Towards Intelligent Semi-automated Game Generation in SimSYS: A Component-based Approach

**UTDCS-0?-13**

**The University of Texas at Dallas**

**MU-CTL-0?-13**

**Marquette University**

**August 2013**

**Kaleb Breault, Marquette University**

**Ricardo Daconceicao, The University of Texas at Dallas**

**Carsten Locke, The University of Texas at Dallas**

**Chris Mojica, The University of Texas at Dallas**

**Kendra M. L. Cooper, The University of Texas at Dallas**

**C. Shaun Longstreet, Marquette University**

# Abstract

The use of serious educational games has many advantages, offering immersive, engaging and fun environments that require deep thinking and complex problem solving within a construct of overcoming obstacles and challenges. Developing new games, however, to support broad and rapidly evolving disciplines has remained time consuming, expensive, and requiring the expertise of game designers, software developers, software engineering educators, and players. Here, an intelligent semi-automated component-based engineering approach for generating serious educational games is presented, which enables educators to rapidly and independently develop their own games across diverse educational topics. This is accomplished with an XML (Extensible Markup Language) based component structure and the Analytic Hierarchy Process (AHP) multi-criteria decision making algorithm. Each game is composed out of six “layers” or components; repositories of each individual component are searched to create new diverse games. Metadata XML files are included in these repositories and are the inputs to the AHP algorithm along with data from the user via a user interface wizard. This approach allows for expansion of both the user input and the component libraries for even more diverse educational games.

# Table of Contents

Abstract ii

Table of Contents iii

List of Figures v

List of Tables vi

1 Introduction 1

2 Overview of the Intelligent Semi-automated Game Generation Approach 3

2.1 Interactive Wizard User Interface 3

2.2 Create Component-based Game Engine 4

2.3 Repository 5

3 Background: Analytic Hierarchy Process 7

3.1 Step 1. Establish the search criteria 7

3.2 Step 2. Establish the weights of the criteria 7

3.3 Step 3. Find the relative weighted score of each element 7

3.4 Step 4. Select the highest ranked element 8

4 Game Component Models 10

4.1 Character Game Component 11

4.2 Theme Game Component 11

4.3 Lesson Game Component 14

4.4 Challenge Game Component 15

4.5 Locale Game Component 18

4.6 Subject Game Component 20

5 Design and Implementation of the Tool Support 21

5.1 Interactive Wizard User Interface 21

5.2 Create Component-based Game Engine 21

5.2.1 Component Selection 21

5.2.2 Component Object Creation (Load) 24

5.2.3 Game Composition 24

5.2.4 Game Script Output 26

6 Validation 27

6.1 Test Case #1 27

6.1.1 Description 27

6.1.2 Input 27

6.1.3 Expected Output 27

6.1.4 Actual Output 29

6.1.5 Test Status 31

6.2 Test Case #2 31

6.2.1 Description 31

6.2.2 Input 32

6.2.3 Expected Output 32

6.2.4 Actual output 34

6.2.5 Test Status 36

6.3 Test Case #3 36

6.3.1 Description 36

6.3.2 Input 36

6.3.3 Expected Output 37

6.3.4 Actual Output 38

6.3.5 Test Status 40

6.4 Test Case #4 40

6.4.1 Description 40

6.4.2 Input 41

6.4.3 Expected Output 41

6.4.4 Actual Output 43

6.4.5 Test Status 45

7 Conclusions and Future work 46

References 48

Appendix A. Characters XML 49

Appendix B. Theme XML 50

Appendix C. Lesson XML 55

Appendix D. Challenge XML 56

Appendix E. Locale XML 59

Appendix F. Subject XML 76

# List of Figures

Figure 1: Overview of the Component-based Serious Educational Game Generation Approach 3

Figure 2 Component-based Game Model 10

Figure 3 Character Game Component Model 11

Figure 4 Theme Game Component Model 12

Figure 5 Lesson Game Component Model 14

Figure 6 Challenge Game Component Model 16

Figure 7 Locale Game Component Model 18

Figure 8 Subject Game Component Model 20

Figure 9: Interactive Wizard User Interface Example 21

Figure 10 Application of the AHP Algorithm in the Select Components Module 23

Figure 11: Test Case #1 Expected Output 29

Figure 12: Test Case #1 Actual Output 31

Figure 13: Test Case #2 Expected Output 34

Figure 14: Test Case #2 Actual Output 36

Figure 15: Test Case #3 Expected Output 38

Figure 17: Test Case #3 Actual Output 40

Figure 18: Test Case #4 Expected Output 43

Figure 19: Test Case #4 Actual Output 45

# List of Tables

[Table 1 Characters XML Example 11](#_Toc364415682)

[Table 2 Theme XML Example 12](#_Toc364415683)

[Table 3 Lesson XML Example 15](#_Toc364415684)

[Table 4 Challenge XML Example 16](#_Toc364415685)

[Table 5 Locale XML File Example 18](#_Toc364415686)

[Table 6 Subject XML Example 20](#_Toc364415687)

[Table 7 Select Game Components Algorithm 22](#_Toc364415688)

[Table 8 Create Game Component Objects Algorithm 24](#_Toc364415689)

[Table 9 Compose Game Components Algorithm 25](#_Toc364415690)

[Table 10 Output Game Algorithm 26](#_Toc364415691)

[Table 15 Characters XML Example 49](#_Toc364415692)

[Table 16: Theme.xml (Complete) 50](#_Toc364415693)

[Table 17 Lesson XML Example 55](#_Toc364415694)

[Table 18: Challenge.xml (Complete) 56](#_Toc364415695)

[Table 19: Locale.xml (Complete) 59](#_Toc364415696)

[Table 20 Subject XML Example 76](#_Toc364415697)

# Introduction

Educational infrastructures face significant challenges including the need to rapidly, widely, and cost effectively introduce new or revised course material; encourage the broad participation of students; support traditional and emerging “classrooms” (e-learning, distributed classrooms), and address changing student motivations and attitudes. The course material needs to address learning objectives, which span subject specific content and transferable skills, such as collaboration, critical thinking, creative thinking, problem solving, reasoning abilities, learning to learn, professionalism, and decision making.

Serious educational games (SEGs) have significant pedagogical potential as they provide immersive, engaging and fun environments that require deep thinking and complex problem solving within a construct of overcoming obstacles and challenges [4][7][13][15]. They create interactive student-centered environments rather than a passive content-centered classroom environment. This allows students to create a personalized learning experience, progressively incorporating new knowledge and scaffolding it into what they already know. Because each student is able to engage course-based material at his or her own pace; underprepared or at-risk students can focus on needed skills at their convenience. Feedback is frequent and immediate, thereby reinforcing mastery of fundamental skills required for advancing further into the game. Integrating games into curriculum creates a highly motivating learning environment; it draws on students’ sense of fantasy and amusement; it is self-directed, appealing to individual student’s curiosity; and it is a continuous challenge wherein any existing tasks or knowledge that appears incomplete, inconsistent or incorrect motivates a student to foster deeper levels of learning.

There are many potential benefits to adopting games into curricula, but where can they be found? As with other software applications, they can be acquired off-the-shelf (e.g.,[10]), by modifying an existing game (e.g., [9][12]), or by developing new ones. Games are complex applications; developing new ones has been time consuming, expensive, and has required substantial expertise from diverse stakeholders: game developers, software developers, educators, and players. Research in the game community to improve this situation with semi-automated game generation approaches is just beginning to receive attention.

The component-based software engineering community continues to invest substantial effort to support the timely, cost effective development of large-scale, complex systems . At a very high level, this community considers software development as a problem involving the selection and composition of re-usable, high quality components, like assembling “lego” blocks. The components have gone through a resource intensive development process to specify, design, implement, test, and document; they have comprehensive interface descriptions including a specification of their functional and non-functional capabilities. The non-functional description includes quality of service attributes (e.g., performance). The specification, selection, and composition of components have received considerable attention. For example, formal notations, which offer a means to specify components concisely and unambiguously, such as XML, fuzzy logic, first-order logic as well as architectural description languages and coordination languages have been explored.

A component-based software engineering approach for rapidly developing serious educational games across diverse educational topics does not appear to be available in the literature. Here, an intelligent semi-automated component-based serious educational game generation approach, ISEGCB, is presented in detail, with a focus on the game component models and intelligent selection. Preliminary results have been presented in [3]. Based on a meta-model presented in earlier results , six game component models are defined in the ISEGCB (theme, locale, subject, characters, lesson, and challenge), which cut across the game. The components are viewed as layers of a game; they have dependencies in order to generate a consistent game. For example, characters may appear throughout the game: across the game introduction, in a challenge midway through, and the wrap-up. The game components are specified in XML, a standard, formal notation that is considered straightforward to understand and load in tool support with established libraries.

Repositories are formed from collections of these components and a multi-criteria decision-analysis algorithm called the Analytic Hierarchy Process (AHP) is used to select one of each component to create a game. The AHP algorithm uses a system of weighted criteria along with intrinsic values on choices to find the choice with the best overall score (see Background: Analytic Hierarchy Process for more detail). As a requirement of the AHP algorithm, there is a separate set of metadata, in XML format, that is used to describe each of the components (see Game Components for more detail). Additionally, there are some inter-component dependencies that are addressed in the form of an input wizard; the input is processed to address these dependencies allowing a coherent game to be created using a small set of basic questions about how the final game should look (see Tool Support for more detail).

The structure of this report is as follows. Section 2: Overview of the Intelligent Semi-automated Game Generation Approach, Section 3: Background: Analytic Hierarchy Process, Section 4: Game Components, Section 5: Tool Support, Section 6: Validation, Section 7: Conclusions and Future work.

# Overview of the Intelligent Semi-automated Game Generation Approach

The ISEGCB approach is organized into three layers (Figure 1): interactive wizard interface; create components based serious educational game engine; and the repository.

**Repository (game assets, game components, game scripts)**

XML

Game Component Collections

Game Assets (image files, audio files, educational material)

XML

Game Scripts

**Create Component-based Game Engine**

Theme

Locale

Subject

Character

Lesson

Challenge

…

**Interactive Wizard User Interface**

**Game**

**Composition**

**Component Selection**

<?xml version="1.0" encoding="UTF-8" standalone="true"?>

<!-- -->

<!-- Top Level Game -->

<!-- Test Game 1 contains 1 act, 1 scene and 2 screens -->

<!-- -->

[<game>](file:///C:\Users\GameSpecifications\Test%20Game%20descriptions\Test%20Game%201\TestGame_1_XML.xml)

<!-- -->

<!-- The game contains a default player who is rewarded with 5000 points after playing the game -->

<!-- -->

[<gameElement xsi:type="**PLAYER**" xmlns:xsi="**http://www.w3.org/2001/XMLSchema-instance**">](file:///C:\Users\GameSpecifications\Test%20Game%20descriptions\Test%20Game%201\TestGame_1_XML.xml)<location>OSR</location><size>LARGE</size>

<!-- -->

<!-- The profile of the player starts here -->

<!-- -->

[<profile>](file:///C:\Users\GameSpecifications\Test%20Game%20descriptions\Test%20Game%201\TestGame_1_XML.xml)<name> Default Player </name><type> Protagonist </type><skills> Software Engineering, Project Management, Configuration Management </skills><experience> 0 </experience><communication> Good </communication><leadership> Good </leadership><teamwork> Good </teamwork><demographics> Male, Caucasian </demographics><degrees> NONE </degrees></profile>



*Components*

*+ metadata*

Wizard support

* Create new game
* Preview (graphics, audio)
* Edit/Save/Load

<?xml version="1.0"?>

[<Challenge><LessonChallenges><LessonChallenge>](file:///C:\Users\IntSemi-automatedGameGenerationComponent\Samples\Sample%20Game%20XMLs\Challenge.xml)<Buttons> </Buttons>[<ChallengeOptions><ChallengeOption>](file:///C:\Users\IntSemi-automatedGameGenerationComponent\Samples\Sample%20Game%20XMLs\Challenge.xml)<ChallengeOptionType>BUTTON</ChallengeOptionType><ButtonLocationType>CHALLENGE\_1</ButtonLocationType>[<Reward>](file:///C:\Users\IntSemi-automatedGameGenerationComponent\Samples\Sample%20Game%20XMLs\Challenge.xml)<Points>-2147483646</Points></Reward><Text>2</Text><Timer/><TransitionType>ADDITIONAL</TransitionType>[<AdditionalScreens><AdditionalScreen xsi:type="**failureScreen**" xmlns:xsi="**http://www.w3.org/2001/XMLSchema-instance**"><Buttons><Button>](file:///C:\Users\IntSemi-automatedGameGenerationComponent\Samples\Sample%20Game%20XMLs\Challenge.xml)<ButtonLocationType>NEXT</ButtonLocationType><Text>Retry</Text><Timer/><TransitionType>CURRENT\_CHALLENGE</TransitionType></Button></Buttons>[<LOCharacters><Character>](file:///C:\Users\IntSemi-automatedGameGenerationComponent\Samples\Sample%20Game%20XMLs\Challenge.xml)<CharacterType>VILLIAN</CharacterType><Timer/></Character>[<Character>](file:///C:\Users\IntSemi-automatedGameGenerationComponent\Samples\Sample%20Game%20XMLs\Challenge.xml)<CharacterType>HERO</CharacterType><Timer/></Character></LOCharacters>

