EG UK Theory and Practice of Computer Graphics (2011)
Hamish Carr, Ian Grimstead (Editors)

Visualisation of blockplay in early childhood

A. Albin-Clark[1], T.L.J. Howard[1] and B. Anderson[2]

[1]School of Computer Science, The University of Manchester, UK
[2]Faculty of Education, Liverpool Hope University, UK

Abstract

Observation of young children is common in educational settings. For trainee practitioners however, access to children is infrequent, and even then, only small amounts of observable activity can be captured. We have developed a cross-platform software application to support observations made by pedagogical practitioners. Our system combines game-oriented technology: 3D graphics, animation, physics and classical game artificial intelligence. The stages of blockplay are used to represent the physical play of characters, and social play stages and egocentrism define their behaviour. We present the graphical side of the system including visualisations options that expose the underlying functionality. The prototype system – Observation – which we believe is the first to employ 3D interactive computer graphics for visualising early childhood play, is available for download and evaluation.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Virtual reality I.6.8 [Simulation and Modeling]: Types of Simulation—Animation

1. Introduction

Young children are observed so that activities can be planned and assessed according to their development and needs. Several schedules of observable behaviour exist, including: Practice Guidance for the Early Years Foundation Stage [Dep08]; covering the six areas of learning and development (in the United Kingdom), which places emphasis upon a stage-related development continuum; and the process-oriented child monitoring system [LVKD02], which identifies stage-independent core behaviour. Observation schedules used for training may be accompanied by textual case studies, perhaps with supporting images and/or video. It is difficult for an observer to be a distant onlooker because children expect adults present to supervise and help them. Consequently, the needs of children may not be met, and there may be health and safety risks. Making notes, recording a video, or taking photographs, can also have an impact on the way children play and behave. Video recording is also problematic. Fixed cameras can only capture activity within one area and may need editing to remove periods of inactivity bur an edited recording is not a true representation of events. Hand-held cameras are more flexible but are intrusive. An observer may miss something interesting. It is also quite hard to walk and film at the same time [BM06]. Given the benefits and limitations of traditional observation, can the observation of children's play be simulated? We decided to explore this and chose the stages of blockplay as the physical observable activity.

Blocks are usually manufactured from hardwood and can be put together in neat, geometric patterns. A unit block from Community Playthings [Com10] has these dimensions: width 140mm; height 70mm; and depth 35mm. Half, double and quadruple unit blocks are half, twice and four times the width of a unit block. Blocks are considered important because they are unstructured toys requiring children to be imaginative. Blocks allow the exploration of many aspects of learning including: science; mathematics; physical development; social studies; social-emotional; art; and language [Hir84]. Blockplay can be considered a symbol system and powerful nonverbal language [GF92]. Early childhood research shows that children pass through several stages in blockplay, with older children doing this faster [WK00]. The stages are: Carrying; Stacking; Bridging; Enclosures; Patterns and symmetry; Early representational; and Later representational.

To give support to early childhood practitioners we have developed an interactive, cross-platform application that simulates the play of young children, drawing upon the research from existing models of behaviour, providing a sandbox for experimentation with humanoid characters which

DOI: 10.2312/LocalChapterEvents/TPCG/TPCG11/053-060

aims: to explain; to discover new questions; to promote a scientific habit of mind; to bound outcomes to plausible ranges; and to train [Eps08]. We have used C++ and OpenGL, together with tools and resources [Isr11] that are freely available and/or open source, with permissive licensing.

The organisation of the rest of this paper is as follows: Section 2 presents related work; Section 3 looks at various aspects of the characters; Section 4 describes the environment; Section 5 explores viewing; Section 6 discusses the definition of objects and physics; Section 7 describes how users can customise the world; Section 8 briefly gives the findings of a small-scale survey and focus groups with early childhood lecturers and students at a collaborating university; and Section 9 gives our conclusion.

2. Related work

There has been limited work that has attempted to simulate the educational environments of young children.

Falender, Jordan-Marsh and Murar [FJMM79] describe a purely paper-based role-playing simulation for administrators, in which participants adopt roles within the early childhood decision-making process, making use of in-box materials.

Ferry, Kervin, Cambourne, Turbill, Puglisi, Jonassen and Hedberg [FKC*04] present a largely text-based web application, aimed at pre-service primary school teachers, for developing decision-making using scenarios. A knowledge-base of expert pedagogical advice can be used for comparing their teaching and classroom management decisions.

