Chapter 3

"The ability to form 'abstract' concepts is probably the basis of man's ability to reason"

- Edward de Bono [37]

Theoretical approach

In this chapter, the theoretical approach and underlying concepts of the Intelligent Tiles (iTiles) virtual laboratory framework are presented. Firstly, the chapter introduces the iTiles framework and secondly the aims of the framework are presented. Next the key theoretical concepts of the iTiles framework are discussed: the components, and behaviour of an iTiles world. These concepts are explained in more detail in the sections that follow: transforming an iTiles world, simulating behaviour for an iTiles world and the movement of characters in the simulation. The chapter ends discussing how young learners can learn with an iTiles virtual laboratory and educational advantages of the iTiles framework are highlighted.

3.1 The Intelligent Tiles virtual laboratory framework

The main research of this thesis involves the creation of a framework that can be used for the design and development of a virtual laboratory, aimed for use in the education of young learners. The virtual laboratories that can be developed through the use of this framework are ecosystem type virtual environments used in teaching biological subjects such as earth science. These virtual environments are worlds consisting of a terrain inhabited by a set of virtual life forms.

3.1.1 Introduction to the framework

Virtual laboratory solutions today are beginning to present students with a kind of interactive encyclopaedia and are becoming more accepted for their benefit in the classroom [39].

The Intelligent Tiles (iTiles) framework, originally introduced in [39], is a method and approach that can be used for the development of a teacher driven virtual laboratory. The framework forms a foundation for a virtual environment that allows for deviations, and accommodates extensions to the framework. Being a framework for a virtual laboratory, the framework attempts to cover both virtual reality and educational aspects involved.

The main principle of the iTiles framework is the use of tiles to author a virtual environment. The virtual environment is simple, without much complexity and little technical know-how and understanding is needed for it to be applied as a teaching aid. The virtual environment produced using the concepts in the iTiles framework is called an **iTiles world**.

The use of a virtual laboratory produced with the iTiles framework, in an educational setting, involves three simple steps (see Figure 12):

1. Author the iTiles world.
2. Specify the behaviour of the world objects that inhabit the authored iTiles world.
3. Present a simulation of the authored iTiles world to young learners to interact and learn from using an experiential/learning-by-doing pedagogy.

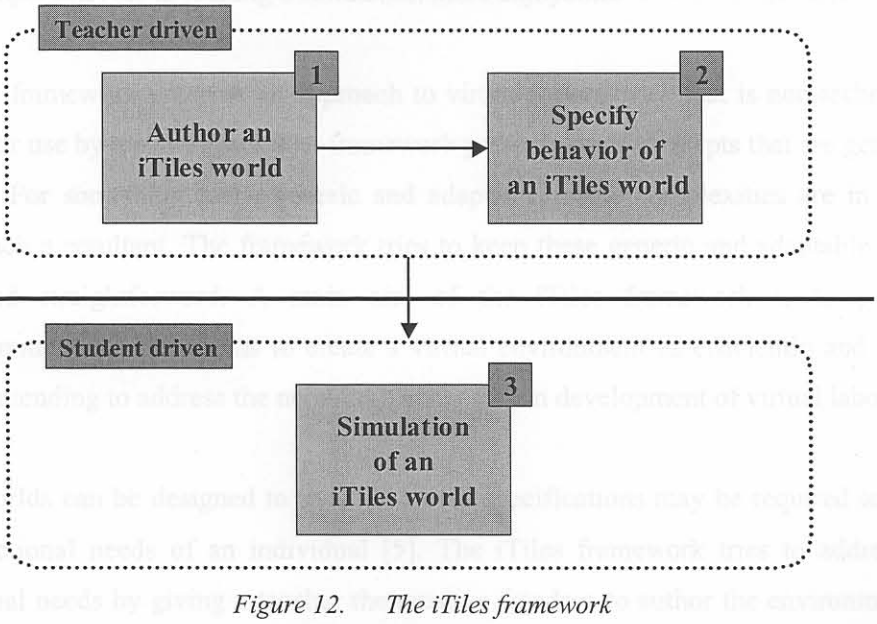

Figure 12. The iTiles framework

Steps 1 and 2 are teacher driver, while step 3 is for students. More advanced students could perform steps 1 and 2.

3.1.2 Aims of the iTiles framework

The main objective of the iTiles framework is to aid the education of elementary school children by facilitating the teacher driven development of virtual laboratories. In addressing the major problems involving current virtual laboratories, the iTiles framework tries to address the problems through the following aims and objectives:

The iTiles framework makes use of virtual environments and multimedia, to support science and technology education and aims to advance the educational use of virtual laboratories. It facilitates the learning process by helping children make sense of the world they live in, in ways that were not possible up to this time, which are uniquely possible by using educational technology as a medium. The framework addresses the current cost and limited accessibility of immersive VR systems, by allowing development of virtual laboratories on a platform of broader use (e.g. Desktop VR). The framework also involves virtual reality concepts, both audio and visual elements, and is aimed for the development of a multimedia and interactive experience. It considers important visualisation and interactivity factors that play a major role in the user experience of a virtual environment. Not only does the framework support visualisation, but it also incorporates sound, which presents a more immersive world for the user to explore. Herewith a world can "come to life" in a more advantageous way since sound plays an important role in making a simulation more enjoyable.

The iTiles framework presents an approach to virtual laboratories that is non-technical and intended for use by a non-expert. The framework presents some concepts that are generic and adaptable. For something to be generic and adaptable, some complexities are in order to achieve such a resultant. The framework tries to keep these generic and adaptable qualities simple and straightforward. A main aim of the iTiles framework is to provide a nonprogrammer with the means to create a virtual environment as efficiently and easily as possible, intending to address the need for teacher driven development of virtual laboratories.

Virtual worlds can be designed to meet whatever specifications may be required to address the instructional needs of an individual [5]. The iTiles framework tries to address these instructional needs by giving a teacher the creative freedom to author the environment for a lesson and specify the behavioural aspects of the environment. If learning is made more interesting and fun, students may remain engaged in an activity for longer periods of time. Research to date indicates that virtual worlds of colour, shape, sound, and feel should amplify the powers of the mind to see complex sets of data and to absorb, manipulate, and interpret information more quickly and completely [4]. Using the iTiles framework, it is aimed that such virtual worlds be produced.

3.2 Composition of an iTiles world

To understand the composition an iTiles world, the components of the virtual environment are explained in this section. The **components** of an iTiles world are **tiles**, and **world objects**.

The **tile components** form the basis of an iTiles world. Each tile is square in dimension and is associated with an element type. These element types are analogous to geographical terrain types that appear in the real world, for example grass and sand. A tile's element type is visually represented by a unique texture or a specific colour. For any iTiles world a tile set needs to be considered with unique tile element types being specified. Tiles elements are grouped together as a two dimensional (2D) matrix layer of tiles, to form the base terrain of an iTiles world. The size of the world is variable and is dependant on the number of rows and columns specified for the matrix when an iTiles world is authored. The visual representation of the world is established by presenting this flat 2D base terrain in a three dimensional (3D) environment. By applying a 2D tile system in a 3D environment, there are many advantages gained. 2D tiled worlds can only be scrolled and there is usually a fixed viewpoint. In 3D, a feeling of depth is achieved since the world can be zoomed in and out, and can also be rotated on an axis for a different point of view [39].