<?xml version="1.0"?>

[<Challenge><LessonChallenges><LessonChallenge>](file:///C:\Users\IntSemi-automatedGameGenerationComponent\Samples\Sample%20Game%20XMLs\Challenge.xml)<Buttons> </Buttons>[<ChallengeOptions><ChallengeOption>](file:///C:\Users\IntSemi-automatedGameGenerationComponent\Samples\Sample%20Game%20XMLs\Challenge.xml)<ChallengeOptionType>BUTTON</ChallengeOptionType><ButtonLocationType>CHALLENGE\_1</ButtonLocationType>[<Reward>](file:///C:\Users\IntSemi-automatedGameGenerationComponent\Samples\Sample%20Game%20XMLs\Challenge.xml)<Points>-2147483646</Points></Reward><Text>2</Text><Timer/><TransitionType>ADDITIONAL</TransitionType>[<AdditionalScreens><AdditionalScreen xsi:type="**failureScreen**" xmlns:xsi="**http://www.w3.org/2001/XMLSchema-instance**"><Buttons><Button>](file:///C:\Users\IntSemi-automatedGameGenerationComponent\Samples\Sample%20Game%20XMLs\Challenge.xml)<ButtonLocationType>NEXT</ButtonLocationType><Text>Retry</Text><Timer/><TransitionType>CURRENT\_CHALLENGE</TransitionType></Button></Buttons>[<LOCharacters><Character>](file:///C:\Users\IntSemi-automatedGameGenerationComponent\Samples\Sample%20Game%20XMLs\Challenge.xml)<CharacterType>VILLIAN</CharacterType><Timer/></Character>[<Character>](file:///C:\Users\IntSemi-automatedGameGenerationComponent\Samples\Sample%20Game%20XMLs\Challenge.xml)<CharacterType>HERO</CharacterType><Timer/></Character></LOCharacters>

<?xml version="1.0"?>

[<Challenge><LessonChallenges><LessonChallenge>](file:///C:\Users\IntSemi-automatedGameGenerationComponent\Samples\Sample%20Game%20XMLs\Challenge.xml)<Buttons> </Buttons>[<ChallengeOptions><ChallengeOption>](file:///C:\Users\IntSemi-automatedGameGenerationComponent\Samples\Sample%20Game%20XMLs\Challenge.xml)<ChallengeOptionType>BUTTON</ChallengeOptionType><ButtonLocationType>CHALLENGE\_1</ButtonLocationType>[<Reward>](file:///C:\Users\IntSemi-automatedGameGenerationComponent\Samples\Sample%20Game%20XMLs\Challenge.xml)<Points>-2147483646</Points></Reward><Text>2</Text><Timer/><TransitionType>ADDITIONAL</TransitionType>[<AdditionalScreens><AdditionalScreen xsi:type="**failureScreen**" xmlns:xsi="**http://www.w3.org/2001/XMLSchema-instance**"><Buttons><Button>](file:///C:\Users\IntSemi-automatedGameGenerationComponent\Samples\Sample%20Game%20XMLs\Challenge.xml)<ButtonLocationType>NEXT</ButtonLocationType><Text>Retry</Text><Timer/><TransitionType>CURRENT\_CHALLENGE</TransitionType></Button></Buttons>[<LOCharacters><Character>](file:///C:\Users\IntSemi-automatedGameGenerationComponent\Samples\Sample%20Game%20XMLs\Challenge.xml)<CharacterType>VILLIAN</CharacterType><Timer/></Character>[<Character>](file:///C:\Users\IntSemi-automatedGameGenerationComponent\Samples\Sample%20Game%20XMLs\Challenge.xml)<CharacterType>HERO</CharacterType><Timer/></Character></LOCharacters>

…



*Existing*

*Standards,*

*Course material*



**Game Script Output**

**Component**

**Object**

**Creation**

Figure 1: Overview of the Component-based Serious Educational Game Generation Approach

## Interactive Wizard User Interface

The User Interface provides wizards to acquire information from the game developer; preview the game generated; and edit the game generated as needed. The information acquired from the game developer is used to characterize and create a useful game. The information includes: Intended Grade Level, Player Gender, Character Age, Character Dress, Game Theme, Game Subject, Game Setting and Challenge Difficulty. For example, a teacher for grade 4 students in WI, U.S.A. may need a short game to use in his class as an early part of the lesson. The teacher may characterize the game needed as follows:

* Intended Grade Level: Primary School
* Player Gender: Male
* Character Age: Young
* Character Dress: No Preference
* Game Theme: Dream
* Game Subject: Math
* Game Setting: Non-Terrestrial
* Challenge Difficulty: Easy

The wizard provides the options the game developer can select from; the current prototype uses a list radio buttons. For example, the user can choose from subject areas (English, Math, Science, Social Studies, Literature, Professional, No Preference), game theme (Gooble, Dream, Virtual Tour, Workplace, Other, No Preference), etc. A more in-depth description of the selection can be seen in Section 5.1 Selecting Game Components and Figure 10. As additional repositories of subject area assets are added to the repository, the options can be readily updated. In the future it will be interesting to explore alternative interface techniques, natural language processing techniques for speech or text input.

Using the information acquired, the game developer can create the game; the Create Component-based Serious Educational Game Engine is used to accomplish this. The teacher can preview the generated game, edit it if desired, and save the game. The preview and edit features are currently handled by a separate viewing tool. These features will be added to the Interactive Wizard User Interface in the future.

## Create Component-based Game Engine

The Create Component-based Serious Educational Game Engine is a pipeline of four modules: component selection; component object creation; game composition; and game script output. Currently, the search, ranking, and selection are done manually for the component selection; the object creation, composition, and game script output modules have been prototyped.

The Component Selection module uses the characterization information provided by the game developer to search, rank, and select a collection of game components that are used to assemble a game. It does so by implementing an AHP algorithm to weigh the possible choices with the characterization information. The AHP is used to find a component that fits the search criteria as best as possible given restrictions on scope of the Repository. The game components are retained in the Repository; there are collections of theme, locale, subject, character, lesson, and challenge components specified in XML. Metadata about the collections are used in the search; these are also represented in XML. The output of this module is a collection of the selected game components’ names (e.g., file names); this collection is used by the Component Creation module.

The Component Creation module uses the collection of component names provided to retrieve the components from the Repository and convert them into a collection of working objects. The component creation is in a separate module in order to encapsulate the capability to load the XML files from the Repository. The output of this module is a collection of working objects; this collection is used by the Game Composition module.

The Game Composition module assembles a game out of the components by using an overall game structure as a wireframe. The games are organized as a sequence of Acts (intro act, internal acts, outro act); the acts have screens. For example, an internal (learning) act has screens to introduce, present, and wrap-up lessons and challenges. The acts are built in order: Intro Act, Learning Act(s), and then the Outre Act. Once all the acts are created, they are added to the game’s wireframe.

Once the game is built it can be passed to the game script output module, which exports the game into the desired format. Currently, the game is output as an XML file in the SimSYS game specification format.

## Repository

The repository contains the re-usable game components, game assets, and game scripts. The game components, described in more detail in Game Components, are: challenge, characters, lesson, locale, subject, and theme. The challenge entity represents the challenge presented at the end of the lesson and is currently limited to a multiple choice quiz of varying length. The characters entity contains information about what names and assets should be used for each of the four supported character types (hero, villain, player, alternate). The lesson entity encompasses a single lesson which directs the student to learn some objective. The locale entity holds all information describing the location of the experience such as the background, character positions, background and foreground object positions and locations of where text can be displayed. This allows other entities to simply reference a named location, such as a character’s speech bubble, and provide the accompanying text. The subject entity describes the overall subject that is being taught and any introductory text. The theme is the story aspect of the game and contains not only the information used to build the intro and outro acts, but also story snippets, which surround each learning act to progress the story throughout the learning experience.

The game asset collection in the repository includes graphic images, audio files, and existing standards and course material. The graphic images are organized by characters, props, backdrops, and interactions; images are in the standard .png format, 200x300 pixels. For each character, there are 56 images to provide a variety of poses (standing, sitting, walking, talking, facing different directions (left, right, straight ahead). Images of props include furniture, easels, podiums, clocks, computers, phones, and so on. The backdrops provide a background image for a setting. For example, backdrop images for offices, classrooms, meeting rooms, medieval castles, forests, outer space, and so on are stored for re-use. The interaction images include information boxes, conversation bubbles, and buttons. Audio files include sound effects and music; these are stored in standard .wav or .mid files format. Existing standards and course material are also stored in the repository as a valuable reference material. Course material may be in presentation slides, lesson notes, course books, homework assignments, examinations; this ad-hoc collection may be stored in a wide variety of file formats.

The game script collection in the repository includes the games generated. Games can be created, played, and then improved (edited/saved) as needed. The games are organized by broad subject area (professional development, science, arts and humanities, medicine, business, and so on). A taxonomy for serious educational games will be explored to improve this categorization.

# Background: Analytic Hierarchy Process

The AHP is an established multi-criteria decision making approach proposed by Thomas Saaty [11]. It has been applied to a wide variety of decision-making problems, such as how best to reduce climate change [1], selecting university faculty [5], and decisions regarding the location of off-shore manufacturing plants [1]. The AHP is a 4-step process that is based on matrix algebra. The four steps are described below symbolically and with a running example on choosing a car to buy as a concrete example to help illustrate the steps. For each step, the inputs, processing (matrix manipulations), and outputs are described.

## Step 1. Establish the search criteria

The search criteria are // include definition here . The search criteria are found how?

The number of search criteria established the size of the input matrix, N xxx

Show symbolic input matrix here as an equation; IM = [] (1)

The criteria in the car selection in this example are defined by the stakeholders to be: safety, cost, seating/storage, and warranty. // describe symbolically first, then give concrete car example

Show car example input matrix here as an equation; IM = [] (2)

## Step 2. Establish the weights of the criteria

The searcher gives each of the attributes a “weight” in relation to each other; this is captured in an NxN matrix. The values of this matrix are how each horizontal component compares to the vertical component in the list (with diagonals of all 1). For example, xxx

Show symbolic weighted criteria matrix here - equation; xxxx (build up the NXN matrix) (3)

For instance, safety is twice as important as cost, and cost is a third as important as seating/storage, xxx.

Show car example matrices here as equations; yyy = [] (4)

## Step 3. Find the relative weighted score of each element

Xxx Then multiply this vector by each member of the search space. The approximation of the eigenvector of the input matrix is found and then multiplied by the search space matrix, which is an MxN matrix where N is the number of criteria and M is the size of the search space. The result gives us a one-dimensional array with the “score” of each component as the one value.

Then after this is established, the elements are compared to each other two at a time, assigning values to every element in relation to all the other elements. The intrinsic values of the choices are then given weight by these elements and the choice with the highest overall score is chosen. The contestants are then sorted into a list with the highest/ best choice at the top. For slight randomization, we randomly generate a number, if it is divisible by three take the highest weighted component, else continue onto the next component with a new random number. (The best choice is the default case if a number divisible by three is never generated).

Show symbolic matrices here as equations; xxxx (?)

A weighted score is then calculated using the “weights” of the attributes. For example, if a car scores a 7/10 in safety it would earn 70% of the possible points for that category.

Show car example matrices here as equations; xxx = [] (?)

## Step 4. Select the highest ranked element

xxxmember of the search space with the highest value in the remaining one-dimensional array.

Show symbolic matrices here as equations; xxxx (?)

Show car example matrices here as equations; xxx = [] (?)