The proprietary simSchool [Cur10] is a web-based application simulating the interactions between a teacher and learners. The teacher makes academic or behavioural assertions and observations, or poses questions. When configuring a simulation, learners and tasks can be random or if they are created, learners or tasks can be defined using identical dimensions. Various graphs enable the teacher to track the performance of the learner over a time period, and determine whether it was as expected, better, or worse than expected. The teacher can decide whether the tasks they assigned to learners were suitably based upon their individual abilities and personalities.

Ferry et al's system and simSchool are centred around adult-initiated activities, which are typical with children after pre-school. simSchool has 2D graphics to show learners in several states between boredom and engaged, but these systems do not show any activities occurring; it is left to the imagination of the user.

3. Characters

3.1. Appearance

In many video games, player-characters can be customised, sometimes to match the appearance of the player [Coo07]. In a similar way, non-player-characters (NPCs) can be customised, though this is not as common, and not usually at the individual level. In both these cases, there may be options for the modification of: gender; facial features; hair; body size and proportions; clothing; and ethnicity. By not giving these options, stereotyping [Isb06] because of appearance is totally eliminated and there is no place for cultural or gender bias in observations. The abstraction of physical appearance therefore removes any cultural labelling, but it also simplifies implementation, since generating variety in a meaningful way amongst individuals is not an easy task [ACH09]. For example, what does a typical four-year-old girl look like? Of course this cannot be answered, unless through the medium of stereotypes.

Abstract characters are used in early childhood pedagogy, where featureless/expressionless dolls in Steiner Waldorf kindergarten are provided so that the young child places their individual interpretation onto a universal figure [Nic07]. The same approach is being used by the Observation platform, by viewing a character as an active embodiment of behaviour and well-being [MP02], a physical manifestation of engagement in play, rather than focussing on other features of the character.

Children do have varying body proportions at different ages, so should this be represented at all? Our approach to this was to globally scale the skeleton so that the height matched the average for a child of a given age. For example, the height of a three to four-year-old is approximately one metre. Consequently, all children of that age will be given identical body proportions.

Figure 1: *Abstract appearance using colour and number for identification.*

Separate meshes are used to represent the parts of the body, including a featureless head. The use of separate, rigid body parts means that these could be swapped for alternatives. The biped model used was taken initially from a Blender rig [Ces08], and was adapted to suit our needs us-

ing Blender [Ble10]. The meshes are loaded using the GLM library [Rob00].

For even more levels of abstract representation, there are options for displaying characters which include one or more of the following: bones; joints; and rigid body parts.

When there are many characters in a scene, what can be used for identification? In video games, player-characters and even non-player characters may be assigned names, but identifiers are usually culturally-bound, and very often gender-specific. We use a more abstract representation giving random colours (a combination of Red, Green and Blue components) to every individual, along with a numerical label, illustrated in Figure 1. If the colour of one character is close to another character, this could be interpreted as similarity in some unspecified dimension.

The chest area can be used to show the character's well-being. The colour fades from black to white as the level of dissatisfaction increases, triggered by events such as not being able to find certain objects with which to build. This visualisation of well-being is a continuous representation of the discrete Leuven well-being scale [LVKD02], which uses a five-point scale: Extremely high; High; Moderate; Low; and Extremely low. As the well-being changes, we adjust the Red, Green and Blue components of the chest geometry's colour in equal quantities to give the many shades of grey between black and white. Internally, well-being is represented as a value between 0 and 1. At intervals of 0.2, it is mapped to the Leuven discrete stages and reported as text through the event log and character information pane.

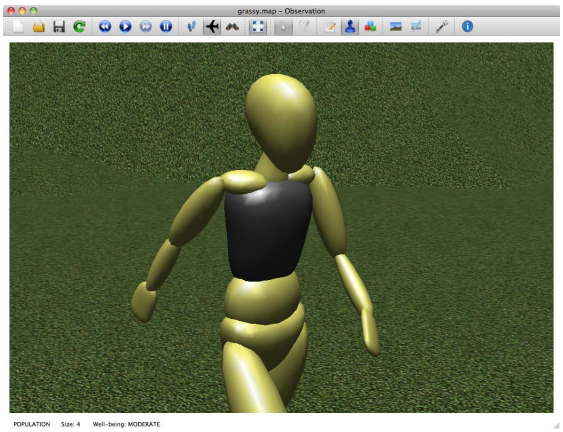

Figure 2: *Well-being vest showing a continuous representation of the discrete, five-point Leuven well-being scale.*

3.2. Movement

Characters can pathfind to obtain the shortest route through the world and around obstacles using our implementation of the A* search algorithm [BS04]. A finite state machine [AN07] is used for pathfinding and following the waypoints generated by the pathfinding. We do not want characters to avoid going over hilly terrain so the A* search is still conducted in 2D space, although the A* nodes (waypoints) are rendered following the contours of the terrain. Figure 3 shows colour-coded paths visualised as a series of connected lines representing the edges between nodes (above), and also as a series of cubes representing the nodes (below). The current target waypoint is shown as a colour-coded cube just in front of the character.