The **world object components** of an iTiles world are representations of real world and imaginary objects that populate the base terrain of the world, and exist on top of individual tiles. In an iTiles world each tile entity of the base terrain may either be free, or may be occupied by an individual world object. World objects are visually represented by 3D models. These 3D model representations may require scaling in order to be placed on top of tiles in proportional dimensions. Therefore each world object has a minimum and maximum scale specified. A world object may face one of eight different directions on a tile, as it can face any of the tile's four edges or corners.

The world object components have been divided into two categories: static and dynamic world objects. **Static world objects** are immovable world objects that have fixed positions in the environment. An example of a static world object is a tree or a rock. **Dynamic world objects** are world objects that can move in the environment. Dynamic world objects are analogous to virtual **characters**, and will be referred to as either characters or dynamic world objects throughout this and the following chapters of this thesis/dissertation.

World objects are placed on top of tiles based on their **tile attribute list**. This tile attribute list is a list of the tile elements that a world object may be placed on in the environment. In the case of dynamic world objects this list also determines what tile elements the dynamic world objects may move on. For example, a dynamic world object with sand and water tile elements in it's tile attribute list can only move on tile elements with the property of sand or water. This tile attribute list also determines the collision detection model, since dynamic world objects will not be able to move on tile elements not listed in their tile attribute list.

An iTiles world does not only come to life by characters moving in the world, but also by the sound they emit and behaviour they exhibit when interacting with the world, other characters and static world objects. The sound features of the environment and behaviour of world objects are presented next.

3.3 Behaviour of an iTiles world

Simulation of a virtual environment involves the animation and movement of the characters that inhabit that virtual environment. Movement can be a result of behaviour and internally defined attributes, such as the relationships of a character with other components of the virtual environment. The **iTiles World Flow** of an iTiles world can be described as the forces of nature, or a specification of a set of rules by which an authored iTiles world must adhere. It is needed for the simulation of the motion and behaviour of characters, for governing the flow of movement of characters, and defining the procedure for the interaction of characters. The properties associated with other components of the environment, such as static world objects and tiles are also specified.

Each iTiles world requires an iTiles World Flow to be defined, since the iTiles World Flow is used as input to achieve the simulation of that iTiles world. Figure 13 illustrates the key iTiles World Flow concepts that are discussed in this chapter. These concepts are:

- *World transformations*. World transformations define visual and auditory changes that can occur in the simulation of an iTiles world. The way a tile element can transform is defined by a tile transformation. Both static and dynamic world objects may have world object transformations defined, indicating how they may transform in the simulation of the iTiles world.
- *World forces*. World forces define the relationships of a dynamic world object with other components of the virtual environment, in terms of the dynamic world object's behaviour and sound emitted. There are forces of attraction called positive forces, forces of

repulsion called negative forces, and forces that represent the relationship of a character's movement on tile elements, called movement forces. Dynamic world objects may have one or more positive forces, negative forces or movement forces specified. Positive forces and movement forces may have an action property defined. When these actions are executed they can trigger world transformations to occur.

These iTiles World Flow concepts are presented in more detail in the sections that follow. World transformations are presented in Section 3.4, followed by a presentation of world forces in Section 3.5

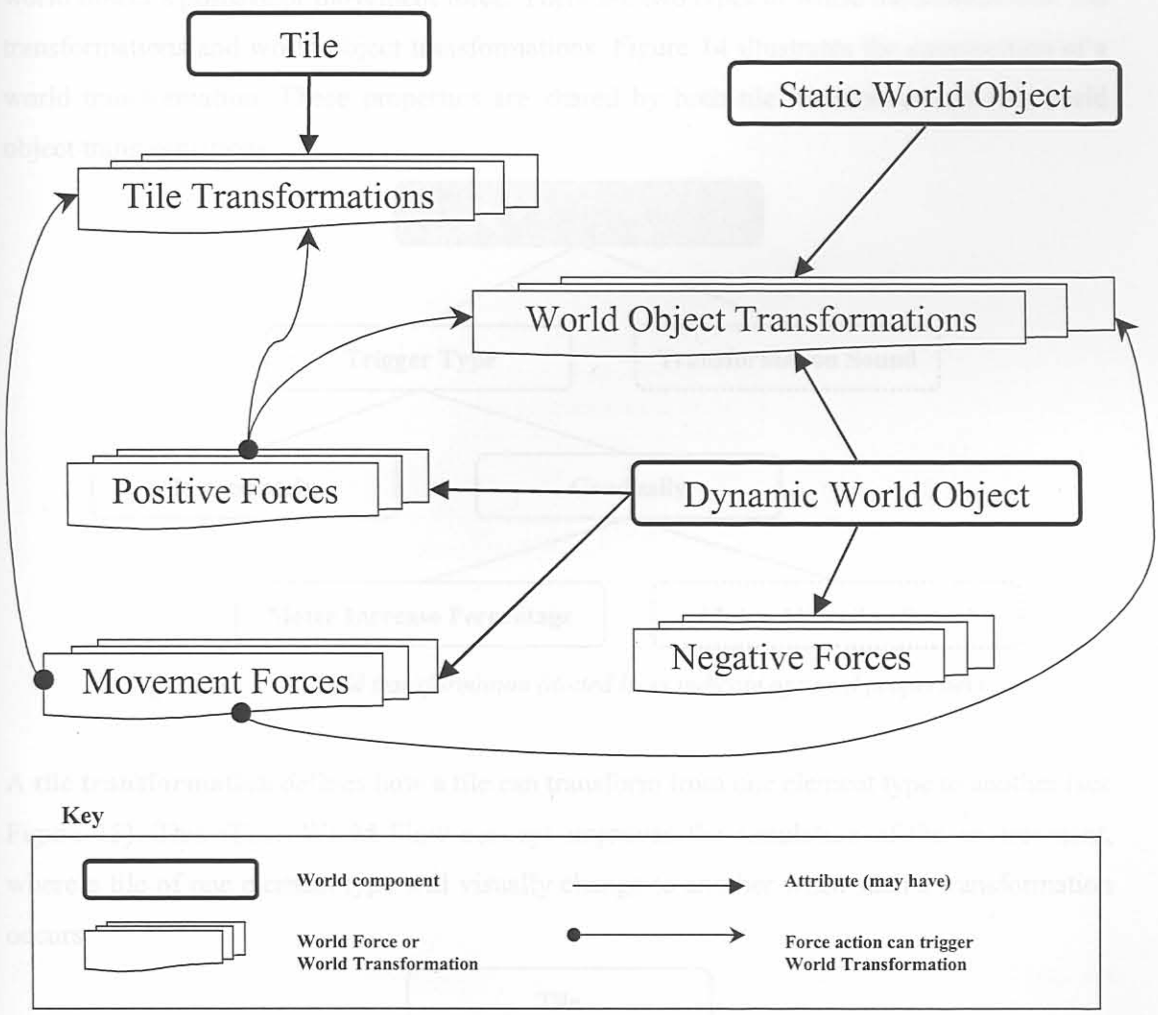

Figure 13. iTiles World Flow summary

In the simulation of an iTiles world, movement is dependent on time. The **iTiles world beat** is a unit of measurement of time used in the simulation of an iTiles world. A single world beat represents one time unit and could be any amount of real time. By changing the amount of time that a world beat represents, the simulation can be slowed down or hurried up. The iTiles World Flow concepts are time dependent and their time attributes are measured in world beats. Animating the movement of characters in the simulation is related to simulation time, so that it is not dependent on the processing power of the underlying computer hardware. The movement of characters is then performed every iTiles world beat. The movement of characters is discussed in more detail in Section 3.6.