Search Space Matrix Input Matrix Eigenvector of Input Matrix

Search Space Matrix \* Eigenvector of Input Matrix = Weighted Score Matrix

The Search Space Matrix is read in from the component’s metadata file, with each row being a different choice. The Input Matrix is the input from the user that specifies which of the criteria is weighted highest, this is the variable matrix. The eigenvector of this matrix is found and multiplied by the Search Space Matrix to get a Weighted Score Matrix, where the largest value is the best fit according to the inputs. In this example, the third (bottom) choice scores the highest and therefore is the best fit.

Matrix1 is the list of choices and their criteria score. Matrix 2 shows the relations of each criterion in respect to each other. In this example, criteria 1 is twice as important as criteria 2 and three times more important than criteria 3, additionally criteria 3 is four times as important as criteria 2. Given these weights, this makes Characters3 the best choice even though Characters1 scored better in the most important category.

# Game Component Models

The SimSYS games are composed of six different layers, or game components. The game components are the Character, Subject, Theme, Lesson, Challenge, and Locale. An overview of the game component models and their associations to create a game is presented in (Figure 2). The game component models (purpose, dependencies, and outputs) are discussed individually in the sections below. The game components models are presented as UML Components [10] and in XML [17], both widely adopted standards. The UML Component representation provides a semi-formal, concise visual model; the XML representation provides a formal model that can be readily manipulated (e.g., loaded) in prototype tool support. For the UML Component model the “lollipop” syntax for interfaces has been selected, as this option highlights the dependencies among components. A full circle represents the interface a component provides; a half circle represents an interface the component depends on; square boxes represent ports on an interface.

Challenge Component

Subject Component

Character Component

Lesson Component

Locale Component

Theme Component

Figure 2 Component-based Game Model

## Character Game Component

Character Metadata

setAvailableCharacters

Retrieve

Characters

Character

Core

*- Role*

*- Name*

Figure 3 Character Game Component Model

The character component is accessed by the theme component, lesson component, and the challenge component. It provides the characters that populate the game. As seen from the diagram, this component is called from other components by calling setAvailableCharacters.

An example of the character XML is presented in Table 1 below.

Table 1 Characters XML Example

|  |
| --- |
| <Characters>  <Alt>  <Directory>character\_1</Directory>  <Name>Chareeeesa</Name>  <Prefix>char1</Prefix>  </Alt>  <Hero>  <Directory>character\_19</Directory>  <Name>Sir Solvesalot</Name>  <Prefix>char19</Prefix>  </Hero>  <Villain>  <Directory>character\_22</Directory>  <Name>Calcutron</Name>  <Prefix>char22</Prefix>  </Villain>  <Player>  <Directory>character\_10</Directory>  <Name>Siva</Name>  <Prefix>char10</Prefix>  </Player>  </Characters> |

## Theme Game Component

Theme

 Theme Core:

Intro/Outro Screens

*- Select Generic Interaction Props*

*- Select Characters on Screen*

*- Create Textboxes*

*- Character Movement (in the works)*

Retrieve

 Theme Core:

Story Intro/Outro Screens

placeTextboxes

selectCharactersOnScreen

createTextboxes

setAvailableCharacters

setAvailableCharacters

placeSetDecorationProps

placeGenericInteractionProps

placeCharacters

placeBackdrop

Theme Metadata

Figure 4 Theme Game Component Model

The theme component is unique in that it has two different functions depending on what screen is being built. Figure 4 Theme Game Component shows that for the intro and outro screens this component receives input from the character component, setAvailableCharacters, and then builds the rest of the screen right here without any other dependencies. The component selects generic interaction props, select characters on screen, create textboxes, handles movement, places backdrop (placeBackdrop), places generic interaction props (placeGenericInteractionProps), places set decoration props (placeSetDecorationProps), places texboxes (placeTextboxes), and places characters (placeCharacters).

The story intro and outro screens are handled similarly to the lesson and challenge screens. This component receives the characters available from the characters component (setAvailableCharacters), selects which characters are used for the screen (selectCharactersOnScreen), and finally creates the textboxes that are used by the locale (createTextboxes).

A partial example of a theme component in XML is below in Table 2 (see Table 16 for the complete example).

Table 2 Theme XML Example

|  |
| --- |
| <Theme>  <IntroScreens>  <IntroScreen>  <Background>Backdrops\Home Office.png</Background>  <ThemeButtons>  <entry>  <key>NEXT</key>  <value>  <Height>50.566680908203125</Height>  <LocX>581.83331298828125</LocX>  <LocY>332.75</LocY>  <PathToAsset></PathToAsset>  <Text>Continue</Text>  <Width>157.86669921875</Width>  <behavior>  <behaviorType>TRANSITION\_BEHAVIOR</behaviorType>  <displayName>Transition Behavior</displayName>  <Transition>NEXT\_SCREEN</Transition>  <trigger>Click</trigger>  </behavior>  <name>Next</name>  </value>  </entry>  </ThemeButtons>  <GameObjects>  <GameObject>  <Height>78.316665649414062</Height>  <LocX>327.14999389648438</LocX>  <LocY>123.69999694824219</LocY>  <PathToAsset>Props\GenericInteraction\TextBubble.png</PathToAsset>  <Width>247.89999389648438</Width>  </GameObject>  </GameObjects>  <InformationBoxes>  <InformationBox>  <Height>59.20001220703125</Height>  <LocX>346.88330078125</LocX>  <LocY>128.63331604003906</LocY>  <Text>Today's lesson was difficult. I wonder if Gooble can help!</Text>  <Width>210.28340148925781</Width>  <name>Today's lesson was difficult. I wonder if Gooble can help!</name>  </InformationBox>  </InformationBoxes>  <ThemeCharacters>  <entry>  <key>PLAYER</key>  <value>  <Height>292.91667175292969</Height>  <LocX>209.36663818359375</LocX>  <LocY>130.48335266113281</LocY>  <Text></Text>  <Width>171.433349609375</Width>  <CharacterAssetType>WALK\_RIGHT\_BEHIND</CharacterAssetType>  <CharacterType>PLAYER</CharacterType>  </value>  </entry>  </ThemeCharacters>  </IntroScreen>  (…) –More IntroScreens  <OutroScreens>  <OutroScreen>  (…) –Mimics The Intro Screen  </OutroScreen>  </OutroScreens>  <ThemeStories>  <ThemeStory>  <StoryIntroScreens>  <StoryIntroScreen xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xsi:type=*"themeStoryScreenIntro"*>  <Buttons>  <Button>  <ButtonLocationType>NEXT</ButtonLocationType>  <Text>Continue</Text>  <Timer></Timer>  <TransitionType>NEXT\_SCREEN</TransitionType>  </Button>  </Buttons>  <LOCharacters>  <Character>  <CharacterType>PLAYER</CharacterType>  <MovementType></MovementType>  <Timer></Timer>  </Character>  (…) –Additional Characters  </LOCharacters>  <InformationBoxes>  <InformationBox>  <TextType>PLAYER</TextType>  <Text>Who are you?</Text>  <Timer></Timer>  </InformationBox>  (…) –Additional Information Boxes  </InformationBoxes>  </StoryIntroScreen>  </StoryIntroScreens>  <StoryOutroScreens>  <StoryOutroScreen xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xsi:type=*"themeStoryScreenOutro"*>  (…) –Mimics the StoryIntroScreen  </StoryOutroScreen>  </StoryOutroScreens>  </ThemeStory>  </ThemeStories>  </Theme> |

## Lesson Game Component

Lesson

Lesson

Metadata

Retrieve

Lesson

Core

*- Character Movement (in the works)*

*- Timer*

createTextboxes

selectCharactersOnScreen

getAvailableCharacters

getSubject

Figure 5 Lesson Game Component Model

The lesson component provides what characters and textboxes are placed on each screen (createTexrboxes, selectCharactersOnScreen). They are then placed on the screen by the locale component. This is also where a movement feature will be implemented in the future.

An example of a complete lesson in XML is presented in Table 3.

Table 3 Lesson XML Example

|  |
| --- |
| <Lesson>  <LessonScreens>  <LessonScreen>  <Buttons>  <Button>  <ButtonLocationType>NEXT</ButtonLocationType>  <Text>Continue</Text>  <Timer></Timer>  <TransitionType>NEXT\_SCREEN</TransitionType>  </Button>  </Buttons>  <LOCharacters>  <Character>  <CharacterType>HERO</CharacterType>  <Timer></Timer>  </Character>  <Character>  <CharacterType>PLAYER</CharacterType>  <Timer></Timer>  </Character>  </LOCharacters>  <InformationBoxes>  <InformationBox>  <Text>To perform the addition operator you must add together the operands. EX: 2+4=6</Text>  <TextType>ALT1</TextType>  <Timer></Timer>  </InformationBox>  </InformationBoxes>  </LessonScreen>  </LessonScreens>  </Lesson> |

## Challenge Game Component

Challenge

Challenge

Metadata

Retrieve

createTextboxes

selectCharactersOnScreen

Challenge

Core

*- Timer*

*- Reward*

*- Transition Type*

*- Character Movement (in the works)*

getAvailableCharacters

getSubject

Figure 6 Challenge Game Component Model

The challenge layer selects the characters (selectCharactersOnScreen) and provides the textboxes (createTextboxes) that are again placed on the challenge screens and failure screens (labeled AdditionalScreen in Table 4 Challenge XML Example) by the locale component. This is also where the timer and reward features are handled. Movement on all challenge screens are implemented here as well.

A partial example of a challenge in XML is in Table 4 (refer to Table 18 for the compete example).

Table 4 Challenge XML Example

|  |
| --- |
| <Challenge>  <LessonChallenges>  <LessonChallenge>  <Buttons>  </Buttons>  <ChallengeOptions>  <ChallengeOption>  <ChallengeOptionType>BUTTON</ChallengeOptionType>  <ButtonLocationType>CHALLENGE\_1</ButtonLocationType>  <Reward>  <Points>-2147483646</Points>  </Reward>  <Text>2</Text>  <Timer></Timer>  <TransitionType>ADDITIONAL</TransitionType>  <AdditionalScreens>  <AdditionalScreen xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xsi:type=*"failureScreen"*>  <Buttons>  <Button>  <ButtonLocationType>NEXT</ButtonLocationType>  <Text>Retry</Text>  <Timer></Timer>  <TransitionType>CURRENT\_CHALLENGE</TransitionType>  </Button>  </Buttons>  <LOCharacters>  <Character>  <CharacterType>VILLIAN</CharacterType>  <Timer></Timer>  </Character>  <Character>  <CharacterType>HERO</CharacterType>  <Timer></Timer>  </Character>  <Character>  <CharacterType>PLAYER</CharacterType>  <Timer></Timer>  </Character>  </LOCharacters>  <InformationBoxes>  <InformationBox>  <Text>You answered incorrectly!</Text>  <TextType>ALT1</TextType>  <Timer></Timer>  </InformationBox>  <InformationBox>  <Text>Don't worry! You can try again.</Text>  <TextType>HERO</TextType>  <Timer></Timer>  </InformationBox>  </InformationBoxes>  <ChallengeOptions />  <Timer></Timer>  </AdditionalScreen>  </AdditionalScreens>  <ChallengeOptionType>BUTTON</ChallengeOptionType>  </ChallengeOption>  <ChallengeOption>  (…)  </ChallengeOption>  </ChallengeOptions>  <LOCharacters>  <Character>  <CharacterType>HERO</CharacterType>  <MovementType></MovementType>  <Timer></Timer>  <Character>  <CharacterType>PLAYER</CharacterType>  <Timer></Timer>  </Character>  </Character>  </LOCharacters>  <InformationBoxes>  <InformationBox>  <Text>You can do it!!</Text>  <TextType>HERO</TextType>  <Timer></Timer>  </InformationBox>  <InformationBox>  <Text>Using Algebra solve:</Text>  <TextType>CHALLENGE\_QUESTION</TextType>  <Timer></Timer>  </InformationBox>  <InformationBox>  <Text>2 + 4</Text>  <TextType>CHALLENGE\_DESCRIPTION</TextType>  <Timer></Timer>  </InformationBox>  </InformationBoxes>  <Timer>60</Timer>  </LessonChallenge>  </LessonChallenges>  </Challenge> |

## Locale Game Component

Locale

Local

Metadata

Retrieve

Locale

Core

*- Select Generic Interaction Props*

*- Select Set Decoration Props*

placeBackdrop

placeGenericInteractionProps

placeSetDecorationProps

placeTextboxes

selectCharactersOnScreen

createTextboxes

placeCharacters

getSubject

Figure 7 Locale Game Component Model

The locale component provides the backbone for all the screens excluding the intro screen and outro screen. As mentioned before, both of those screens are created completely by the theme and character components. The locale places the backdrop (placeBackdrop), generic interaction props (placeGenericInteractionProps), decoration props (placeSetDecorationProps), and textboxes (placeTextboxes) on the screen at a specified location. The textboxes and characters are inputted (selectCharactersOnScreen, createTextboxes) for the designated screen. The interaction props and decoration props are created here as shown in the Locale Core.