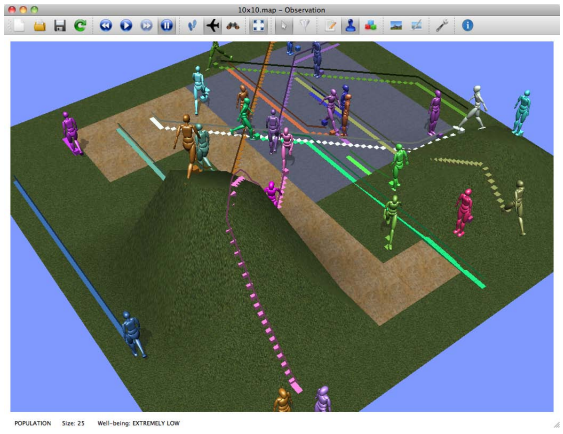

Figure 3: *A* paths through the scene.*

Local steering is used to help characters avoid each other, using three feelers (as illustrated in Figure 4, near the head) to test for intersections with other characters, using spherical containment to calculate steering forces, in conjunction with a movement priority system.

Rather than giving characters global knowledge of the objects within the environment, characters are given perception for a more individual representation. A uniform grid is used as the spatial indexing method. Objects with their centre in a cell of the grid are registered with the data structure for that cell. A ray cast from the character, simulates vision for objects in the environment; each frame, the length and angle is adjusted, giving a sweeping motion rather like a searchlight beam, scanning grid cells within a certain range.

An occupancy grid is maintained for the purpose of path planning. Since there are many objects of various dimensions, at any orientation, and particularly because some blocks are long and thin, it is not possible to quantise each object to fit one cell neatly. A grid with a higher resolution is used for the purpose of recording occupancy. We use Bresenham's line algorithm to calculate lines between vertices that form the edges of bounding boxes and use the squares that form the Bresenham line to record an occupied piece of the world. A* search can now take place using the occupancy grid. However, in order to pick up or put down an object, we

do not want the character to be in the exact location where the object is, or is going to be, so we calculate reach points around the object's position, using points of the compass, a distance away from the object within a character's reach, where the terrain is free from obstacles, then proceed to find a path to the closest reach point.

The location of an object and planning a path to obtain it is illustrated in Figure 4. The goal object is shown above the character in red, signifying that it has not been obtained yet. The white line points to the uniform grid cell that is being queried. The green line points to the goal object which has been found in the world. The blue line points to the goal position, which is a calculated reach point. The red squares (actually cubes) are the occupancy grid. The small colour-coded cubes around the goal object are the reach points. The A* path is shown around the obstacles.

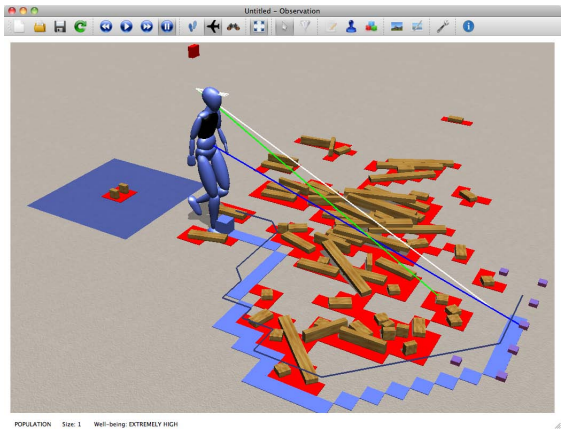

Figure 4: *Locating an object within the field of vision and planning a route to obtain it.*

QAvimator [Ree10] was used to create and edit our skeletal and animation data which conforms to the Biovision Hierarchy (BVH) motion capture standard, except the walk animation which is from Second Life [Lin10]. QAvimator was originally developed for the users of Second Life and it supports only one unisex skeleton. The simplified skeleton does not have any digits on the hands, which is a minor disadvantage.