3.4 Transforming an iTiles world

An iTiles **world transformation** represents a visual change or an auditory event that can occur in the simulation of an iTiles world. World transformations are associated with tile elements, world objects and sounds, and are triggered to occur by an action of a dynamic world object's positive or movement force. There are two types of world transformations: tile transformations and world object transformations. Figure 14 illustrates the composition of a world transformation. These properties are shared by both tile transformations and world object transformations.

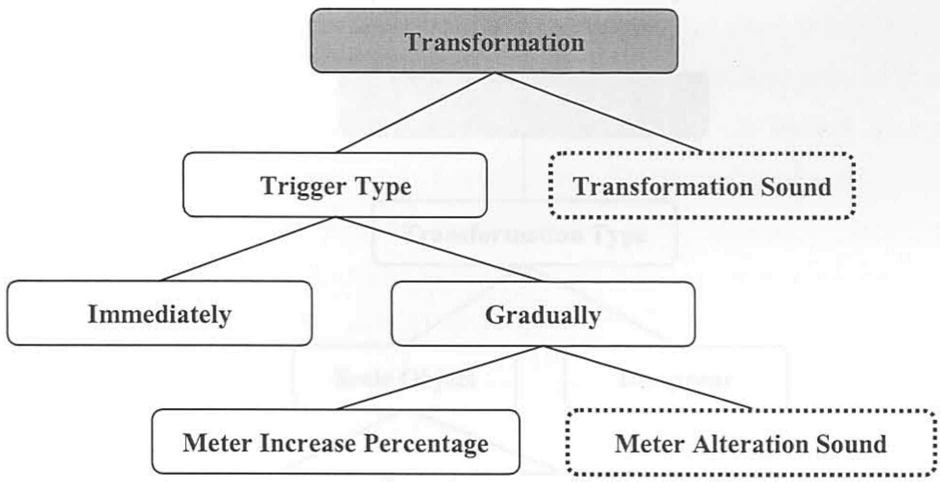

Figure 14. A world transformation (dotted lines indicate optional properties)

A **tile transformation** defines how a tile can transform from one element type to another (see Figure 15). This iTiles World Flow concept improves the simulation of the environment, where a tile of one element type will visually change to another when such a transformation occurs.

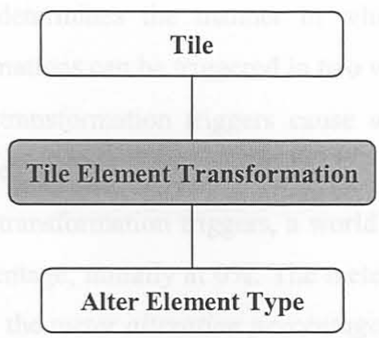

Figure 15. Tile element transformation

A **world object transformation** defines how a world object may transform in the simulation. Three types of transformations exist for a world object (see Figure 16):

- *Scale up.* World objects can transform to be scaled up by a certain percentage. In order to retain visual conformation, world objects will continuously scale up until their maximum scale is reached.
- *Scale down.* World objects can transform to be scaled down by a certain percentage. In order to retain visual conformation, world objects will continuously scale down until their minimum scale is reached.
- *Disappear.* World objects can be removed from the simulation. Once a world object has disappeared from the world, it shall be removed from the tile it is currently on, from the simulation and not reappear.

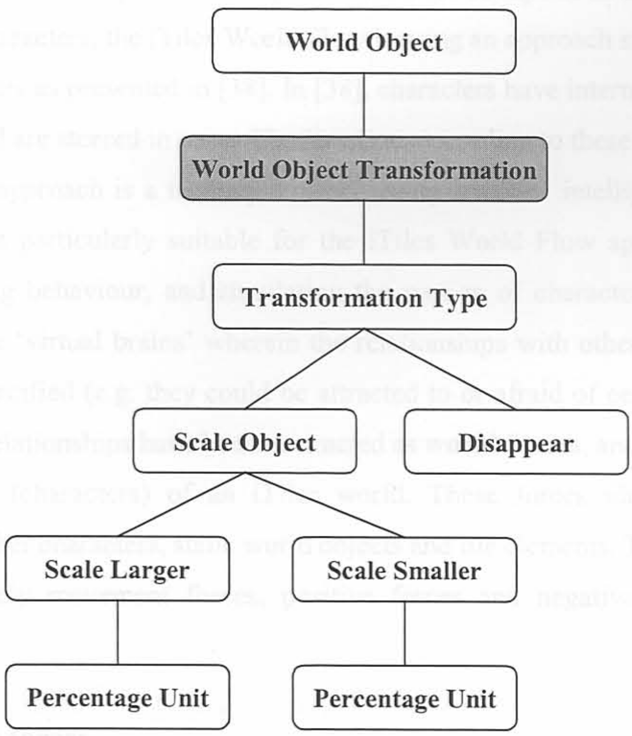

Figure 16. World object transformation

A **transformation trigger** determines the manner in which world transformations are influenced to occur. Transformations can be triggered in two ways:

- *Immediately.* Immediate transformation triggers cause world transformations to occur immediately when triggered.
- *Gradually.* With gradual transformation triggers, a world transformation contains meter, which is stored as a percentage, initially at 0%. The meter's value can be increased by a certain percentage, called the *meter alteration percentage*, specified with the influencing

triggering force. The world transformation only occurs when the meter reaches full capacity at 100%.

World transformations can have an audio representation where a sound clip may be played when the transformation occurs (called the *transformation sound*). With world transformations that have gradual transformation triggers, a sound clip may be specified that is played when the meter is influenced (called the *meter alteration sound*).

3.5 Simulating behaviour for an iTiles World

Sophisticated algorithms and AI techniques are required to define the behaviour and movement of a virtual character. The advantage of using these techniques is that the number of possible outcomes is nearly unlimited, given a rich set of inputs. In achieving movement and behaviour of characters, the iTiles World Flow is using an approach similar to the method of steering behaviours as presented in [38]. In [38], characters have internal forces that define their movement, and are steered in a specific direction, according to these internal forces. The steering behaviour approach is a technique of achieving artificial intelligence with minimal intelligence. This is particularly suitable for the iTiles World Flow approach. In order to achieve this steering behaviour, and simulating the motion of characters, characters of an iTiles world require 'virtual brains' wherein the relationships with other components in the environment are specified (e.g. they could be attracted to or afraid of certain components in the world). These relationships have been abstracted as **world forces**, and are only applicable to moving objects (characters) of an iTiles world. These forces identify a character's relationship with other characters, static world objects and tile elements. There are three types world forces, namely movement forces, positive forces and negative forces, which are discussed below.

3.5.1 Movement forces

The movement force of a character (see Figure 17) represents how the movement of that character affects the tile elements it is moving on. This movement force may also reflect the manner in which the character is influenced moving on a certain tile element type. Movement forces may be specified for any of the tile element types appearing in a character's tile attribute list. A movement force has an action, which when executed can trigger a tile transformation of the tile moved on to occur. This action may also trigger a world object transformation that belongs to the character to occur. If any of the trigger types of the world transformations specified in the movement force's action are of the gradual type, a meter alteration percentage must be specified, indicating the percentage by which the meter of these world transformations will increase. The action of a movement force is executed when the character moves on or to a tile with a movement force specified for that tile element type. As an example of a movement force, consider a movement force of a dog character on grass. When the dog moves on a patch of grass this can trigger the tile transformation of grass to transform to sand in a gradual manner with a meter alteration percentage of 15% for the gradual transformation trigger type. The more the dog character moves on that patch of grass, the faster it will transform to sand. A movement force may also have an audio representation called a *movement sound*. The movement sound is played when a character moves on or to a tile with a movement force specified for that tile element type. For example, a water splashing sound can be played as a duck moves on water.