A partial example of a Locale in XML is in Table 5 (refer to Table 19 for the complete example).

Table 5 Locale XML File Example

|  |
| --- |
| <Locale>  <LocaleScreens>  <entry>  <key>LESSON</key>  <value>  <Background>Backdrops\Medieval.png</Background>  <Buttons>  <entry>  <key>BACK</key>  <value>  <Height>1</Height>  <LocX>1</LocX>  <LocY>1</LocY>  <Width>1</Width>  </value>  </entry>  (…) –Additional Buttons  </Buttons>  <Characters>  <entry>  <key>PLAYER</key>  <value>  <Height>91.26666259765625</Height>  <LocX>135.98329162597656</LocX>  <LocY>313.01666259765625</LocY>  <Width>104.21666717529297</Width>  <CharacterAssetType>RIGHT\_POINT\_NO</CharacterAssetType>  <CharacterType>PLAYER</CharacterType>  <Movements>  <Movement>  <AnimationSequences>  <AnimationSequence>WALK\_RIGHT\_OPEN</AnimationSequence>  <AnimationSequence>WALK\_LEFT\_BEHIND</AnimationSequence>  </AnimationSequences>  <EndX>136</EndX>  <EndY>313</EndY>  <MovementType>WALK\_ONTO\_SCREEN</MovementType>  <Speed>1</Speed>  <StartX>1</StartX>  <StartY>313</StartY>  </Movement>  <Movement>  <AnimationSequences>  <AnimationSequence>WALK\_RIGHT\_BEHIND</AnimationSequence>  <AnimationSequence>WALK\_LEFT\_OPEN</AnimationSequence>  </AnimationSequences>  <EndX>1</EndX>  <EndY>313</EndY>  <MovementType>WALK\_OFF\_SCREEN</MovementType>  <Speed>1</Speed>  <StartX>136</StartX>  <StartY>313</StartY>  </Movement>  </Movements>  </value>  </entry>  (…) –Additional Characters  </Characters>  <GameObjects>  <!-- Player text box -->  <GameObject>  <Height>143.68331909179688</Height>  <LocX>36.083301544189453</LocX>  <LocY>194.00001525878906</LocY>  <PathToAsset>Props\GenericInteraction\Regular Speech 2.png</PathToAsset>  <Width>155.40000534057617</Width>  </GameObject>  (…) –Additional Game Objects  <InformationBoxes>  <entry>  <key>PLAYER</key>  <value>  <Height>46.866683959960938</Height>  <LocX>61.36663818359375</LocX>  <LocY>224.21665954589844</LocY>  <Width>84.48333740234375</Width>  </value>  </entry>  (…) –Additional Information Boxes  </InformationBoxes>  </value>  <key>CHALLENGE</key> (…) –Mimics the Lesson Format  <key>FAILURE</key> (…) –Mimics the Lesson Format  <key>LESSON\_STORY\_INTRO</key> (…) –Mimics the Lesson Format  <key>LESSON\_STORY\_OUTRO</key> (…) –Mimics the Lesson Format  </LocaleScreens>  </Locale> |

## Subject Game Component

Subject

Subject

Metadata

Retrieve

Subject

Core:

getSubject

Figure 8 Subject Game Component Model

The subject currently does not have any function. This component will be used in the future to reduce hardcoded titles in the intro and outro textboxes to allow more interchangeability.

A complete example of a subject in XML is presented in Table 6.

Table 6 Subject XML Example

|  |
| --- |
| <Subject>  <IntroText>Algebra</IntroText>  <Subject>Algebra Adventures</Subject>  </Subject> |

# Design and Implementation of the Tool Support

The architectural components presented in Section 2 (Interactive Wizard User Interface, Create Component-based Game Engine, and the Repository) have been realized in a prototype tool. In this section the design and implementation of the tool support is discussed.

## Interactive Wizard User Interface

The selection of the components is now done through a graphical user interface consisting of several basic questions on how the game should look. Currently, the GUI is in its beta stages, and therefore is not very complex or attractive. However, it is easily expandable, in both the options/answers to questions and the number of questions themselves. This is just the uppermost layer however, and the real selection of component pieces happens on a much lower level.

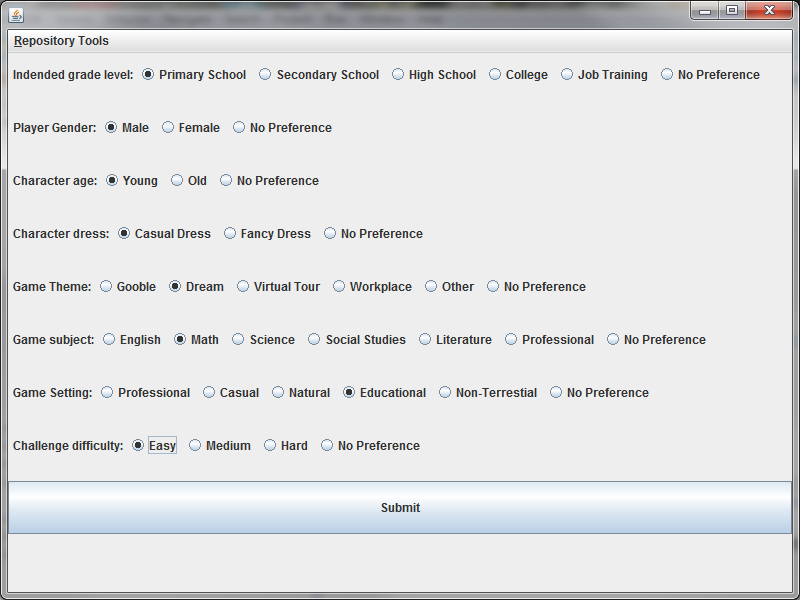
****

Figure 9: Interactive Wizard User Interface Example

## Create Component-based Game Engine

### Component Selection

The Component Selection module uses the input provided by the user (acquired with the Interactive Wizard User Interface module) to create an “Input Matrix” for each game component. These matrices can be loosely interpreted as block matrixes (a matrix composed of multiple sub-matrices), with each “block” consisting of one set of criteria, for instance, the Characters Input Matrix consists of three 2x2 blocks, one for gender age and dress with their diagonals aligned from top left to bottom right, making the Characters Input Matrix 6x6. Each matrix is analyzed and the eigenvector is extracted, this is a matrix with the same number of rows, however only one column.

The metadata for the game components, stored in XML format are loaded. The metadata consists of scores of the given component for each input criteria option. The data are stored in matrices, one for each component, called its Component Search Space. Each Search Space, as with the “Input Matrix,” has its own size with the number of rows equal to the number of eligible components, and number of columns equal to the number of criteria/metadata for that component. In the example, the Characters component, there are currently five different versions to choose from and six metadata criteria scores per choice, therefore the Characters Search Space is 5x6.

These two matrices, the eigenvector and the Search Space, are then multiplied together to make a new matrix with each of the eligible components to a row and only one column. The values in this one column are each of the components “scores.” The component with the highest score is the best suited to the search criteria.

To address the question of dependencies among the components, the GUI level of abstraction and analyzing of initial search criteria allows the program to take the inputs and reshape them to fit better with the available components to make coherent games. This option was chosen over limiting search space based on previous component choices, as this could potentially leave the user with no remaining search space.

Table 7 Select Game Components Algorithm

|  |
| --- |
| **Input:** The application opens a wizard which asks several multiple choice questions.  **Output:** A String array with the six component names.  Function searchAlgo  componentInputs[]: Matrix[6] ;  searchSpaces[]: SearchSpace[6];  componentInputSearchSpace: Matrix[6]  allFiles[]: String[6]  **for** x=0:searchSpace[]  searchSpaces[x]←new SearchSpace(gameComponents[x]);  componentInputSearchSpace[x]←new Matrix(searchSpace[x].getSearchSpace());  componentInputs[x]←newMatrix(new  double[searchSpaces[x].getNumberOfCriteria()][searchSpaces[x].getNumberOfCriteria()]);  weightedMatrix: Matrix  weightedMatrix← eigenvectorCalculation(componentInputs[x]); //returns the eigenvector of componentInputs[x]  criteriaScore:Matrix;  criteriaScore ← componentInputSearchSpace[x].times(weightedMatrix); // matrix multiplication method  allFiles[x] ← getLargestScore(criteriaScore,x); |

A sample test case from the AHP algorithm has many more criteria and choices to choose from. For instance here is an example from a test case searching for a relevant Subject XML file:

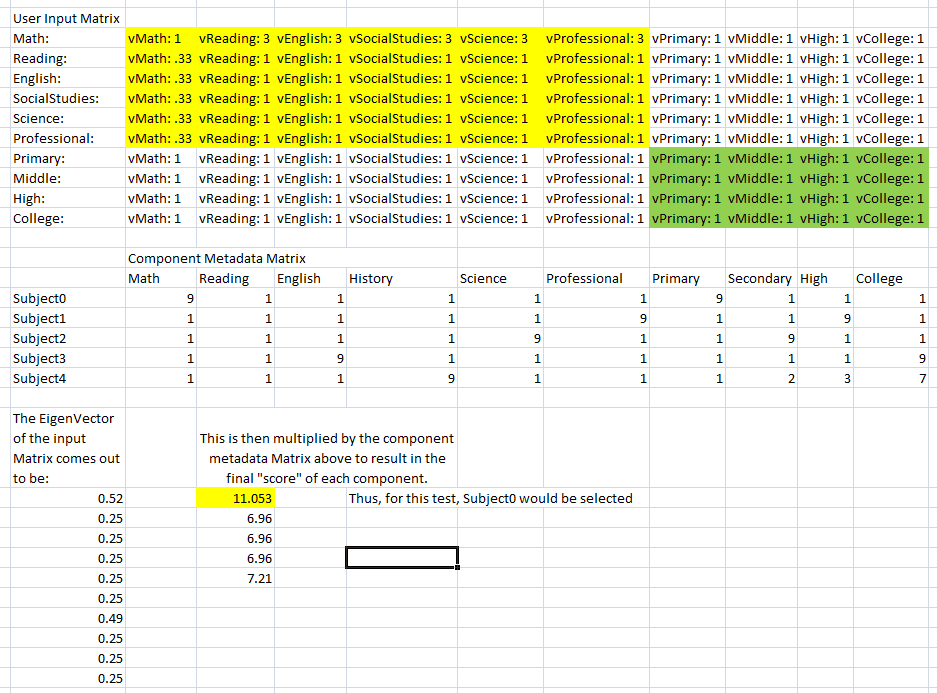


Figure 10 Application of the AHP Algorithm in the Select Components Module

The top matrix is an example of the inputs and the component’s weights while the middle matrix is the list of choices and their values for the given criteria.

Show the four steps and their matrices clearly in this section

Wizard swing library for wizard

Mit thing

### Component Object Creation (Load)

The Component Object Creation module is provided with the names of the components selected and creates the virtual java objects. The components are loaded from the XML representations using the JAXB library. All the new virtual java objects are stored as their corresponding component name in the Layers object. The Layers class also creates an empty Structure object that can be thought of as the skeleton of the final Game object. The algorithm for this module is presented in Table 8.

Table 8 Create Game Component Objects Algorithm

|  |
| --- |
| **Input:** A map of string to string representing the association of layers to their respective filenames in the repository.  **Output:** Layers, which is an object containing all layers or components used to build the game.  function loadXmlComponents  layers : Layers;  jaxbContext : JAXBContext;  file : File;  unmarshaller : Unmarshaller;  **for** layer ϵ layers – {lesson, challenge}  jaxbContext ← JAXBContext.newInstance(layer.class)  unmarshaller ← jaxbContext.createUnmarshaller();  file ← File(xmlFiles.layer);  layers.layer ← unmarshaller.unmarshal(file);  jaxbContext ← JAXBContext.newInstance(lesson.class)  unmarshaller ← jaxbContext.createUnmarshaller();  lessons : Lesson[];  **for** lessonFile ϵ xmlFiles.lessons  lesson : Lesson;  file ← File(lessonFile);  lesson ← unmarshaller.unmarshal(file);  lessons ← lessons ∪ {lesson};  jaxbContext ← JAXBContext.newInstance(challenge.class)  unmarshaller ← jaxbContext.createUnmarshaller();  challenges : Challenge[];  **for** challengeFile ϵ xmlFiles.challenges  challenge : Challenge;  file ← File(challengeFile);  challenge ← unmarshaller.unmarshal(file);  challenges ← challenges ∪ {challenge};  learningActs : LearningAct[];  **for** (lesson ϵ lessons) && (challenge ϵ challenges)  learningAct: LearningAct;  lessonActs : LessonAct[];  lessonAct : LessonAct;  lessonAct.lessonScreens ← lesson;  lessonAct.challengeScreens ← challenge;  lessonActs ← lessonActs ∪ {lessonAct};  learningAct.lessonActs ← lessonActs;  learningActs ← learningActs ∪ {learningAct};  layers. learningActs ← learningActs; |

### Game Composition

The Game Composition module uses the empty Structure object to create the final Game object. The Structure class organizes all the component objects, stored in the Layers object, into their corresponding screens and acts. The output Game object consists of a list of Act objects. The algorithm for this module is presented in Table 9.