3.3. Activity

On adding a new character to the world, a 1 square metre activity patch is allocated at that location, as long as the patch is: flat, where all four terrain vertices have the same height value (so constructions do not collapse under the control of the physics simulation); and unoccupied by any geometry. If the patch location is not optimal for the social play stage of the character, a relocation algorithm will attempt to find a suitable location, while satisfying the former criteria. The number of characters in the world is therefore determined by the number of patches. Each patch is colour-coded by character, as illustrated in Figure 5, and its display can be toggled on/off. The colour is applied to the patch geometry as it is textured. Unless a character is egocentric, they do not interfere with the patch contents of other characters.

To simulate construction during blockplay, after characters have been assigned a blockplay stage, they make a selection from construction plans on file, representative of that stage. These are called scripted construction plans because the position and orientation of all the elements in the construction have been specified in advance. A finite state machine controls the character's behaviour during scripted construction. Figure 5 illustrates several characters, at various stages, engaged in blockplay.

Characters are constrained in their building by the blocks available in the scene; they cannot 'spawn' objects. However, the observer may choose to place additional blocks into the scene.

Figure 5: *Characters engaged in blockplay at their colour-coded patch.*

3.4. Selection

Characters can be selected within the scene to display dynamic data in the character information pane such as: their well-being; their action; and the physical object of interest. Static data such as: their play stage; their blockplay stage; and whether they are egocentric, can also be displayed. If full screen graphics are not in use, a timestamped event log can show both archived and real-time events, filtered on characters, if needed. The simulation can be paused to make monitoring the data easier, as illustrated in Figure 6. Picking has been implemented by grabbing the mouse coordinates from Qt [Nok10], using OpenGL's gluUnProject to obtain a world coordinate, then testing to see if this point is inside an approximated bounding box of any character.

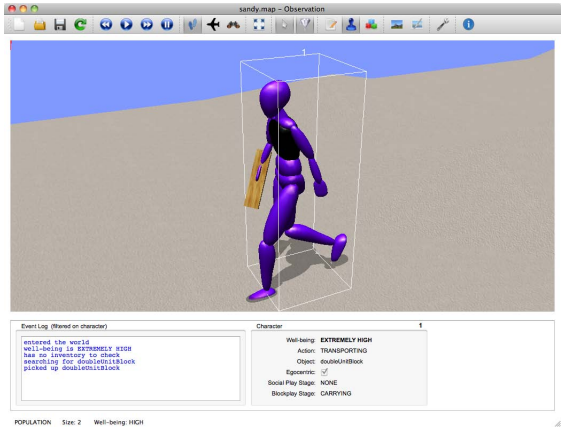

Figure 6: *Character selection, showing a filtered event log and the character information pane.*

3.5. Social

We have implemented the social play stages identified by Parten [Par32], which, although not described here, are: Unoccupied; Solitary; Onlooker; Parallel; Associative; and Cooperative. These relationships are illustrated in Figure 7. A spectating character is connected to an active character with a colour-coded line to match that of the active character. Group members are connected to a group node above their heads, which has a coordinate based upon the average of the group member's coordinates. Each group has a random colour assigned, which is applied to the group node and connecting lines, and also a numerical identifier.

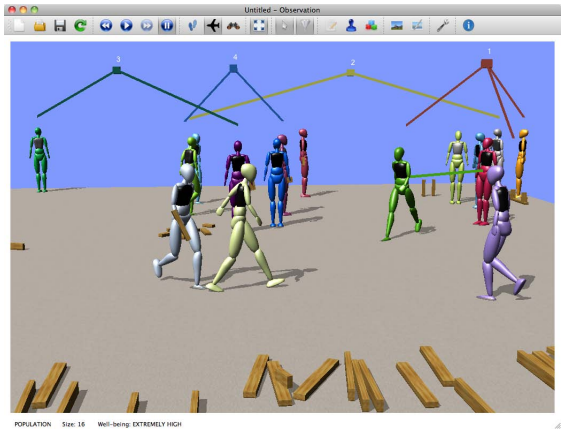

Figure 7: *Groups.*

3.6. Inventory

In order to maintain a collection of blocks we have implemented a personal inventory system, as illustrated in Figure 8. A database is kept of how many of each object a character possesses, rather like a player-character inventory system in a video game. The major benefit is that virtual physical objects occupy no world space at all when they are in inventories, which allows the patch to be used just for displaying constructions. The inventory is the first place a character checks when they are seeking a block, so they may not need to travel to obtain it. Inventories are shared in Associative and Cooperative groups.