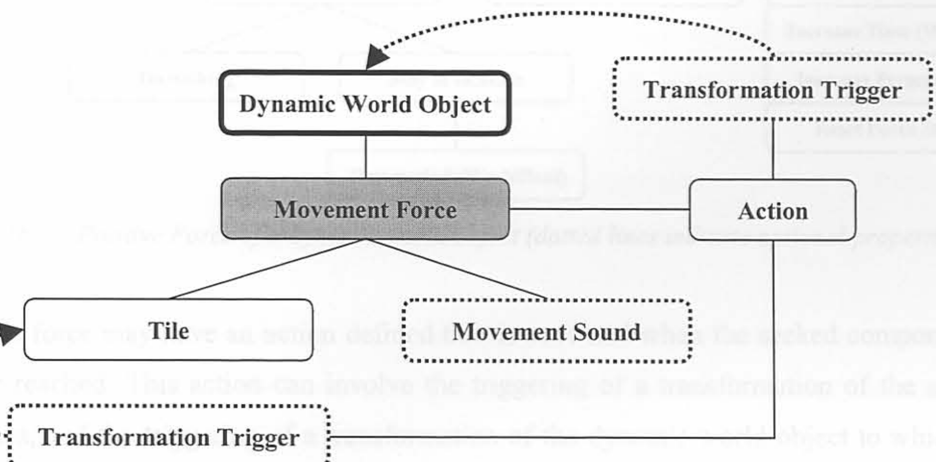

Figure 17. Movement force of a dynamic world object (dotted lines indicate optional properties)

3.5.2 Positive forces

A positive force of a dynamic world object (see Figure 18) defines a dynamic world object's force of attraction towards a component of the environment (i.e. a dynamic world object can have a positive force towards a static world object, a dynamic world object, or a tile element type). A positive force is expressed in terms of a force strength (in percentage), which indicates the strength by which the dynamic world object seeks the component to which it is attracted.

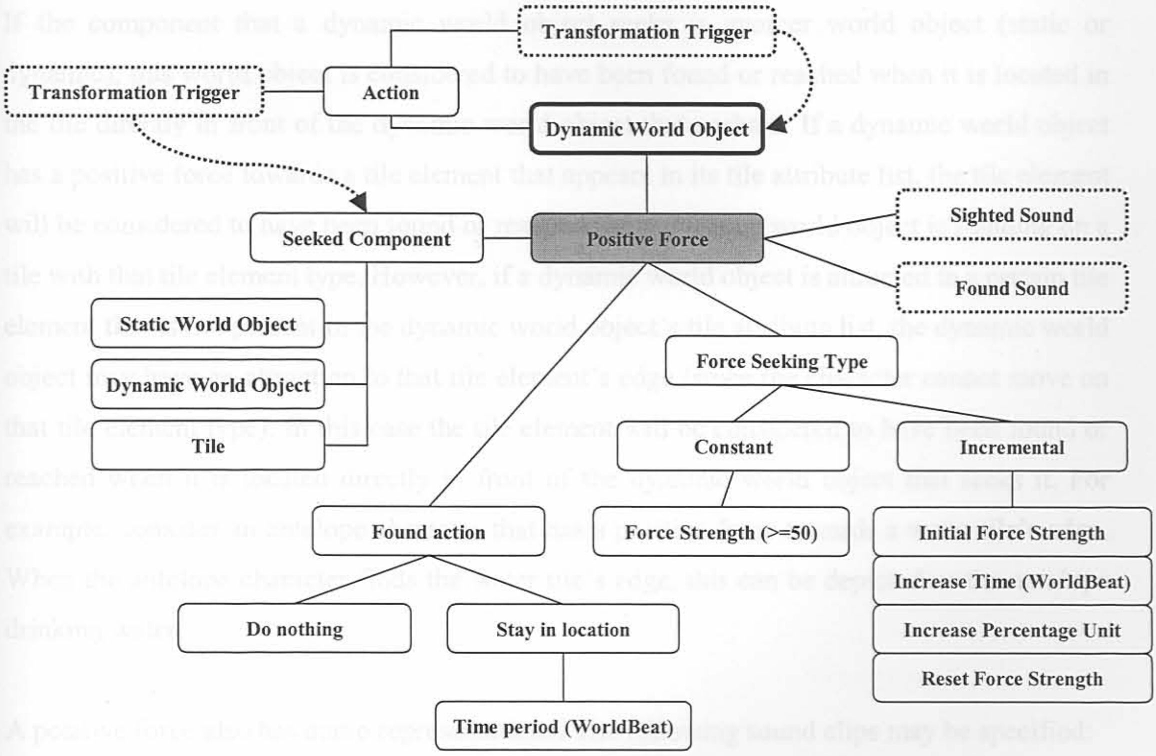

Figure 18. *Positive Force of a dynamic world object (dotted lines indicate optional properties)*

A positive force may have an action defined that is executed when the seeked component is found or reached. This action can involve the triggering of a transformation of the seeked component, and the triggering of a transformation of the dynamic world object to which the positive force belongs. This action therefore defines how the positive force has an influence on the seeked component and on the character to which the force belongs. When a dynamic world object reaches or finds the seeked component, there are two types of 'found actions' that may occur, which precede an action's execution:

- *Do nothing.* The action will therefore execute immediately
- *Stay in location.* Specifying that a dynamic world object stay in the location for a specified number of world beats indicates a time period that will occur before the positive force's action is executed. Here the dynamic world object will remain frozen in the same location for the number of world beats specified when the seeked component is found. If the seeked component is another dynamic world object, it too will be frozen for that time period (i.e. a dynamic world object can freeze another dynamic world object). The action will only be executed once this time has elapsed. Once the action has been executed the dynamic world object will be unfrozen and continue as normal in the simulation. If the seeked component is another dynamic world object, this dynamic world object would too be unfrozen.

If the component that a dynamic world object seeks is another world object (static or dynamic), this world object is considered to have been found or reached when it is located in the tile directly in front of the dynamic world object that seeks it. If a dynamic world object has a positive force towards a tile element that appears in its tile attribute list, the tile element will be considered to have been found or reached if the dynamic world object is standing on a tile with that tile element type. However, if a dynamic world object is attracted to a certain tile element that is not present in the dynamic world object's tile attribute list, the dynamic world object may have an attraction to that tile element's edge (since the character cannot move on that tile element type). In this case the tile element will be considered to have been found or reached when it is located directly in front of the dynamic world object that seeks it. For example, consider an antelope character that has a positive force towards a water tile's edge. When the antelope character finds the water tile's edge, this can be depicted as the antelope drinking water.

A positive force also has audio representations. The following sound clips may be specified:

- *Sighted sound.* A sound clip may be played when a character sees the seeked component directly in front of it.
- *Found sound.* A sound clip may be played when a character finds or reaches the seeked component. Only the found sound (not the sighted sound) is played when the character finds or reaches the seeked component.