Table 9 Compose Game Components Algorithm

|  |
| --- |
| **Input:** The layers object containing all entities with all dependencies set.  **Output:** A Game object containing the built and assembled game.  function buildGame  game : Game;  game ← layers.getStructure().createGame();  **Input:** All inputs are dependencies.  **Output:** A Game object representing the created game.  function createGame  acts : Act[];  screens : ScreenNode[];  screens ← theme.getIntro();  acts ← acts ∪ createActFromScreens(screens);  **for**(int i = 0; i < locale.getLearningActs().size(); i++)  screens ← locale.getAct(i);  acts ← acts ∪ createActFromScreens(screens);  screens ← theme.getOutro();  acts ← acts ∪ createActFromScreens(screens);  game : Game;  game.acts ← acts;  **call** wireUpActs(acts);  **return** game;  **Input:** The learning act id, and the screen type.  **Output:** A list of ScreenNode which represents the screens.  function buildScreens  lessonScreens : ScreenNode[];  currentScreen : UUID;  nextScreen : UUID;  currentScreen ← UUID.randomUUID();  themeStory : ThemeStory;  themeStory ← theme.getThemeStories()[learningActId];  themeStoryScreen : BaseScreen[];  **if** (screenType == ScreenType.LESSON\_STORY\_INTRO)  themeStoryScreen ← themeStory.getIntro();  **else**  screenTransitions[TransitionType.END\_OF\_STORY] ← currentScreen;  themeStoryScreen ← themeStory.getOutro();  **for** screen ϵ themeStoryScreen  nextScreen ← UUID.randomUUID();  lessonScreens ← lessonScreens ∪ buildScreen(learningActId, screen, localeScreens[screenType], currentScreen, nextScreen);  currentScreen ← nextScreen;  **if** (screenType == ScreenType.LESSON\_STORY\_INTRO)  screenTransitions[TransitionType.BEGINNING\_OF\_LESSON] ← nextScreen;  **return** lessonScreens |

### Game Script Output

The Game Script Output module takes the Game object created from the Compose Game Components function and converts it into one SimSYS game script, represented in XML. Similar to the load, the output also uses the JAXB library. The algorithm for this module is in Table 10.

Table 10 Output Game Algorithm

|  |
| --- |
| **Input:** A Game object with a complete game and a filename where the game should be exported.  **Output:** An xml file representing the game which is written to the disk.  function exportGame  jaxbContext : JAXBContext;  jaxbContext ← JAXBContext.newInstance(Game.class);  marshaller : Marshaller;  marshaller ← jaxbContext.createMarshaller();  marshaller[Marshaller.JAXB\_FORMATTED\_OUTPUT] ← true;  file : File;  file ← new File(exportFilename);  **call** marshaller.marshal(game, file); |

# Validation

The ISEGCB approach has been validated using four test cases. Each test case has a description, user inputs, expected output, actual output, and test status.

Tests have been completed using the Tool Support for the selection and a separate viewing tool for the pictures. The expected output has been created using Microsoft Word and individual calculations. Our current repository consists of five test games: SpaceGame, MathGame, VocabGame, SpaceGame, and HistoryGame. Each of which consists of six component xml files that can be selected for the new game.

The test cases have been made to test three major areas. Test Case #1 will test the old functionality of the game by selecting “No Preference” as the input. Test Case #2 and #3 will validate the selection algorithm by attempting to create games in the repository. Test Case #4 will test the ability to create a unique coherent game.

## Test Case #1

### Description

Test Case #1 is the output of the search algorithm when “No Preference” is selected for each of the input questions. This test will test if all old features are working properly. (Reads in six game component xml files and the output is one complete game xml file.) This test will also show if the MathGame is properly selected as the default game if there is no selected input criteria.

### Input

None- No Preference for all questions

### Expected Output

Characters0- Young, Player: Female, Casual Dress  
Lesson0- Math Related  
Challenge0- Math Related  
Locale0- Medieval Background  
Subject0- Math  
Theme0- Home Office Background, Gooble Theme

|  |  |
| --- | --- |
| Act 1:  Intro Screen 1 | Backdrop: Home Office  (Gooble Theme)  Player: Young, Female, Casual Dressed  Generic Interaction Prop  Textbox  Button: Next |
| Act 2  Lesson Screen 1 | Backdrop: Medieval  Player: Young, Female, Casual Dressed  Textbox:  Lesson- Math Related  Hero  Button: Continue |
| Act 2:  Challenge Screen 1 | Backdrop: Medieval  Player: Young, Female, Casual Dressed  Textbox  Hero  Button 1  Button 2  Button 3  Button 4  Textbox: Hero Interaction |
| Act 3:  Outro Screen | Backdrop: Home Office (Gooble Theme)  Player: Young, Female, Casual Dressed  Generic Interaction Prop  Textbox  Button: End Game |

**Figure 11: Test Case #1 Expected Output**

### Actual Output

|  |  |
| --- | --- |
| Act 1:  Intro Screen 1 |  |
| Act 2  Lesson Screen 1 |  |
| Act 2:  Challenge Screen 1 |  |
| Act 3:  Outro Screen |  |

**Figure 12: Test Case #1 Actual Output**

### Test Status

Pass

## Test Case #2

### Description

The purpose of this test case is to recreate the MathGame by searching its characteristics instead of relying on it being selected by default. This test will not only demonstrate the correctness of the selection algorithm but also its completeness as this is the first test implementing the selection algorithm.

### Input

Player Gender: Female

Character Age: Young

Character Dress: Casual

Game Theme: Gooble

Game Subject: Math

Game Setting: Natural

Challenge Difficulty: Easy

### Expected Output

MathGame:

Characters0- Young, Player: Female, Casual Dress  
Lesson0- Math Related  
Challenge0- Math Related  
Locale0- Medieval Background  
Subject0- Math  
Theme0- Home Office Background, Gooble Theme

|  |  |
| --- | --- |
| Act 1:  Intro Screen 2 | Backdrop: Computer Screen  (GoobleTheme)  Textbox:  Subject Title  Button: Continue  Textbox |
| Act 2:  Story Intro Screen | Backdrop: Medieval  Player: Young, Female, Casual Dressed  Textbox: Hero Interaction  Hero  Button: Continue  Textbox: Player Interaction |
| Act 2:  Lesson Screen | Backdrop: Medieval  Player: Young, Female, Casual Dressed  Textbox:  Lesson- Math Related  Hero  Button: Continue |
| Act 3:  Outro Screen | Backdrop: Home Office (Gooble Theme)  Player: Young, Female, Casual Dressed  Generic Interaction Prop  Textbox  Button: End Game |

Figure 13: Test Case #2 Expected Output

### Actual output

As expected.

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |

Figure 14: Test Case #2 Actual Output

### Test Status

Pass

## Test Case #3

### Description

This test will again test the correctness of the selection algorithm. With “professional” set as the setting and lesson selection inputs, the purpose of this test is to recreate the OfficeGame as closely as possible.

### Input

Player Gender: Male

Character Age: Young

Character Dress: Casual

Game Theme: Workplace

Game Subject: Professional

Game Setting: Professional

Challenge Difficulty: No Preference

### Expected Output

The calculations resulted in the selection of all the OfficeGame components except the Characters component. This is because Characters2 has all young characters while the OfficeGame, Character1, character component has some older characters making Character2 the most ideal for this search.

Characters2- Young, Player: Male, Casual  
Lesson1- Profession Related   
Challenge1- Profession Related  
Locale1- Cubical Background  
Subject1- Office Place  
Theme1- Building Front Background, Workplace Storyline

|  |  |
| --- | --- |
| Act 1:  Intro Screen 1 | Backdrop: Building Front  (Workplace Theme)  Textbox:  Subject Title  Button: Continue  Textbox |
| Act 2:  Lesson Screen | Backdrop: Cubicle  Player: Young, Male, Casual Dress  Textbox:  Lesson- Math Related  Hero  Button: Continue |
| Act 2:  Challenge Screen | Backdrop: Computer Screen  Textbox  Textbox:  Question- Office Related  Button 1  Button 2  Button 3  Button 4  Textbox: Hero InteractionTextbox:  Question- Math Related |
| Act 3:  Outro Screen | Backdrop: Office Night  Player: Young, Male, Casual Dress  Textbox  Button: End Game |

Figure 15: Test Case #3 Expected Output

### Actual Output

As expected.

|  |  |
| --- | --- |
| Act 1:  Intro Screen 1 |  |
| Act 2:  Lesson Screen |  |
| Result: |  |
| Result: |  |

Figure 17: Test Case #3 Actual Output

### Test Status

Pass

## Test Case #4

### Description

This test case tests the selection algorithm’s ability to create a unique coherent game. The input selection is random with the purpose of creating a completely new game that does not exist in the repository.

### Input

Player Gender: No Preference

Character Age: Young

Character Dress: Formal

Game Theme: Dream

Game Subject: Science

Game Setting: Natural

Challenge Difficulty: Hard

### Expected Output

Characters1- Young, Player: Male, Formal Dress  
Lesson3- English Related  
Challenge3 English Related  
Locale0- Medieval Background  
Subject2- Solar System  
Theme3- Dream

|  |  |
| --- | --- |
| Act 1:  Intro Screen 1 | Backdrop: Bedroom  (Dream Theme)  Player:  Male, Young, Formal Dress  Generic Interaction Prop  Textbox  Button: Next |
| Act 2:  Lesson Screen | Backdrop: Medieval  Player:  Male, Young, Formal Dress  Textbox:  Lesson- English Related  Hero  Button: Continue |
| Act 2:  Challenge Screen | Backdrop: Medieval  Player:  Male, Young, Formal Dress  Textbox  Hero  Textbox:  Question- English Related  Button 1  Button 2  Button 3  Button 4 |
| Act 3:  Outro Screen | Backdrop: Bedroom  (Dream Theme)  Player:  Male, Young, Formal Dress  Generic Interaction Prop  Textbox  Button: End Game |

Figure 18: Test Case #4 Expected Output

### Actual Output

Characters1- Young, Player: Male, Formal Dress  
Lesson0 – Math Related  
Challenge2 - Science Related  
Locale0- Medieval Background  
Subject2- Solar System  
Theme3- Bedroom Background, Dream Storyline

|  |  |
| --- | --- |
| Act 1:  Intro Screen 1 |  |
| Act 2:  Lesson Screen |  |
| Act 2:  Challenge Screen |  |
| Act 3:  Outro Screen |  |

Figure 19: Test Case #4 Actual Output

### Test Status

Fail

This test failed, but only slightly. Regardless of the expected output, the challenge and lesson need to match to make a coherent game. Further specification of search input options and a larger library to choose from may help. In addition, more metadata would also be useful.

# Conclusions and Future work

Serious educational games (SEGs) are recognized as valuable educational tools, with significant potential to provide immersive, engaging and fun environments for students across diverse domains. Developing these games, however, remains challenging. A component-based semi-automated game generation approach has the potential to support educators in rapidly creating games. This document has introduced a proposed ISEGCB approach, which has an interactive wizard user interface, a game generation engine, and a repository of components, game scripts, and game assets. The game components proposed in this preliminary work are the theme, characters, subject, locale, lesson, and challenge. The components have been captured in XML; selected, loaded, composed, and output as a game script [1].

The ability of mix and matching game components are now being tested and have shown great promise. As shown in the four test cases, the ability to combine different components is successful but not without limitations. One limitation with the components is that each game component should only be paired with components of a similar subject or theme. For example, a space lesson should not be paired with the challenge about unrelated vocabulary. The ability to create a variety of different games pertaining to one subject or theme will grow as the repository of games continues to grow.