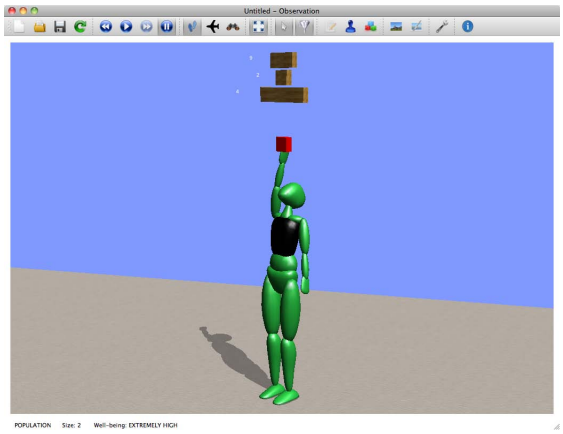

Figure 8: *Inventory system. The character reaches up for a half unit block because there is one in the inventory.*

4. Environment

New scenes can be constructed quickly by adjusting two sliders, which modify the ground plane along two dimensions.

Typically, zones of activity within an early childhood setting determine where certain resources are based, and the flooring appropriate for the related activities [Com06]. Our sandbox application enables the user to create their own scenes very quickly, so this constraint was relaxed, to allow characters and objects to be added anywhere. To facilitate observation we have not implemented a building, dividing walls, or doors. This is similar to the practice of free movement between the activity areas found in children's centres.

A terrain generator can produce random height variations in the scene from terrain ranges defined in a map file, which is particularly useful for outdoor areas. Terrain vertices can be manipulated directly to modify the height of the terrain, even when the world is populated with characters and physical objects, as illustrated in Figure 9. If the vertices fall on a character's patch, they are locked, and cannot be selected. Hollow blockplay, which uses larger wooden blocks than unit blocks, usually occurs outside, so the facility to represent terrain is useful. Textures can be applied to the terrain at any time.

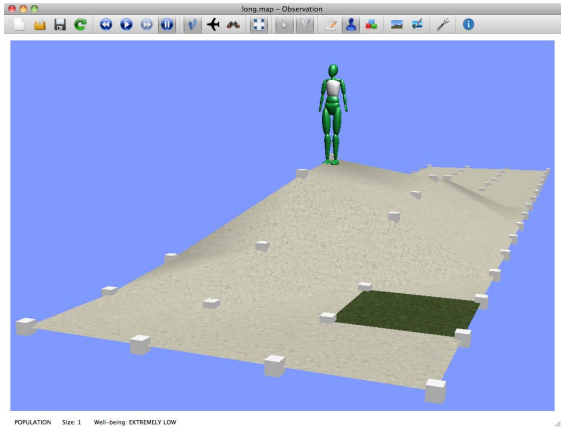

Figure 9: *Character following the contours of the terrain as the terrain vertex heights and textures are being edited.*

With the exception of physical geometry, all rendered geometry is offset from the terrain using bilinear interpolation of the four underlying terrain vertices that form a cell of the uniform grid.

Whilst we do want characters to walk up and down variable height terrain, we do not want this to happen when the terrain appears too steep to do so. To accomplish this, whenever the terrain is modified, a set of vertices is overlaid onto the surface of the terrain, at the same resolution as the occupancy grid used for A* pathfinding. We check for the differences in heights between neighbouring vertices and if the difference is beyond a certain threshold, the pair is flagged as steep. The higher vertex in a steep pair is then recorded in the occupancy grid as blocked. When characters plan a route through the scene, they will disregard any areas that are considered too steep, as illustrated in Figure 10 by the red areas.

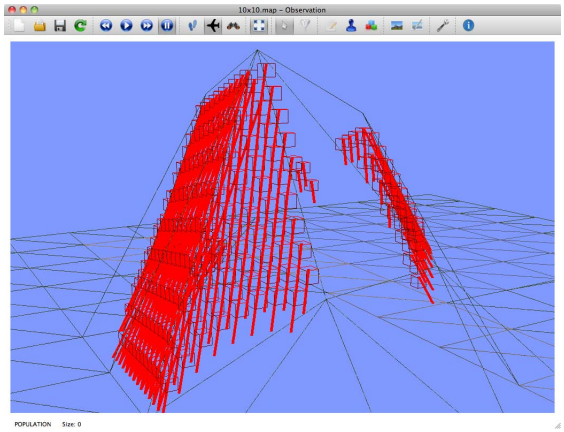

Figure 10: *Geometry calculated as steep.*

To ground characters and objects onto the terrain, shadows are created using OpenGL stencilling with the accepted limitation that this particular method is for planar use only. For characters, there is also the option to display a Blob Shadow, an early video game technique, where some coloured and/or textured geometry (in this case a triangulated circle), is displayed constantly beneath the character's feet.