There are two types of positive forces which influence how the force strength (the level of attraction towards the seeked component) may change: constant seeking and incremental seeking. A **constant seeking positive force** has a constant force strength specified between 50% and 100%, which remains unchanged throughout the simulation. Figure 19a shows the fuzzy logic interpretation of this force strength scale. With an **incremental seeking positive force**, the force strength increases linearly over time (i.e. the force gets stronger over time) by a specified percentage unit. This positive force resets to a certain percentage once the seeked component is found and the positive force's action has been executed. An incremental seeking positive force has the following attributes (see Figure 19b, 19c and 19d for the fuzzy logic interpretation of these attributes):

- *Initial force strength.* The force strength that will be allocated to a positive force at the start of the simulation. An initial force strength of 50% is considered of weak strength, whereas a force strength of 100% is considered the strongest.
- *Increase time unit.* This is the amount of time that will pass till the force strength will increase by the increase percentage unit. This time unit is measured in world beats.

- *Increase percentage unit.* The percentage by which the force strength will increase. The more force strength specified for the increase percentage unit, the stronger the resultant force strength will become.
- *Reset force strength.* This is the force strength to which the positive force will reset once the seeked component is reached or found. A force strength between 0% and 50% is considered to have no strength. A force strength of 50% is considered of weak strength, whereas a force strength of 100% is considered the strongest.

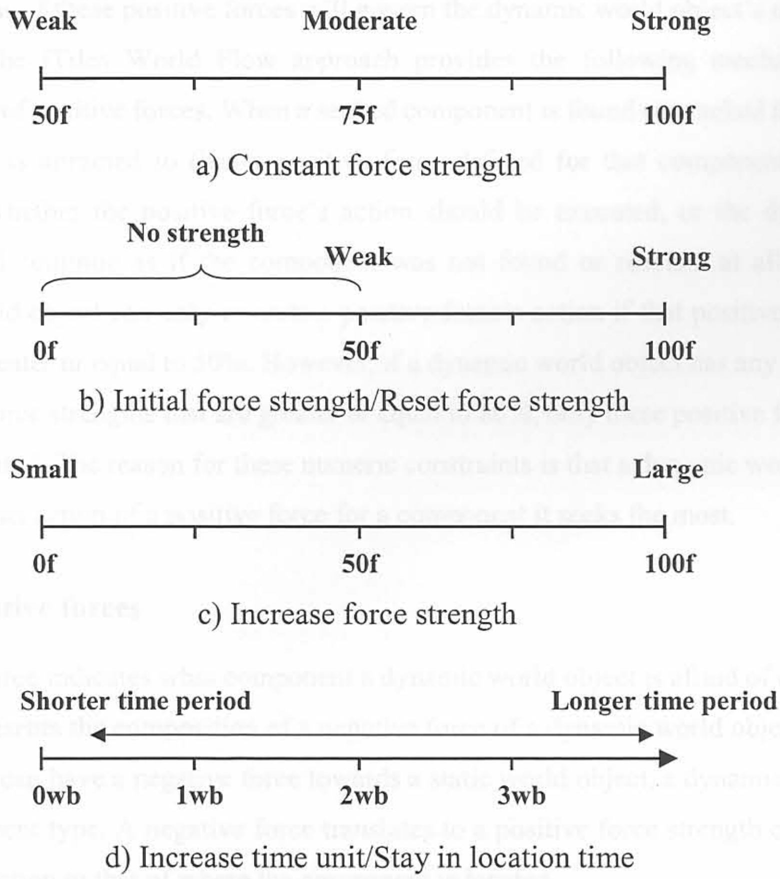

Figure 19. Constant and incremental positive forces' metric unit features explained as fuzzy logic (The unit 'wb' refers to world beat and 'f' refers to force strength)

A positive force's incremental force-seeking type can best be explained by using a metaphor of hunger. Consider a person (a dynamic world object) that is a little hungry (initial force strength). Some time passes (increase time) and he/she gets a little hungrier (increase percentage unit). Now in order for the person to satisfy their hunger they start looking for food (the seeked static world object). Eventually the person finds some tasty food in the kitchen and starts eating, which takes some time, as he/she is really hungry (found action type of stay in location for time period). After this time has passed, the person is no longer hungry (reset force strength), the food has disappeared and the person has a full stomach (This positive force's action could specify that when executed the food should disappear by triggering the food static world object's world object transformation of disappearing, and the person character should grow larger by triggering the person character's world object transformation of scaling up). Such a cycle can be continuously repeated for characters in the simulation.

A dynamic world object can be attracted to various components in an environment, and its level of attraction or force strength to those components can vary during the simulation. The force strengths of these positive forces will govern the dynamic world object's movement and behaviour. The iTiles World Flow approach provides the following mechanism for the prioritisation of positive forces. When a seeked component is found or reached that a dynamic world object is attracted to (has a positive force defined for that component), it must be determined whether the positive force's action should be executed, or the dynamic world object should continue as if the component was not found or reached at all. Therefore a dynamic world object can only execute a positive force's action if that positive force's force strength is greater or equal to 50%. However, if a dynamic world object has any other positive forces with force strengths that are greater or equal to 80%, only these positive forces' actions may be executed. The reason for these numeric constraints is that a dynamic world object will only execute an action of a positive force for a component it seeks the most.

3.5.3 Negative forces

A negative force indicates what component a dynamic world object is afraid of or repelled by. Figure 20 presents the composition of a negative force of a dynamic world object. A dynamic world object can have a negative force towards a static world object, a dynamic world object, or a tile element type. A negative force translates to a positive force strength of 100% in the opposite direction to that of where the component is located.

Consider two dynamic world objects: $character_1$ and $character_2$. If $character_1$ is afraid of $character_2$, then a sound clip (called *caught sound*) may be specified that is played when $character_2$ finds or reaches $character_1$. For example, if $character_1$ were an elephant and $character_2$ were a mouse, the *caught sound* could be specified as an elephant trumpeting when the mouse reaches the elephant. A *sighted sound* may also be specified for a negative force, which is played when a character sees the feared component directly in front of it.

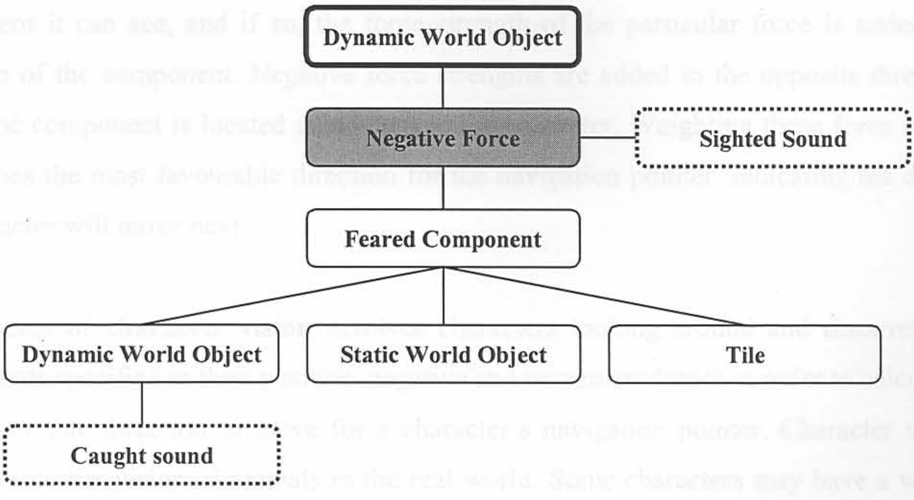

Figure 20. A dynamic world object's negative force

3.6 Movement of characters in the simulation

Characters' movement in an iTiles world can be described as 'robot-like' since characters can either change the direction they are facing (in 45° angles) or move to an adjacent tile if permitted by their tile attribute list, in the direction they are facing. Characters navigate in the world in terms of compass navigation. The 45° angles that a character can be facing relate to the compass directions north, south, east, west, northeast, southeast, southwest and northwest. In performing a check whether the tile that a character wishes to move to is free (i.e. there is no world object on top of it) collision detection mechanisms are put in place. Characters should also be intelligent in that they know the location of the edge of the world.