Limitations pertaining to game components include:

* Characters are limited to four
* All lessons are limited to one backdrop
* All challenges are limited to one backdrop
* Theme component currently handling two separate features (Story Intro/Outro Screens and Intro/Outro Screens)
* Subject component currently has no function

Improvements with such limitations would enhance the user’s playing experience giving each game a more unique feel.

Some future works include a preview and editing feature on the new selection wizard. These features have been previously provided in a separate tool. Outside of the actual search feature and game component improvements, tools are also being developed to simplify the addition of new components along with their metadata to their respective repositories. This would allow each repository to be updated quickly by running one tool.

# References

1. Atthirawong, Walailak; Bart McCarthy (September, 2002). "An Application of the Analytical Hierarchy Process to International Location Decision-Making". In Gregory, Mike. Proceedings of The 7th Annual Cambridge International Manufacturing Symposium: Restructuring Global Manufacturing. Cambridge, England: University of Cambridge. pp. 1–18.
2. Berrittella, M.; A. Certa, M. Enea, P. Zito (January, 2007). "An Analytic Hierarchy Process for the Evaluation of Transport Policies to Reduce Climate Change Impacts". *Fondazione Eni Enrico Mattei (Milano)*.
3. Daconceicao, Ricardo, Carsten Locke, Kendra Cooper, and Shaun Longstreet. “Semi-automated Serious Educational Game Generation: a Component-based Game Engineering Approach.” IEEE CGAMES, 2013 (to appear).
4. Gee, J. P. (2007). What Video Games Have to Teach Us About Learning and Literacy. 2nd Edition, Palgrave Macmillan.
5. Grandzol, John R. (August 2005). "Improving the Faculty Selection Process in Higher Education: A Case for the Analytic Hierarchy Process" (PDF). IR Applications 6.
6. Heineman, G. T., Councill, W.T., Component-Based Software Engineering – Putting the Pieces Together, Addison Wesley, 2001.
7. Klopfer, E., Osterweil, S., & Salen, K. (2009). Moving Learning Games Forward. Massachusetts Institute of Technology, USA: The Education Arcade.
8. Longstreet, S. and Cooper, K., “A Meta-model for Developing Simulation Games in Higher Education and Professional Development Training”, IEEE 17th International Conference on Computer Games, 2012.
9. Lopes, R. and Bidarra, R. (2011). Adaptivity challenges in games and simulations: a survey, *IEEE Trans.on Computational Intelligence and AI in Games*, Vol. 3, Issue 2, pp. 85-99.
10. Object Management Group Unified Modeling Language (UML), version 2.4.1, 2011, available at http://www.omg.org.
11. Sanford, R. (2006). Teaching with Games: COTS games in the classroom. Innovating e-learning: Transforming Learning Experiences online conference.
12. Saaty, Thomas L. (2008-06). "Relative Measurement and its Generalization in Decision Making: Why Pairwise Comparisons are Central in Mathematics for the Measurement of Intangible Factors – The Analytic Hierarchy/Network Process". *RACSAM (Review of* the Royal *Spanish Academy of Sciences, Series A, Mathematics)* **102** (2): 251–318.
13. Scacchi, W. (2011). Modding as a basis for developing game systems. In *Proceeding of the 1st International Workshop on Games and Software Engineering*, pp. 5-8.
14. Shaffer, D. W. (2007). How Computer Games Help Children Learn. Palgrave Macmillan.
15. Szyperski, C., *Component Software*, Addison-Wesley, 2nd edition, 2002.
16. Walker, A. and Shelton, B.E. (2008). Problem-based educational games: connections, prescriptions, and assessment. .*J. of Inter. Learning Res*.,vol.19, no.4, pp. 663-684.
17. World Wide Web Consortium Extensible Markup Language (XML), version 1.0 (fifth edition), 2008, available at <http://www.w3.org>

# Appendix A. Characters XML

Table 15 Characters XML Example

|  |
| --- |
| <Characters>  <Alt>  <Directory>character\_1</Directory>  <Name>Chareeeesa</Name>  <Prefix>char1</Prefix>  </Alt>  <Hero>  <Directory>character\_19</Directory>  <Name>Sir Solvesalot</Name>  <Prefix>char19</Prefix>  </Hero>  <Villain>  <Directory>character\_22</Directory>  <Name>Calcutron</Name>  <Prefix>char22</Prefix>  </Villain>  <Player>  <Directory>character\_10</Directory>  <Name>Siva</Name>  <Prefix>char10</Prefix>  </Player>  </Characters> |

# Appendix B. Theme XML

Table 16: Theme.xml (Complete)

|  |
| --- |
| <?xml version=*"1.0"* encoding=*"utf-8"*?>  <Theme>  <IntroScreens>  <IntroScreen>  <Background>Backdrops\Home Office.png</Background>  <ThemeButtons>  <entry>  <key>NEXT</key>  <value>  <Height>50.566680908203125</Height>  <LocX>581.83331298828125</LocX>  <LocY>332.75</LocY>  <PathToAsset></PathToAsset>  <Text>Continue</Text>  <Width>157.86669921875</Width>  <behavior>  <behaviorType>TRANSITION\_BEHAVIOR</behaviorType>  <displayName>Transition Behavior</displayName>  <Transition>NEXT\_SCREEN</Transition>  <trigger>Click</trigger>  </behavior>  <name>Next</name>  </value>  </entry>  </ThemeButtons>  <GameObjects>  <GameObject>  <Height>78.316665649414062</Height>  <LocX>327.14999389648438</LocX>  <LocY>123.69999694824219</LocY>  <PathToAsset>Props\GenericInteraction\TextBubble.png</PathToAsset>  <Width>247.89999389648438</Width>  </GameObject>  </GameObjects>  <InformationBoxes>  <InformationBox>  <Height>59.20001220703125</Height>  <LocX>346.88330078125</LocX>  <LocY>128.63331604003906</LocY>  <Text>Today's lesson was difficult. I wonder if Gooble can help!</Text>  <Width>210.28340148925781</Width>  <name>Today's lesson was difficult. I wonder if Gooble can help!</name>  </InformationBox>  </InformationBoxes>  <ThemeCharacters>  <entry>  <key>PLAYER</key>  <value>  <Height>292.91667175292969</Height>  <LocX>209.36663818359375</LocX>  <LocY>130.48335266113281</LocY>  <Text></Text>  <Width>171.433349609375</Width>  <CharacterAssetType>WALK\_RIGHT\_BEHIND</CharacterAssetType>  <CharacterType>PLAYER</CharacterType>  </value>  </entry>  </ThemeCharacters>  </IntroScreen>  <IntroScreen>  <Background>Backdrops\FaceingPC.png</Background>  <ThemeButtons>  <entry>  <key>NEXT</key>  <value>  <Height>35.76666259765625</Height>  <LocX>422.11663818359375</LocX>  <LocY>266.1500244140625</LocY>  <Text>View</Text>  <Width>98.050018310546875</Width>  <behavior>  <behaviorType>TRANSITION\_BEHAVIOR</behaviorType>  <displayName>Transition Behavior</displayName>  <Transition>NEXT\_SCREEN</Transition>  <trigger>Click</trigger>  </behavior>  <name>View</name>  </value>  </entry>  </ThemeButtons>  <GameObjects>  </GameObjects>  <InformationBoxes>  <InformationBox>  <Height>67.833343505859375</Height>  <LocX>191.48330688476563</LocX>  <LocY>152.06666564941406</LocY>  <Text>Gooble!</Text>  <Width>390.35000610351562</Width>  <name>Gooble!</name>  </InformationBox>  <InformationBox>  <Height>36.999984741210938</Height>  <LocX>193.33329772949219</LocX>  <LocY>224.2166748046875</LocY>  <Text>Algebra</Text>  <Width>220.15000915527344</Width>  <name>Algebra</name>  </InformationBox>  <InformationBox>  <Height>64.133331298828125</Height>  <LocX>192.0999755859375</LocX>  <LocY>266.14999389648438</LocY>  <Text>Algebra Adventures</Text>  <Width>221.38333129882813</Width>  <name>Algebra Adventures</name>  </InformationBox>  </InformationBoxes>  <ThemeCharacters />  </IntroScreen>  <IntroScreen>  <Background>Backdrops\FaceingPC \_Warping.png</Background>  <ThemeButtons>  <entry>  <key>NEXT</key>  <value>  <Height>52.416656494140625</Height>  <LocX>636.71661376953125</LocX>  <LocY>336.45001220703125</LocY>  <Text>Continue</Text>  <Width>129.50006103515625</Width>  <behavior>  <behaviorType>TRANSITION\_BEHAVIOR</behaviorType>  <displayName>Transition Behavior</displayName>  <transition>NEXT\_ACT</transition>  <trigger>Click</trigger>  </behavior>  <name>Continue</name>  </value>  </entry>  </ThemeButtons>  <GameObjects />  <InformationBoxes />  <ThemeCharacters />  </IntroScreen>  </IntroScreens>  <OutroScreens>  <OutroScreen>  <Background>Backdrops\Home Office.png</Background>  <ThemeButtons>  <entry>  <key>NEXT</key>  <value>  <Height>46.86669921875</Height>  <LocX>570.7332763671875</LocX>  <LocY>325.9666748046875</LocY>  <PathToAsset>PathToAsset10</PathToAsset>  <Text>End Game</Text>  <Width>121.48333740234375</Width>  <behavior>  <behaviorType>END\_GAME\_BEHAVIOR</behaviorType>  <trigger>Click</trigger>  </behavior>  <name>End Game</name>  </value>  </entry>  </ThemeButtons>  <GameObjects>  <GameObject>  <Height>109.76666259765625</Height>  <LocX>362.29995727539062</LocX>  <LocY>98.416656494140625</LocY>  <PathToAsset>Props\GenericInteraction\TextBubble.png</PathToAsset>  <Width>277.50003051757812</Width>  </GameObject>  </GameObjects>  <InformationBoxes>  <InformationBox>  <Height>66.599990844726562</Height>  <LocX>375.86663818359375</LocX>  <LocY>113.83334350585938</LocY>  <Text>What a great game! I learned so much.</Text>  <Width>249.75003051757813</Width>  <name>What a great game! I learned so much.</name>  </InformationBox>  </InformationBoxes>  <ThemeCharacters>  <entry>  <key>PLAYER</key>  <value>  <Height>294.76665496826172</Height>  <LocX>218.61663818359375</LocX>  <LocY>126.16666412353516</LocY>  <PathToAsset>PathToAsset10</PathToAsset>  <Width>239.88330078125</Width>  <CharacterAssetType>RIGHT\_POINT\_UP</CharacterAssetType>  <CharacterType>PLAYER</CharacterType>  <Movements>  </Movements>  </value>  </entry>  </ThemeCharacters>  </OutroScreen>  </OutroScreens>  <ThemeStories>  <ThemeStory>  <StoryIntroScreens>  <StoryIntroScreen xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xsi:type=*"themeStoryScreenIntro"*>  <Buttons>  <Button>  <ButtonLocationType>NEXT</ButtonLocationType>  <Text>Continue</Text>  <Timer></Timer>  <TransitionType>NEXT\_SCREEN</TransitionType>  </Button>  </Buttons>  <LOCharacters>  <Character>  <CharacterType>PLAYER</CharacterType>  <MovementType></MovementType>  <Timer></Timer>  </Character>  <Character>  <CharacterType>HERO</CharacterType>  <MovementType></MovementType>  <Timer></Timer>  </Character>  </LOCharacters>  <InformationBoxes>  <InformationBox>  <TextType>PLAYER</TextType>  <Text>Who are you?</Text>  <Timer></Timer>  </InformationBox>  <InformationBox>  <TextType>HERO</TextType>  <Text>I am Sir Solvesalot!</Text>  <Timer></Timer>  </InformationBox>  <InformationBox>  <TextType>ALT1</TextType>  <Text>You awake in a strange land and run into strange man..</Text>  <Timer></Timer>  </InformationBox>  </InformationBoxes>  </StoryIntroScreen>  </StoryIntroScreens>  <StoryOutroScreens>  <StoryOutroScreen xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xsi:type=*"themeStoryScreenOutro"*>  <Buttons>  <Button>  <ButtonLocationType>NEXT</ButtonLocationType>  <Text>Warp!</Text>  <Timer></Timer>  <TransitionType>NEXT\_ACT</TransitionType>  </Button>  </Buttons>  <LOCharacters>  <Character>  <CharacterType>HERO</CharacterType>  <MovementType></MovementType>  <Timer></Timer>  </Character>  </LOCharacters>  <InformationBoxes>  <InformationBox>  <TextType>HERO</TextType>  <Text>Have a safe journey back to your world!</Text>  <Timer></Timer>  </InformationBox>  <InformationBox>  <TextType>ALT1</TextType>  <Text>After defeating Lord Calcutron, Sir Solvesalot takes you back to the warp zone that will take you back to your own world.</Text>  <Timer></Timer>  </InformationBox>  </InformationBoxes>  </StoryOutroScreen>  </StoryOutroScreens>  </ThemeStory>  </ThemeStories>  </Theme> |