5. Viewing

The observer could be embodied, fully immersed in the scene, rather like a video game, colliding with scenery, characters and objects. However, we have not represented them as a 3D character; rather, viewing in the graphics pane is designed to be through the eyes of the observer.

Viewing options include: walking; flying; and following a character. For walking and flying, rotation is constrained to pitch and yaw. When walking, since females constitute the majority of early childhood professionals, the observer is based on the terrain with an eyepoint set at 1.6 metres initially (illustrated in Figure 11), which is approximately the average height for an adult female in the United Kingdom [The08]. The height can be modified to match that of the standing observer by dragging a slider. In fact, the height can be set to any value all the way down to just above the terrain level.

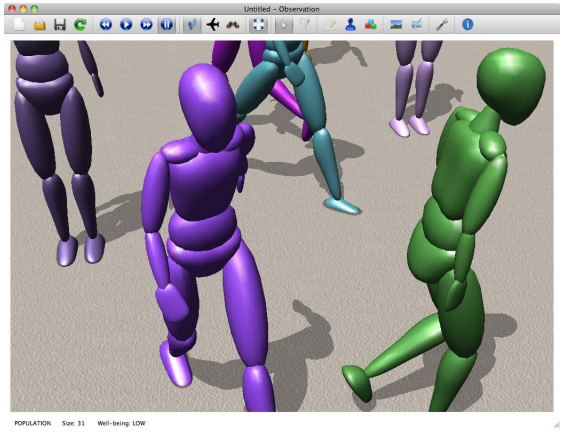

Figure 11: *View when walking with eyepoint at 1.6 metres.*

The observer can optionally have their viewpoint rise and fall relative to the contours of the terrain. Flying takes advantage of using a computer simulation to do things that would not be possible in real life, by enabling the observer to have a top-down view of the world, as illustrated in Figure 12.

Multiple viewports have not been used as these were considered too similar to Closed Circuit Television (CCTV) surveillance. There is, however, the option to follow a selected character, where the observer's viewpoint is calcu-

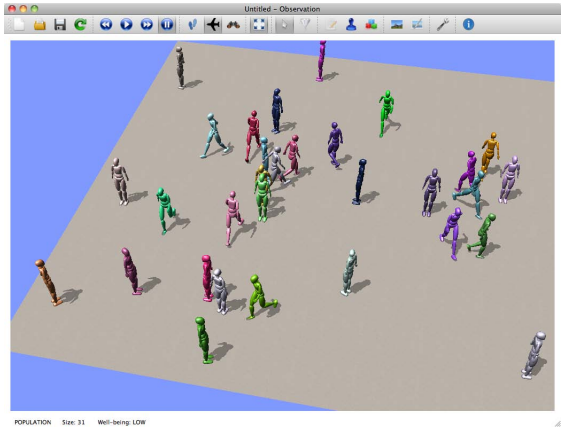

Figure 12: *View when flying.*

lated automatically, and focussed upon the position of the centred character.

6. Objects

Objects in the scene can be selected, picked up, and placed down by the observer at any time. Selected objects can be moved up and down, rotated, and also flipped in 90 degree increments about each axis.

6.1. Definition

Custom shapes are defined in terms of other primitives, inspired by example Virtual Reality Markup Language (VRML) worlds [ANM96]. For example, a unit block, as described in Section 1, is represented as *cube unitBlock 140 70 35*, where *cube* has been defined within Observation. This allows further custom shapes such as blocks of any dimensions to be defined, and referred to by name when creating map files.

3D models such as furniture can also be defined, provided that they are already triangulated and in the OBJ file format. If texture coordinates are not specified, they will have any specified texture mapped linearly to the geometry. A cuboid collision shape (and subsequent rigid body) will be generated for the mesh in the physics simulation, based upon its bounding box, as it is added to the scene.