3.6.1 Character vision and navigation

In the simulation of an iTiles world, the characters in the world 'come to life' through their movements. The world forces described in Section 3.5 govern the movement of characters. Positive and negative forces are used for calculating a favourable direction of movement for characters. These forces of characters need to be taken into account at every world beat as the force strength of a character's positive force can change. A character's behaviour is also dependent on the position it is located in the world and the world objects and tiles it can see. For calculating the most favourable direction to move, characters have 'sight' in terms of **character vision** and a **navigation pointer** to guide them in the right direction.

A character's **navigation pointer** is used in order to determine the best favourable direction to move. This navigation pointer is calculated when a character looks around and determines it's relation to the tiles and the world objects on those tiles that it can see. During the process of character vision a character determines if it has a positive or negative force towards the component it can see, and if so, the force strength of the particular force is added in the direction of the component. Negative force strengths are added in the opposite direction to which the component is located in relation to the character. Weighting these force strengths determines the most favourable direction for the navigation pointer, indicating the direction the character will move next.

The process of **character vision** involves characters looking around and discovering the components specified in their positive, negative and movement forces in order to calculate the best favourable direction to move for a character's navigation pointer. Character vision is analogous to the vision of animals in the real world. Some characters may have a wide and short depth of vision such as a herbivore, or a far and long depth of vision such as a carnivore. The **vision type** of a character specifies how a character sees their world. It takes into account the direction the character is facing and other directions in relation to the direction the character is facing, in order to identify what tiles the character can see. **Vision depth** is a specification on how far a character can see. This is the amount of tiles a character can see in a particular direction defined by its vision type and is expressed in terms of tile units. Vision types such as carnivore vision and herbivore vision can be defined. For example, Figure 21 presents a character facing in the north direction, with a herbivore vision type and a vision depth of two tile units. A character's vision type does not necessarily depict the ecological behaviour of that character, but merely the way the character see an iTiles world.

Character vision takes into account both the vision type and the distance a character can see. However when calculating what tiles a character can see it needs to be determined whether a tile is located either diagonally, horizontally or vertically in relation to the position of the character, as the distance covered from the character to the tile may not entirely satisfy the vision depth specified. This is because the distance from a tile's edge to the same parallel edge is shorter than the distance covered between two opposite corners. The iTiles approach uses a Pythagoras calculation to check whether a character can see a tile that is diagonally located in relation to a character's position. This is illustrated in Figure 21, where the tiles that are diagonally located in relation to the character have been marked as not visible by that character.

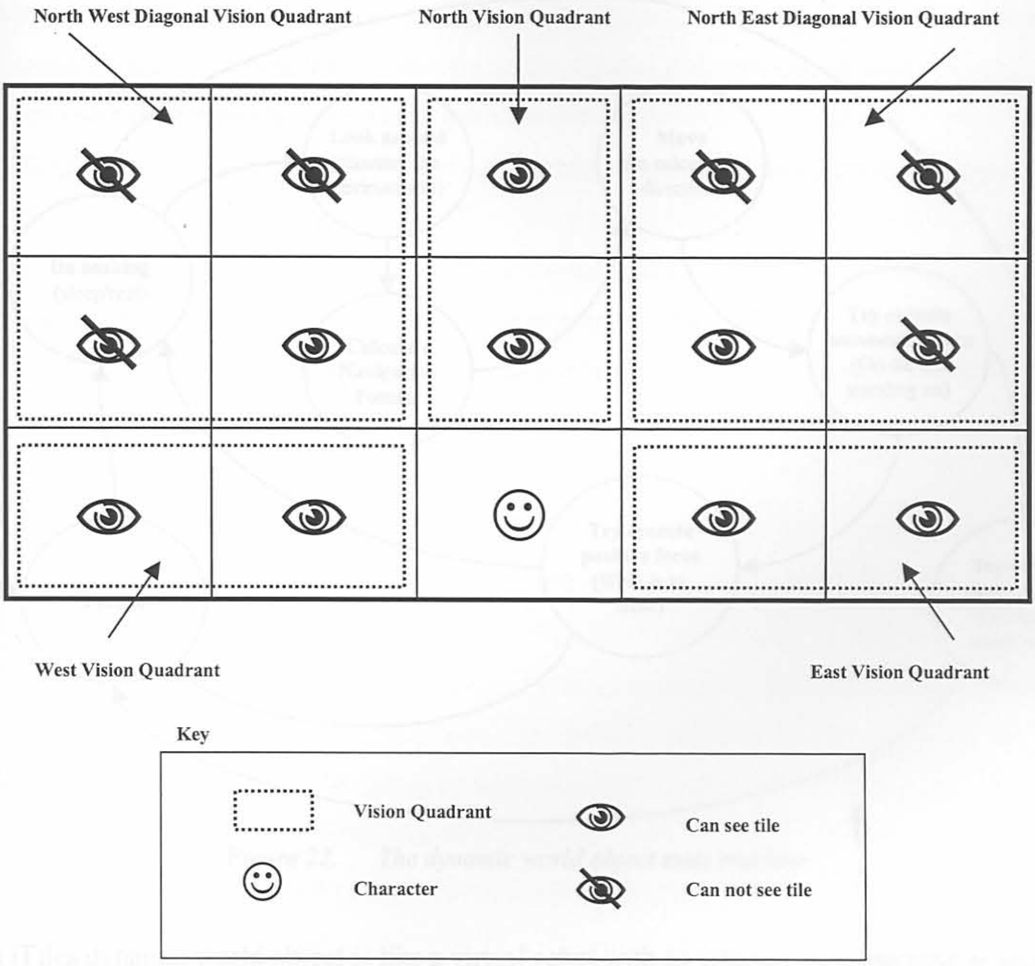

Figure 21. *A character facing north, with herbivore vision type with vision depth 2*

3.6.2 The dynamic world object state machine

A state machine is described as any device that stores the status of something at a given time and can operate on input to change the status and/or cause an action or output to take place for any given change. In summary a state machine has an initial state or record of something stored, a set of possible input events, a set of new states that may result from the input and lastly a set of possible actions or output events that result from a new state. [3, 49]