# Appendix C. Lesson XML

Table 17 Lesson XML Example

|  |
| --- |
| <Lesson>  <LessonScreens>  <LessonScreen>  <Buttons>  <Button>  <ButtonLocationType>NEXT</ButtonLocationType>  <Text>Continue</Text>  <Timer></Timer>  <TransitionType>NEXT\_SCREEN</TransitionType>  </Button>  </Buttons>  <LOCharacters>  <Character>  <CharacterType>HERO</CharacterType>  <Timer></Timer>  </Character>  <Character>  <CharacterType>PLAYER</CharacterType>  <Timer></Timer>  </Character>  </LOCharacters>  <InformationBoxes>  <InformationBox>  <Text>To perform the addition operator you must add together the operands. EX: 2+4=6</Text>  <TextType>ALT1</TextType>  <Timer></Timer>  </InformationBox>  </InformationBoxes>  </LessonScreen>  </LessonScreens>  </Lesson> |

# Appendix D. Challenge XML

Table 18: Challenge.xml (Complete)

|  |
| --- |
| <Challenge>  <LessonChallenges>  <LessonChallenge>  <Buttons>  </Buttons>  <ChallengeOptions>  <ChallengeOption>  <ChallengeOptionType>BUTTON</ChallengeOptionType>  <ButtonLocationType>CHALLENGE\_1</ButtonLocationType>  <Reward>  <Points>-2147483646</Points>  </Reward>  <Text>2</Text>  <Timer></Timer>  <TransitionType>ADDITIONAL</TransitionType>  <AdditionalScreens>  <AdditionalScreen xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xsi:type=*"failureScreen"*>  <Buttons>  <Button>  <ButtonLocationType>NEXT</ButtonLocationType>  <Text>Retry</Text>  <Timer></Timer>  <TransitionType>CURRENT\_CHALLENGE</TransitionType>  </Button>  </Buttons>  <LOCharacters>  <Character>  <CharacterType>VILLIAN</CharacterType>  <Timer></Timer>  </Character>  <Character>  <CharacterType>HERO</CharacterType>  <Timer></Timer>  </Character>  <Character>  <CharacterType>PLAYER</CharacterType>  <Timer></Timer>  </Character>  </LOCharacters>  <InformationBoxes>  <InformationBox>  <Text>You answered incorrectly!</Text>  <TextType>ALT1</TextType>  <Timer></Timer>  </InformationBox>  <InformationBox>  <Text>Don't worry! You can try again.</Text>  <TextType>HERO</TextType>  <Timer></Timer>  </InformationBox>  </InformationBoxes>  <ChallengeOptions />  <Timer></Timer>  </AdditionalScreen>  </AdditionalScreens>  <ChallengeOptionType>BUTTON</ChallengeOptionType>  </ChallengeOption>  <ChallengeOption>  <ChallengeOptionType>BUTTON</ChallengeOptionType>  <ButtonLocationType>CHALLENGE\_2</ButtonLocationType>  <Reward>  <Points>-2147483646</Points>  </Reward>  <Text>4</Text>  <Timer></Timer>  <TransitionType>BEGINNING\_OF\_CHALLENGE</TransitionType>  </ChallengeOption>  <ChallengeOption>  <ChallengeOptionType>BUTTON</ChallengeOptionType>  <ButtonLocationType>CHALLENGE\_3</ButtonLocationType>  <Reward>  <Points>2</Points>  </Reward>  <Text>6</Text>  <Timer></Timer>  <TransitionType>END\_OF\_STORY</TransitionType>  </ChallengeOption>  <ChallengeOption>  <ChallengeOptionType>BUTTON</ChallengeOptionType>  <ButtonLocationType>CHALLENGE\_4</ButtonLocationType>  <Reward>  <Points>-2147483646</Points>  </Reward>  <Text>8</Text>  <Timer></Timer>  <TransitionType>BEGINNING\_OF\_LESSON</TransitionType>  </ChallengeOption>  </ChallengeOptions>  <LOCharacters>  <Character>  <CharacterType>HERO</CharacterType>  <MovementType></MovementType>  <Timer></Timer>  <Character>  <CharacterType>PLAYER</CharacterType>  <Timer></Timer>  </Character>  </Character>  </LOCharacters>  <InformationBoxes>  <InformationBox>  <Text>You can do it!!</Text>  <TextType>HERO</TextType>  <Timer></Timer>  </InformationBox>  <InformationBox>  <Text>Using Algebra solve:</Text>  <TextType>CHALLENGE\_QUESTION</TextType>  <Timer></Timer>  </InformationBox>  <InformationBox>  <Text>2 + 4</Text>  <TextType>CHALLENGE\_DESCRIPTION</TextType>  <Timer></Timer>  </InformationBox>  </InformationBoxes>  <Timer>60</Timer>  </LessonChallenge>  </LessonChallenges>  </Challenge> |

# Appendix E. Locale XML

Table 19: Locale.xml (Complete)