Objects that have been defined, can be selected from a dialog box accessible from the toolbar. The selected object will appear as being within the grasp of the observer, represented by the closed hand cursor icon, until it is released into the scene.

6.2. Physics

The physics library Bullet [Cou10] has been integrated into Observation, which gives added realism to objects in motion and at rest, as illustrated in Figure 13 on the terrain and shelves. Four vertical planes (which are effectively invisible walls) contain the dynamic physics objects within the scene. The triangles that constitute the terrain are passed to Bullet to form a static triangular collision shape and rigid body. When objects are added to the scene, a collision shape is created based upon the bounding box, and used for all objects of that type. Shelf objects have a static collision shape defined and can have other objects rest on them. All other objects are defined as dynamic, subject to physics, colliding with other objects in a natural way. All physical geometry has a rigid body based upon the collision shape, shown in wireframe view as green in Figure 13. For dynamic objects, transforms are obtained from motion states for spatial indexing and rendering the objects. Picking up and putting down of objects is handled by the destruction and creation of dynamic rigid bodies.

7. Data-driven scripting

Customisation is possible, by taking advantage of data held in external text files. Maps (scenes) can be made containing the terrain and its textures, and objects. Scripted construction plans can be defined in plan files. Maps and plan files both reference custom shapes and models, defined in other files.

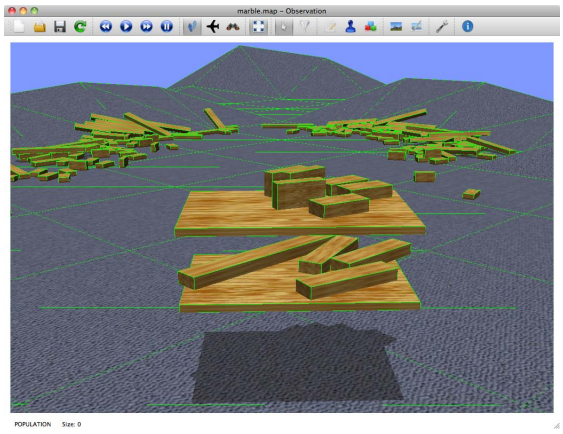

Figure 13: *Physical geometry.*

8. User experience and evaluation

In the first semester of the academic year 2010-2011, a focus group meeting was held with early childhood lecturers at Liverpool Hope University, during which a semi-structured walkthrough of Observation was given. The format of the simulation was considered intriguing and it was believed that such an environment would appeal to today's learners. In the second semester, after further development which had taken on board other comments made by the lecturers, we performed a pilot evaluation which included an online survey and focus group with some early childhood students at

the same university. Students agreed that it was useful: to have an influence on the well-being of the characters (8/9); to be present among the characters (7/9); to adjust their viewing position (6/9); and to define the behaviour of the characters (5/9). Students reported increased understanding in the stages of blockplay (5/9). In the focus group, several were very enthusiastic about changing the environment.

9. Conclusion

We have described our prototype application designed for simulated observations and looked at some of the visualisation options within it. Feedback from lecturers and students within the field of early childhood has been positive, suggesting that an interactive real-time 3D graphics simulation may have something to offer observers of young children that traditional methods such as text and video do not. We intend to roll out a large-scale survey to include more early childhood students, plus widen the audience to include those outside the field.

Observation is available for download and evaluation from: `http://www.cs.man.ac.uk/~aac/observation/`.

References

[ACH09] ALBIN-CLARK A., HOWARD T. L. J.: Automatically Generating Virtual Humans using Evolutionary Algorithms. In *Proceedings of Theory and Practice of Computer Graphics* (June 2009), Eurographics, pp. 61–64. 2

[AN07] AHLQUIST J. B., NOVAK J.: *Game Development Essentials: Game Artificial Intelligence*, first ed. Delmar Cengage Learning, July 2007. 3

[ANM96] AMES A. L., NADEAU D. R., MORELAND J. L.: *The VRML Sourcebook*. John Wiley & Sons Inc., 1996. 7

[Ble10] BLENDER FOUNDATION: *Blender version 2.48a (3D content creation suite)*. Available from: http://blender.org/. Accessed: December 2010, 2010. 3

[BM06] BRUCE T., MEGGITT C.: *Child Care and Education*, fourth ed. Hodder Arnold, May 2006. 1

[BS04] BOURG D. M., SEEMANN G.: *AI for Game Developers*, first ed. O'Reilly Media, July 2004. 3