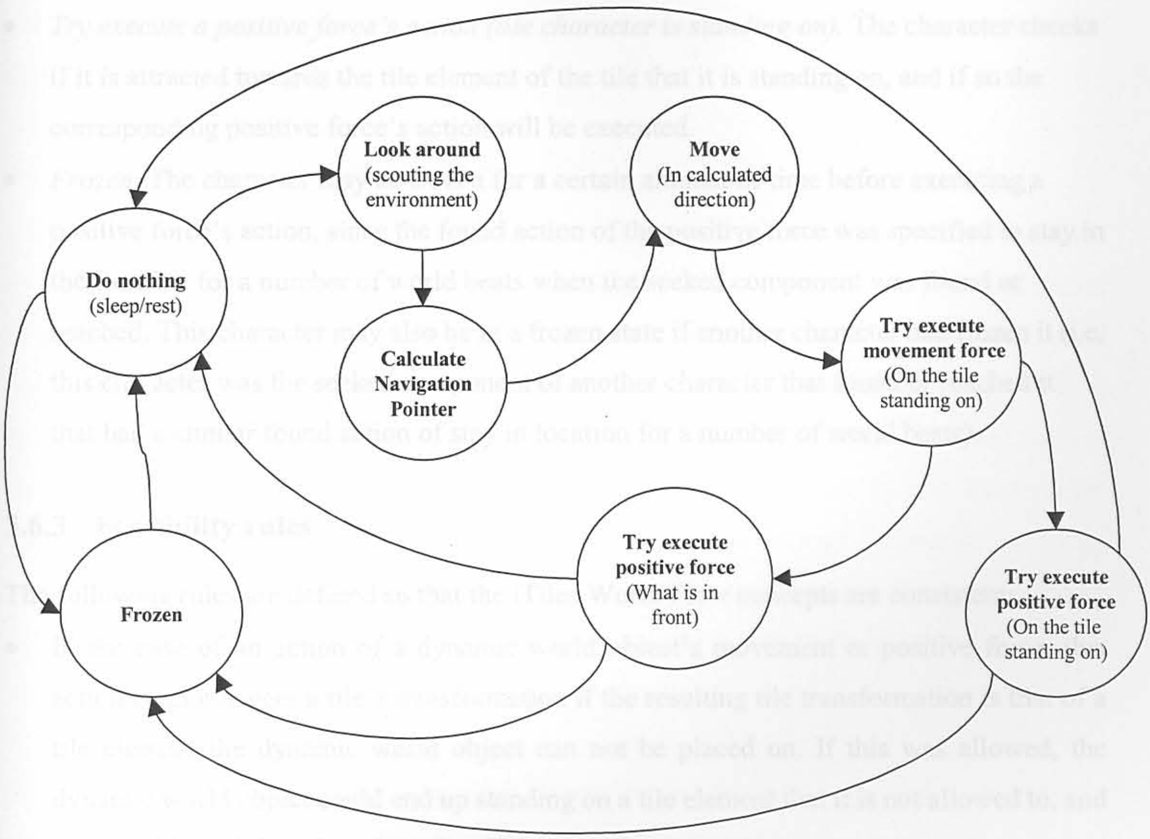

Figure 22. *The dynamic world object state machine*

An iTiles dynamic world object is like a virtual robot with an internal state machine as shown in Figure 22. In the simulation of an iTiles world, a dynamic world object can be in either of the following states:

- *Do nothing*. The character is paused for the remainder of the world beat.
- *Look around.* The character scouts the environment and determines what it can see in the environment in terms of the character's vision type and depth.
- *Calculate navigation pointer*. The character determines it's relationship with the components it can see and determines the most favourable direction to move for the navigation pointer.
- *Move.* The character moves from tile to tile, or changes the direction it is currently facing as indicated by the navigation pointer.
- *Try execute a movement force's action.* The character checks if it has a movement force for the tile element type of tile it is standing on, and if so, the movement force's action will be executed.
- *Try execute a positive force's action (tile in front of character).* The character checks if it is attracted towards a component in front of it (either the tile or the world object that is standing on that tile), and if so, the corresponding positive force's action will be executed.

- *Try execute a positive force's action (tile character is standing on).* The character checks if it is attracted towards the tile element of the tile that it is standing on, and if so the corresponding positive force's action will be executed.
- *Frozen.* The character may be frozen for a certain amount of time before executing a positive force's action, since the found action of the positive force was specified to stay in the location for a number of world beats when the seeked component was found or reached. This character may also be in a frozen state if another character had frozen it (i.e. this character was the seeked component of another character that found or reached it that had a similar found action of stay in location for a number of world beats).

3.6.3 Feasibility rules

The following rules are defined so that the iTiles World Flow concepts are consistent:

- In the case of an action of a dynamic world object's movement or positive force, this action cannot trigger a tile's transformation if the resulting tile transformation is that of a tile element the dynamic world object can not be placed on. If this was allowed, the dynamic world object could end up standing on a tile element that it is not allowed to, and this would invalidate the rules of an iTiles world.
- A dynamic world object cannot freeze another dynamic world object that is already frozen (i.e. if the seeked component of a dynamic world object's positive force is a dynamic world object, the seeked dynamic world object will only be considered found for the first dynamic world object that finds it). If this were allowed situations of deadlock could arise, since a component may be seeked by more than one character.
- A dynamic world object cannot have a negative force towards a tile element appearing in its tile attribute list.
- A dynamic world object can either be attracted to a specific component or repelled by it. In other words either a positive force or a negative force can be specified for a certain component.
- Movement forces for dynamic world objects can only be specified for tile elements in the dynamic world object's attribute list.

3.7 Learning with an iTiles virtual laboratory

The simulation of an iTiles world is presented to young learners in computer-aided education. The simulation is the fun part of an iTiles virtual laboratory, experienced by young learners to enjoy. The iTiles framework provides several ways of experiencing the simulation, ranging from "seeing through a character's eyes" to taking control of a character. In this section advantages of using the iTiles World Flow concepts are discussed, presenting examples of the use of such concepts for specifying ecological relationships. Educational advantages of the iTiles framework are also presented.

3.7.1 Learning through interaction

Interactivity during the simulation forms the basis of the pedagogy and the learning approach of the iTiles framework. A user can learn by doing and learn by experiencing the simulation of an iTiles world. The iTiles framework allows the user (learner) to select different world objects inhabiting an iTiles world. When a world object is selected the camera is focussed on that selected world object. If the selected world object is a dynamic world object, selection of that dynamic world object results in camera tracking and the learner can observe the character's behaviour and "follow" the character as it moves through the world. An added advantage of selecting a dynamic world object is that a learner may take control of that dynamic world object, and is given the ability to control the movement of that character in the simulation.

The 3D virtual environment can be rotated, zoomed in and out and the learner can experience the same ecosystem from different viewpoints in terms of perspective. Virtual reality techniques can also be used to present a first or third person view from and of the selected character. With a third person view, learners will be able to follow the movements of a character from a distance, whether they are currently observing passively or taking control of the character. In first person view, a learner can experience the world from the world object's point of view, as if looking through the character's eyes, viewing the world in the direction the world object is facing. The iTiles approach includes a mode in which only the tiles visible by the selected character can be displayed in the simulation. This approach is called **inner vision** and takes into account the vision type and depth of the selected character. When this inner vision mode is activated only the tiles and world objects visible to the selected character are displayed, as illustrated in Figure 21.

The novel perspective of oneself experience and shaping a natural phenomenon, instead of acting as a passive observer, is intrinsically motivating [33]. By taking control of a character and experiencing the different perspectives of an iTiles world, this is realised. A third person view of a world object can be considered a perspective of the virtual world from an exocentric frame of reference, and the first person view a perspective of the virtual world from an egocentric frame of reference. By enabling learners to explore the world from different perspectives, the bicentric frame of reference [34] adopted here helps learners increase knowledge and gain more information of the virtual world (See Section 2.4 –Virtual Reality).

