising:
means for tracking a user's position within a physical space in three dimensions;
display means operatively linked to said tracking means for indicating the user's position within said physical space in essentially real time;
means for defining a physical activity for said user operatively connected to said display means; and
means for assessing the user's performance in executing said physical activity.

12. A system as in Claim 11 further comprising:
measuring in essentially real time Y-plane excursion displacements of the user's center of gravity as the user responds to interactive protocols;
calculating the user's movement velocities and/or accelerations during
performance of said protocols;

determining a user's most efficient dynamic posture; and

providing numerical and graphical results of said measuring, calculating,
and determining.

13. A system as in Claim 11, further comprising:
calibrating the system for a dynamic posture that a user wishes to
train; selected by the user;

providing varying interactive movement challenges over distances
and directions;

providing real-time feedback of a measurement of compliance with
the desired dynamic posture during performance of the protocols, and

providing results of the user's performance.

14. A system as set forth in Claim 11, further comprising:
providing a wireless heart-rate monitor for the user to wear, said
monitor coupled to said computer;

providing means for a user to enter desired target heart-rate range;

providing interactive, movements over varying distances and
directions;

providing instructions to a user comparing in real time the actual
versus the desired heart-rate to determine compliance with a selected heart-rate zone
during a user's performance;
monitoring in real time physical activity and heart rate so that a
physical activity to heart rate ratio can be ascertained; and

presenting results of a user’s performance.

15. A system as in Claim 11 further comprising:

tracking at sufficient sampling rate the user’s movement in three-
degrees-of-freedom during his performance of protocols, including unplanned
movements over various vector distances;

calculating in essentially real-time the user’s movement
accelerations and decelerations;

categorizing each movement leg to a particular vector; and
displaying feedback of bilateral performance.

16. A system as in Claims 11-15 further comprising:

providing a means for a user to input data unique to such user
relating to energy expenditure; and

calculating a user’s energy expenditure during physical activity.

17. A testing and training system comprising:

a tracking system for providing a set of three dimensional
coordinates of a user within a physical space;

a computer operatively linked to said tracking system to receive
said coordinates from said tracking system and indicate the user’s position within said
physical space on a display monitor in essentially real time;

wherein said computer includes a program to define a physical
activity for the user and measure the user’s performance in executing the activity.
18. A system as in claim 17, further comprising:

measuring in essentially real time Y-plane excursion displacements of the user's center of gravity as the user responds to interactive protocols;
calculating the user's movement velocities and/or accelerations during performance of said protocols;
determining a user's most efficient dynamic posture; and providing numerical and graphical results of said measuring, calculating, and determining.

19. A system as in Claim 17, further comprising:
calibrating the system for a dynamic posture that a user wishes to train; selected by the user;
providing varying interactive movement challenges over distances and directions;
providing real-time feedback of a measurement of compliance with the desired dynamic posture during performance of the protocols, and providing results of the user's performance.

20. A system as set forth in Claim 17, further comprising:

providing a wireless heart-rate monitor for the user to wear, said monitor coupled to said computer;
providing means for a user to enter desired target heart-rate range;
providing interactive, movements over varying distances and directions;
providing instructions to a user comparing in real time the actual versus the desired heart-rate to determine compliance with a selected heart-rate zone during a user’s performance;

monitoring in real time physical activity and heart rate so that a physical activity to heart rate ratio can be ascertained; and

presenting results of a user’s performance.

21. A system as in Claim 17 further comprising:

tracking at sufficient sampling rate the user’s movement in three-degrees-of-freedom during his performance of protocols, including unplanned movements over various vector distances;

calculating in essentially real-time the user’s movement accelerations and decelerations;

categorizing each movement leg to a particular vector; and

displaying feedback of bilateral performance.

22. A system in Claims 17-21 further comprising:

providing a means for a user to pinput data unique to such user relating to energy expenditure; and

calculating a user’s energy expenditure during physical activity.

23. A method of testing a user’s physical abilities, said method comprising:

providing a physical space;

defining a physical activity for said user;
continuously tracking the three dimensional position of the user
within the physical space as said user executes the defined physical activity;

displaying a representation of the position of the user within the
physical space in real time; and

assessing a user's performance in executing the physical activity.

24. The system of claims 11-15 wherein said display means comprises:
means for displaying a virtual space proportional in dimensions to
said physical space; and

means for displaying a user icon in said virtual space at a location
which is a spatially correct representation of the user's position within said physical
space.

25. The system of claim 11-15 wherein said display means further
comprises means for displaying at least one protagonist icon in said virtual space.