[Ces08] CESSEN: *Biped rig*. Available from: http://blenderartists.org/. Accessed: December 2010, 2008. 2

[Com06] COMMUNITY PLAYTHINGS: *Spaces: Room layout for early childhood education*. Community Playthings, 2006. 5

[Com10] COMMUNITY PLAYTHINGS: *2011 Catalogue*. Community Playthings, 2010. 1

[Coo07] COOPER R.: *Alter Ego: Avatars and their Creators*. Chris Boot, May 2007. 2

[Cou10] COUMANS E.: *Bullet version 2.76 (physics library)*. Available from: http://bulletphysics.org/. Accessed: December 2010, 2010. 7

[Cur10] CURVESHIFT INC.: *simSchool (classroom simulation software)*. Available from: http://www.simschool.net/. Accessed: December 2010, 2010. 2

[Dep08] DEPARTMENT FOR CHILDREN, SCHOOLS AND FAMILIES: *Practice Guidance for the Early Years Foundation Stage*. Department for Children, Schools and Families, May 2008. 1

[Eps08] EPSTEIN J. M.: Why Model? *Journal of Artificial Societies and Social Simulation 11*, 4 (2008), 12. 2

[FJMM79] FALENDER C. A., JORDAN-MARSH M., MURAR R.: The simulation model for child care programs: Expanding staff-child-parent development. *Child and Youth Care Forum 8*, 2 (June 1979), 113–125. 2

[FKC*04] FERRY B., KERVIN L., CAMBOURNE B., TURBILL J., PUGLISI S., JONASSEN D., HEDBERG J.: Online classroom simulation: The next wave for pre-service teacher education? In *Beyond the comfort zone: Proceedings of the 21st Australasian Society for Computers in Learning in Tertiary Education (ASCILITE) Conference* (Dec. 2004), pp. 294–302. 2

[GF92] GURA P., FROEBEL BLOCKPLAY RESEARCH GROUP (Eds.): *Exploring Learning: Young Children and Blockplay*, first ed. Paul Chapman Publishing Ltd, Apr. 1992. 1

[Hir84] HIRSCH E. S.: *The Block Book*, revised ed. National Association for the Education of Young Children, 1984. 1

[Isb06] ISBISTER K.: *Better Game Characters by Design: A Psychological Approach (The Morgan Kaufmann Series in Interactive 3D Technology)*. Morgan Kaufmann, June 2006. 2

[Isr11] ISRAEL J.: *Open Icon Library (icon collection)*. Available from: http://openiconlibrary.sourceforge.net/. Accessed: February 2011, 2011. 2

[Lin10] LINDEN RESEARCH, INC.: *Second Life BVH animations*. Available from: http://static.secondlife.com/downloads/avatar/bvh_files.zip. Accessed: December 2010, 2010. 4

[LVKD02] LAEVERS F., VANDENBUSSCHE E., KOG M., DEPONDT L.: *A Process-Oriented Child Monitoring System for Young Children*. Experiential Education. Centre for Experiential Education, 2002. 1, 3

[MP02] MERLEAU-PONTY M.: *RC Series Bundle: Phenomenology of Perception (Routledge Classics)*, 2 ed. Routledge, May 2002. 2

[Nic07] NICOL J.: *Bringing the Steiner Waldorf Approach to your Early Years Practice*. David Fulton Publishers, May 2007. 2

[Nok10] NOKIA CORPORATION: *Qt version 4.7 (application and user interface framework)*. Available from: http://qt.nokia.com/. Accessed: December 2010, 2010. 4

[Par32] PARTEN M. B.: Social participation among pre-school children. *The Journal of Abnormal & Social Psychology 27*, 3 (Oct. 1932), 243–269. 5

[Ree10] REE Z.: *QAvimator version January 2010 (BVH animation editor)*. Available from: http://qavimator.org/. Accessed: December 2010, 2010. 4

[Rob00] ROBINS N.: *GLM (Wavefront OBJ model file format manipulation library)*. Available from: http://xmission.com/~nate/tutors.html. Accessed: December 2010, 2000. 3

[The08] THE NHS INFORMATION CENTRE FOR HEALTH AND SOCIAL CARE: *Health Survey for England 2008*. Available from: http://www.it.nhs.uk/. Accessed: April 2011, 2008. 6

[WK00] WELLHOUSEN K., KIEFF J.: *A Constructivist Approach to Block Play in Early Childhood*, first ed. Wadsworth Publishing, Dec. 2000. 1