In [35] an approach called **virtual identity** is presented. A virtual identity is defined by knowledge about itself, its perception of the environment, its virtual embodiment and the physical simulation and kinematics of the virtual embodiment. Virtual identities and their virtual existence relate to their interaction with the environment, their reaction to the environment, their cognition of the environment and their emotional reactions. Taking on the role of, or selecting a world object in an iTiles simulation, can be expressed as a learner taking on a virtual identity. The first person view and character vision concepts further exemplify the concept of taking on a virtual identity.

3.7.2 Learning with world transformations

World transformations are visual or auditory changes that can occur in the simulation of an iTiles world, and are associated with the tile and world object components. World transformations of the gradual type can represent changes that require more time or exposure for the change to occur. This can often happen in nature, since for example, the growth of animal takes a considerable amount of time.

The concept of tile transformations presents a method to change to the terrain of an environment during a simulation. Depending on the tile element type, altering this tile element type with a tile transformation can represent environmental occurrences. For example, transforming grass tiles to sand tiles can depict erosion and overgrazing, and transforming water tiles to sand tiles can depict evaporation and depletion of water in times of drought. Other concepts related to animal growth, death and the depletion of resources (e.g. plants), can be denoted by the different types of world object transformations:

- A world objects growth can be represented by scaling a world object up.
- The depletion of a resource can be represented by scaling a world object down. For examples, plants could get smaller as they are eaten.
- A world object's death can be represented by a world object disappearing.

3.7.3 Learning with world forces

World forces represent the relationships of characters to one another and to other components of the environment. As in real life, forces of attraction and repulsion are present, which are similarly depicted by the iTiles World Flow concept of world forces. This enables for natural ecosystem relationships and behaviour to be portrayed from the relationships defined within world forces.

Different types of ecosystem relationships and behaviour can be described with the specified seeked component of a character's positive force:

- *Static world object*. A herbivore character can have a positive force towards a static world object such as a plant. When the plant is found by that character, the character's positive force's action can be set to trigger the plant's world object transformation to scale the plant down. The positive force's action can also be set to trigger the character's world object transformation to scale the character up. Triggering these world object transformations to occur can depict the characters growth and the plant's depletion.
- *Tile element*. Herbivore characters such as sheep or cows can, for example, have a positive force towards a grass tile element. When the sheep or cow characters find grass tile elements their behaviour can be depicted as grazing. Thirst can be represented by a character's positive force towards a water tile element. When the character finds the water tile element, the behaviour can be depicted as drinking.
- *Dynamic world object*. A predator-prey relationship or a mutual relationship can be defined by a character's positive force towards another character. The transformation triggers that are specified in the character's positive force's action can depict the type of relationship defined. For example, a mutual relationship can be depicted if no transformation triggers are specified for the action, which could illustrate, for example, the characters' need to converse with each other. However, if world transformations of those characters were triggered in the character's positive force's action (consider for example that the seeked character's world object transformation to disappear was triggered), this could illustrate one character's desire to eat the other character if it were a carnivore.

A character's positive force need not represent an ecological relationship. For example, a character may have an attraction towards a static world object such as a tree, however the relationship is neutral in that no transformations occur when the character finds the tree. In this case a bird chirping sound clip is specified as the character's positive force's found sound. This sound clip is played when the character finds the tree, which makes for a more enjoyable simulation.

Negative forces, which represent repulsion of characters with other components of the environment, can also help learners in understanding ecosystem relationships. For, example, by being afraid of a certain character, that scared character could represent the prey in a predator-prey ecosystem relationship. Also, a character's inability to swim can be depicted by being afraid of a tile element such as water.

The sound clips specified for positive forces and negative forces can also help establish ecosystem relationships and behaviour. For example, consider a leopard and baboon character in a predator-prey relationship. The leopard character has a positive force towards a baboon character, and the baboon character has a negative force towards the leopard character. The leopard's positive force's sighted sound is specified as a growl sound clip. The baboon's negative force's sighted sound is specified as a danger/warning cry sound clip. Therefore when the leopard sees the baboon it make a growling sound, and when the baboon sees the leopard it will alert the baboon troop of danger by making a warning cry sound.

Movement forces represent the relationship of the movement of characters on tiles. Environmental effects such as erosion can be represented by a character's movement force's action triggering transformations of the tile elements the character is moving on. With a gradual tile transformation, a certain character may influence the tile transformation's meter more than another. This can depict the weight of characters, as heavier characters would cause more erosion (i.e. weight could be directly proportional to the meter alteration percentage specified).

Many more ecosystem relationships are possible to be created using the iTiles World Flow concepts of world transformations and world forces. It is the task and imagination of the author of an iTiles world to provide these relationships. Education advantages of the iTiles framework are presented next.

3.7.4 Educational advantages of the iTiles framework

As exemplified in the previous sections an iTiles virtual laboratory can be used in teaching ecological biological processes involved in the field of earth science. In this section teaching with an iTiles virtual laboratory and the pedagogies of an iTiles virtual laboratory are discussed.

An iTiles virtual laboratory produced using the iTiles framework provides teachers a tool to aid in teaching. It provides a teacher with the following:

- Ability to present lessons on various topics in the field of earth science.
- Control of lesson contents through authoring an iTiles world.
  - Creativity in an iTiles world creation/lesson contents.

- In specifying an iTiles World Flow for an iTiles world, the iTiles World Flow concepts can be used to teach ecological relationships and behaviour in the simulation of that iTiles world.

Interactive simulations offer a fun and effective way to enable students to learn by doing and learn by experiencing. By using computer-based simulations, we can broaden the range of things students can learn. The iTiles framework has addressed what young learners can do with, and what they learn from a simulation of an iTiles world. Young learners can improve and learn fundamental cognitive skills such as mathematics and communication by:

- *Observation.* Learners can deduce the rules of an environment through observation of the simulation. They experience and observe the behaviour exhibited by characters through these characters' actions, movements, transformations and sounds they make.
- *Understanding.* Children learn to understand the interactions and relationships of characters within a world.
- *Exploration.* The simulation presents an opportunity for learners to explore. Children of different abilities can explore an iTiles world through the interaction mechanisms provided.
- *Communication.* A similar goal is likely to stimulate interaction between learners. This interaction improves communication skills as children can learn about an ecosystem together.
- *Authoring a virtual environment.* More advanced students could author their own worlds.

Pedagogy can also be embedded when presenting an iTiles world simulation to students. Elementary school science is about asking questions, collecting data that bear on those questions, and building support for answers [30].

Students can learn the following early skills in mathematics:

- *Matching.* Understanding the inhabitants of a world, and relating them to world objects of a similar type.
- *Counting.* Students could be given data collection tasks. For example, they could count the number of world inhabitants of a certain type.
- *Classification.* Students could be given the task of classifying different types of world objects according to their ecological relationships.
- *Ordering.*
- *Making patterns.*

3.8 Summary

In this chapter the Intelligent Tiles framework was introduced and the aims in terms of addressing the major problems of existing virtual laboratories were presented. The composition of a virtual environment produced with the iTiles framework was then discussed, followed by how one could simulate behaviour in such a world. How students can learn through the simulation of an iTiles world was presented, followed by the educational advantages of the iTiles framework. The next chapter presents an implementation of the iTiles framework using the concepts described in this chapter.