26. The system of claim 25 wherein said at least one protagonist icon
comprises at least one obstacle icon.

27. The system of claim 26 wherein said physical activity defining
means comprises:

means for defining a set of behavioral characteristics for said at
least one protagonist icon; and

means for moving said at least one protagonist icon in accordance
with said behavioral characteristics.
28. The system of claim 27 wherein said behavioral characteristics comprise a number of protagonist icons, a protagonist icon speed and a protagonist icon intelligence level.

29. The system of claims 11-15 wherein said physical activity defining means comprises means for selecting between an intercept and evade objective.

30. The system of claim 11 wherein said performance assessing means comprises:
   means for measuring a distance traveled by the user; and
   means for measuring an elapsed time for the user to travel said distance.

31. The system of claim 30 wherein said performance assessing means further comprises means for calculating velocity of the user.

32. The system of claim 30 wherein said performance assessing means further comprises means for calculating acceleration of the user.

33. The system of claims 11-15 wherein said performance assessing means comprises:
   means for recording user position data;
   means for fitting said user position data by spline curves.

34. The system of claims 11-15 wherein said performance assessing means comprises means for calculating work experienced by the user.

35. The system of claims 11-15 wherein said performance assessing means comprises means for calculating the energy expended by the user.
36. The system of claims 11-15 wherein said performance assessing means comprises means for measuring a user’s dynamic posture while performing said physical activity.

37. The system of claims 11-15 further comprising a heart rate monitor operatively linked to said performance assessing means.

38. The system of claims 11-15 further comprising:

   providing a means for a user to input data unique to such user relating to energy expenditure and calculating a user’s energy expenditure during physical activity.

39. The system of claim 31 wherein said heart-rate monitor is wireless.

40. The system of claims 11-15 wherein said performance assessing means comprises means for producing bi-lateral comparisons of acceleration.
Record player icon's change in position

Objective(s) met?

Compute & display final score

END

FIGURE 7
Figure 9

Change direction of protagonist(s)

Record player icon's change in position

Objectives met?

Yes

Compute & display final score & calories expended

End

No

Obstacle contact?
INTERNATIONAL SEARCH REPORT

A. CLASSIFICATION OF SUBJECT MATTER
IPC(6): G01L 3/24
US CL: 73/379.01
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED
Minimum documentation searched (classification system followed by classification symbols)
U.S.: 73/379.01; 463/4, 30-34, 258; 364/410

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
Examiner Personal Art Digests: Virtual Reality

Electronic database consulted during the international search (name of database and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

<table>
<thead>
<tr>
<th>Category*</th>
<th>Citation of document, with indication, where appropriate, of the relevant passages</th>
<th>Relevant to claim No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y</td>
<td>US 4,751,642 A (SILVA et al.) 14 June 1988, Figure 2, Col. 5, lines 43-68, &amp; Col. 6, lines 1-19.</td>
<td>1-4, 6, 7, 10, 11, 14, 16, 17, 20, 22, 23, 25-40</td>
</tr>
<tr>
<td>A</td>
<td>US 5,320,538 A (BAUM) 14 June 1994.</td>
<td>1-40</td>
</tr>
</tbody>
</table>

X Further documents are listed in the continuation of Box C. See patent family annex.

* Special categories of cited documents:
  *"A": document defining the general state of the art which is not considered to be of particular relevance
  *"E": earlier document published on or after the international filing date
  *"L": document which may throw doubt on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
  *"O": document referring to an oral disclosure, use, exhibition or other means
  *"P": document published prior to the international filing date but later than the priority date claimed
  *"T": later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
  *"X": document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
  *"Y": document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
  *"&": document member of the same patent family

Date of the actual completion of the international search: 31 MARCH 1997
Date of mailing of the international search report: 23 APR 1997

Name and mailing address of the ISA/US Commissioner of Patents and Trademarks
Box PCT
Washington, D.C. 20231
Facsimile No. (703) 305-3230

Authorized officer: Elizabeth L. Dougherty
Telephone No. (703) 305-8578

Form PCT/ISA/210 (second sheet)(July 1992)*
<table>
<thead>
<tr>
<th>Category</th>
<th>Citation of document, with indication, where appropriate, of the relevant passages</th>
<th>Relevant to claim No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y, P</td>
<td>US 5,524,637 A (ERICKSON) 11 June 1996, Figure 1, Col. 7, lines 34-67, Col. 2, lines 1-67, Col. 9, lines 1-67, Col. 10, lines 1-67, &amp; Col. 11, lines 1-18.</td>
<td>8-11, 14, 16, 20, 22, 23, 35, 37-39</td>
</tr>
</tbody>
</table>