|  |
| --- |
| <?xml version=*"1.0"* encoding=*"utf-8"*?>  <Locale>  <LocaleScreens>  <entry>  <key>LESSON</key>  <value>  <Background>Backdrops\Medieval.png</Background>  <Buttons>  <entry>  <key>BACK</key>  <value>  <Height>1</Height>  <LocX>1</LocX>  <LocY>1</LocY>  <Width>1</Width>  </value>  </entry>  <entry>  <key>NEXT</key>  <value>  <Height>48.0999755859375</Height>  <LocX>571.96661376953125</LocX>  <LocY>335.2166748046875</LocY>  <Width>128.88336181640625</Width>  </value>  </entry>  </Buttons>  <Characters>  <entry>  <key>PLAYER</key>  <value>  <Height>91.26666259765625</Height>  <LocX>135.98329162597656</LocX>  <LocY>313.01666259765625</LocY>  <Width>104.21666717529297</Width>  <CharacterAssetType>RIGHT\_POINT\_NO</CharacterAssetType>  <CharacterType>PLAYER</CharacterType>  <Movements>  <Movement>  <AnimationSequences>  <AnimationSequence>WALK\_RIGHT\_OPEN</AnimationSequence>  <AnimationSequence>WALK\_LEFT\_BEHIND</AnimationSequence>  </AnimationSequences>  <EndX>136</EndX>  <EndY>313</EndY>  <MovementType>WALK\_ONTO\_SCREEN</MovementType>  <Speed>1</Speed>  <StartX>1</StartX>  <StartY>313</StartY>  </Movement>  <Movement>  <AnimationSequences>  <AnimationSequence>WALK\_RIGHT\_BEHIND</AnimationSequence>  <AnimationSequence>WALK\_LEFT\_OPEN</AnimationSequence>  </AnimationSequences>  <EndX>1</EndX>  <EndY>313</EndY>  <MovementType>WALK\_OFF\_SCREEN</MovementType>  <Speed>1</Speed>  <StartX>136</StartX>  <StartY>313</StartY>  </Movement>  </Movements>  </value>  </entry>  <entry>  <key>HERO</key>  <value>  <Height>133.816650390625</Height>  <LocX>295.08334350585938</LocX>  <LocY>277.25</LocY>  <Width>83.249984741210938</Width>  <CharacterAssetType>LEFT\_POINT\_UP</CharacterAssetType>  <CharacterType>HERO</CharacterType>  <Movements>  <Movement>  <AnimationSequences>  <AnimationSequence>WALK\_RIGHT\_OPEN</AnimationSequence>  <AnimationSequence>WALK\_LEFT\_BEHIND</AnimationSequence>  </AnimationSequences>  <EndX>136</EndX>  <EndY>313</EndY>  <MovementType>WALK\_ONTO\_SCREEN</MovementType>  <Speed>1</Speed>  <StartX>1</StartX>  <StartY>313</StartY>  </Movement>  <Movement>  <AnimationSequences>  <AnimationSequence>WALK\_RIGHT\_BEHIND</AnimationSequence>  <AnimationSequence>WALK\_LEFT\_OPEN</AnimationSequence>  </AnimationSequences>  <EndX>1</EndX>  <EndY>313</EndY>  <MovementType>WALK\_OFF\_SCREEN</MovementType>  <Speed>1</Speed>  <StartX>136</StartX>  <StartY>313</StartY>  </Movement>  </Movements> </value>  </entry>  <entry>  <key>VILLIAN</key>  <value>  <Height>111.61668395996094</Height>  <LocX>444.31666564941406</LocX>  <LocY>129.86666870117188</LocY>  <Width>98.050003051757812</Width>  <CharacterAssetType>LEFT\_EVIL</CharacterAssetType>  <CharacterType>VILLIAN</CharacterType>  <Movements>  <Movement>  <AnimationSequences>  <AnimationSequence>WALK\_RIGHT\_OPEN</AnimationSequence>  <AnimationSequence>WALK\_LEFT\_BEHIND</AnimationSequence>  </AnimationSequences>  <EndX>136</EndX>  <EndY>313</EndY>  <MovementType>WALK\_ONTO\_SCREEN</MovementType>  <Speed>1</Speed>  <StartX>1</StartX>  <StartY>313</StartY>  </Movement>  <Movement>  <AnimationSequences>  <AnimationSequence>WALK\_RIGHT\_BEHIND</AnimationSequence>  <AnimationSequence>WALK\_LEFT\_OPEN</AnimationSequence>  </AnimationSequences>  <EndX>1</EndX>  <EndY>313</EndY>  <MovementType>WALK\_OFF\_SCREEN</MovementType>  <Speed>1</Speed>  <StartX>136</StartX>  <StartY>313</StartY>  </Movement>  </Movements>  </value>  </entry>  </Characters>  <GameObjects>  <!-- Player text box -->  <GameObject>  <Height>143.68331909179688</Height>  <LocX>36.083301544189453</LocX>  <LocY>194.00001525878906</LocY>  <PathToAsset>Props\GenericInteraction\Regular Speech 2.png</PathToAsset>  <Width>155.40000534057617</Width>  </GameObject>  <!-- Hero text box -->  <GameObject>  <Height>79.550003051757812</Height>  <LocX>278.43331909179688</LocX>  <LocY>251.35002136230469</LocY>  <PathToAsset>Props\GenericInteraction\TextBubble.png</PathToAsset>  <Width>235.56668090820313</Width>  </GameObject>  <!-- Villian text box -->  <GameObject>  <Height>146.15000915527344</Height>  <LocX>68.149978637695312</LocX>  <LocY>205.71669006347656</LocY>  <PathToAsset>Props\GenericInteraction\TextBubble.png</PathToAsset>  <Width>219.5333251953125</Width>  </GameObject>  </GameObjects>  <InformationBoxes>  <entry>  <key>PLAYER</key>  <value>  <Height>46.866683959960938</Height>  <LocX>61.36663818359375</LocX>  <LocY>224.21665954589844</LocY>  <Width>84.48333740234375</Width>  </value>  </entry>  <entry>  <key>HERO</key>  <value>  <Height>51.183334350585938</Height>  <LocX>292</LocX>  <LocY>261.83332824707031</LocY>  <Width>207.816650390625</Width>  </value>  </entry>  <entry>  <key>VILLIAN</key>  <value>  <Height>107.91666436195374</Height>  <LocX>-0.30003738403320312</LocX>  <LocY>-1.4833300113677979</LocY>  <Width>772.0666389465332</Width>  </value>  </entry>  <entry>  <key>ALT1</key>  <value>  <Height>151.70001220703125</Height>  <LocX>9.5666418075561523</LocX>  <LocY>129.25</LocY>  <Width>333.00000858306885</Width>  </value>  </entry>  </InformationBoxes>  </value>  </entry>  <entry>  <key>CHALLENGE</key>  <value>  <Background>Backdrops\Medieval.png</Background>  <Buttons>  <entry>  <key>CHALLENGE\_1</key>  <value>  <Height>33.91668701171875</Height>  <LocX>322.83331298828125</LocX>  <LocY>230.38333129882813</LocY>  <Width>284.9000244140625</Width>  </value>  </entry>  <entry>  <key>CHALLENGE\_2</key>  <value>  <Height>33.91668701171875</Height>  <LocX>322.83331298828125</LocX>  <LocY>265.53335571289062</LocY>  <Width>284.9000244140625</Width>  </value>  </entry>  <entry>  <key>CHALLENGE\_3</key>  <value>  <Height>33.91668701171875</Height>  <LocX>322.83331298828125</LocX>  <LocY>303.76666259765625</LocY>  <Width>284.9000244140625</Width>  </value>  </entry>  <entry>  <key>CHALLENGE\_4</key>  <value>  <Height>33.91668701171875</Height>  <LocX>322.83331298828125</LocX>  <LocY>343.23333740234375</LocY>  <Width>284.9000244140625</Width>  </value>  </entry>  </Buttons>  <Characters>  <entry>  <key>PLAYER</key>  <value>  <Height>91.26666259765625</Height>  <LocX>135.98329162597656</LocX>  <LocY>313.01666259765625</LocY>  <Width>104.21666717529297</Width>  <CharacterAssetType>RIGHT\_POINT\_NO</CharacterAssetType>  <CharacterType>PLAYER</CharacterType>  <Movements>  <Movement>  <AnimationSequences>  <AnimationSequence>WALK\_RIGHT\_OPEN</AnimationSequence>  <AnimationSequence>WALK\_LEFT\_BEHIND</AnimationSequence>  </AnimationSequences>  <EndX>136</EndX>  <EndY>313</EndY>  <MovementType>WALK\_ONTO\_SCREEN</MovementType>  <Speed>1</Speed>  <StartX>1</StartX>  <StartY>313</StartY>  </Movement>  <Movement>  <AnimationSequences>  <AnimationSequence>WALK\_RIGHT\_BEHIND</AnimationSequence>  <AnimationSequence>WALK\_LEFT\_OPEN</AnimationSequence>  </AnimationSequences>  <EndX>1</EndX>  <EndY>313</EndY>  <MovementType>WALK\_OFF\_SCREEN</MovementType>  <Speed>1</Speed>  <StartX>136</StartX>  <StartY>313</StartY>  </Movement>  </Movements>  </value>  </entry>  <entry>  <key>HERO</key>  <value>  <Height>133.816650390625</Height>  <LocX>295.08334350585938</LocX>  <LocY>277.25</LocY>  <Width>83.249984741210938</Width>  <CharacterAssetType>LEFT\_POINT\_UP</CharacterAssetType>  <CharacterType>HERO</CharacterType>  <Movements>  <Movement>  <AnimationSequences>  <AnimationSequence>WALK\_RIGHT\_OPEN</AnimationSequence>  <AnimationSequence>WALK\_LEFT\_BEHIND</AnimationSequence>  </AnimationSequences>  <EndX>136</EndX>  <EndY>313</EndY>  <MovementType>WALK\_ONTO\_SCREEN</MovementType>  <Speed>1</Speed>  <StartX>1</StartX>  <StartY>313</StartY>  </Movement>  <Movement>  <AnimationSequences>  <AnimationSequence>WALK\_RIGHT\_BEHIND</AnimationSequence>  <AnimationSequence>WALK\_LEFT\_OPEN</AnimationSequence>  </AnimationSequences>  <EndX>1</EndX>  <EndY>313</EndY>  <MovementType>WALK\_OFF\_SCREEN</MovementType>  <Speed>1</Speed>  <StartX>136</StartX>  <StartY>313</StartY>  </Movement>  </Movements> </value>  </entry>  <entry>  <key>VILLIAN</key>  <value>  <Height>111.61668395996094</Height>  <LocX>444.31666564941406</LocX>  <LocY>129.86666870117188</LocY>  <Width>98.050003051757812</Width>  <CharacterAssetType>LEFT\_EVIL</CharacterAssetType>  <CharacterType>VILLIAN</CharacterType>  <Movements>  <Movement>  <AnimationSequences>  <AnimationSequence>WALK\_RIGHT\_OPEN</AnimationSequence>  <AnimationSequence>WALK\_LEFT\_BEHIND</AnimationSequence>  </AnimationSequences>  <EndX>136</EndX>  <EndY>313</EndY>  <MovementType>WALK\_ONTO\_SCREEN</MovementType>  <Speed>1</Speed>  <StartX>1</StartX>  <StartY>313</StartY>  </Movement>  <Movement>  <AnimationSequences>  <AnimationSequence>WALK\_RIGHT\_BEHIND</AnimationSequence>  <AnimationSequence>WALK\_LEFT\_OPEN</AnimationSequence>  </AnimationSequences>  <EndX>1</EndX>  <EndY>313</EndY>  <MovementType>WALK\_OFF\_SCREEN</MovementType>  <Speed>1</Speed>  <StartX>136</StartX>  <StartY>313</StartY>  </Movement>  </Movements>  </value>  </entry>  </Characters>  <GameObjects>  <!-- Hero text bubble -->  <GameObject>  <Height>61.666656494140625</Height>  <LocX>69.383308410644531</LocX>  <LocY>273.55001831054688</LocY>  <PathToAsset>Props\GenericInteraction\TextBubbleLarge.png</PathToAsset>  <Width>164.65000152587891</Width>  </GameObject>  </GameObjects>  <InformationBoxes>  <entry>  <key>CHALLENGE\_DESCRIPTION</key>  <value>  <Height>49.95001220703125</Height>  <LocX>184.08331298828125</LocX>  <LocY>115.68333435058594</LocY>  <Width>568.566650390625</Width>  </value>  </entry>  <entry>  <key>CHALLENGE\_QUESTION</key>  <value>  <Height>56.116668701171875</Height>  <LocX>186.54997253417969</LocX>  <LocY>171.18333435058594</LocY>  <Width>562.40003967285156</Width>  </value>  </entry>  <entry>  <key>HERO</key>  <value>  <Height>40.70001220703125</Height>  <LocX>76.783298492431641</LocX>  <LocY>280.94998168945312</LocY>  <Width>149.23333358764648</Width>  </value>  </entry>  </InformationBoxes>  </value>  </entry>  <entry>  <key>FAILURE</key>  <value>  <Background>Backdrops\Medieval.png</Background>  <Buttons>  <entry>  <key>BACK</key>  <value>  <Height>1</Height>  <LocX>1</LocX>  <LocY>1</LocY>  <Width>1</Width>  </value>  </entry>  <entry>  <key>NEXT</key>  <value>  <Height>48.0999755859375</Height>  <LocX>571.96661376953125</LocX>  <LocY>335.2166748046875</LocY>  <Width>128.88336181640625</Width>  </value>  </entry>  </Buttons>  <Characters>  <entry>  <key>PLAYER</key>  <value>  <Height>91.26666259765625</Height>  <LocX>135.98329162597656</LocX>  <LocY>313.01666259765625</LocY>  <Width>104.21666717529297</Width>  <CharacterAssetType>RIGHT\_POINT\_NO</CharacterAssetType>  <CharacterType>PLAYER</CharacterType>  <Movements>  <Movement>  <AnimationSequences>  <AnimationSequence>WALK\_RIGHT\_OPEN</AnimationSequence>  <AnimationSequence>WALK\_LEFT\_BEHIND</AnimationSequence>  </AnimationSequences>  <EndX>136</EndX>  <EndY>313</EndY>  <MovementType>WALK\_ONTO\_SCREEN</MovementType>  <Speed>1</Speed>  <StartX>1</StartX>  <StartY>313</StartY>  </Movement>  <Movement>  <AnimationSequences>  <AnimationSequence>WALK\_RIGHT\_BEHIND</AnimationSequence>  <AnimationSequence>WALK\_LEFT\_OPEN</AnimationSequence>  </AnimationSequences>  <EndX>1</EndX>  <EndY>313</EndY>  <MovementType>WALK\_OFF\_SCREEN</MovementType>  <Speed>1</Speed>  <StartX>136</StartX>  <StartY>313</StartY>  </Movement>  </Movements>  </value>  </entry>  <entry>  <key>HERO</key>  <value>  <Height>133.816650390625</Height>  <LocX>295.08334350585938</LocX>  <LocY>277.25</LocY>  <Width>83.249984741210938</Width>  <CharacterAssetType>LEFT\_POINT\_UP</CharacterAssetType>  <CharacterType>HERO</CharacterType>  <Movements>  <Movement>  <AnimationSequences>  <AnimationSequence>WALK\_RIGHT\_OPEN</AnimationSequence>  <AnimationSequence>WALK\_LEFT\_BEHIND</AnimationSequence>  </AnimationSequences>  <EndX>136</EndX>  <EndY>313</EndY>  <MovementType>WALK\_ONTO\_SCREEN</MovementType>  <Speed>1</Speed>  <StartX>1</StartX>  <StartY>313</StartY>  </Movement>  <Movement>  <AnimationSequences>  <AnimationSequence>WALK\_RIGHT\_BEHIND</AnimationSequence>  <AnimationSequence>WALK\_LEFT\_OPEN</AnimationSequence>  </AnimationSequences>  <EndX>1</EndX>  <EndY>313</EndY>  <MovementType>WALK\_OFF\_SCREEN</MovementType>  <Speed>1</Speed>  <StartX>136</StartX>  <StartY>313</StartY>  </Movement>  </Movements> </value>  </entry>  <entry>  <key>VILLIAN</key>  <value>  <Height>111.61668395996094</Height>  <LocX>444.31666564941406</LocX>  <LocY>129.86666870117188</LocY>  <Width>98.050003051757812</Width>  <CharacterAssetType>LEFT\_EVIL</CharacterAssetType>  <CharacterType>VILLIAN</CharacterType>  <Movements>  <Movement>  <AnimationSequences>  <AnimationSequence>WALK\_RIGHT\_OPEN</AnimationSequence>  <AnimationSequence>WALK\_LEFT\_BEHIND</AnimationSequence>  </AnimationSequences>  <EndX>136</EndX>  <EndY>313</EndY>  <MovementType>WALK\_ONTO\_SCREEN</MovementType>  <Speed>1</Speed>  <StartX>1</StartX>  <StartY>313</StartY>  </Movement>  <Movement>  <AnimationSequences>  <AnimationSequence>WALK\_RIGHT\_BEHIND</AnimationSequence>  <AnimationSequence>WALK\_LEFT\_OPEN</AnimationSequence>  </AnimationSequences>  <EndX>1</EndX>  <EndY>313</EndY>  <MovementType>WALK\_OFF\_SCREEN</MovementType>  <Speed>1</Speed>  <StartX>136</StartX>  <StartY>313</StartY>  </Movement>  </Movements>  </value>  </entry>  </Characters>  <GameObjects>  <!-- Player text box -->  <GameObject>  <Height>143.68331909179688</Height>  <LocX>36.083301544189453</LocX>  <LocY>194.00001525878906</LocY>  <PathToAsset>Props\GenericInteraction\Regular Speech 2.png</PathToAsset>  <Width>155.40000534057617</Width>  </GameObject>  <!-- Hero text box -->  <GameObject>  <Height>79.550003051757812</Height>  <LocX>278.43331909179688</LocX>  <LocY>251.35002136230469</LocY>  <PathToAsset>Props\GenericInteraction\TextBubble.png</PathToAsset>  <Width>235.56668090820313</Width>  </GameObject>  <!-- Villian text box -->  <GameObject>  <Height>146.15000915527344</Height>  <LocX>68.149978637695312</LocX>  <LocY>205.71669006347656</LocY>  <PathToAsset>Props\GenericInteraction\TextBubble.png</PathToAsset>  <Width>219.5333251953125</Width>  </GameObject>  </GameObjects>  <InformationBoxes>  <entry>  <key>PLAYER</key>  <value>  <Height>46.866683959960938</Height>  <LocX>61.36663818359375</LocX>  <LocY>224.21665954589844</LocY>  <Width>84.48333740234375</Width>  </value>  </entry>  <entry>  <key>HERO</key>  <value>  <Height>51.183334350585938</Height>  <LocX>292</LocX>  <LocY>261.83332824707031</LocY>  <Width>207.816650390625</Width>  </value>  </entry>  <entry>  <key>VILLIAN</key>  <value>  <Height>107.91666436195374</Height>  <LocX>-0.30003738403320312</LocX>  <LocY>-1.4833300113677979</LocY>  <Width>772.0666389465332</Width>  </value>  </entry>  <entry>  <key>ALT1</key>  <value>  <Height>151.70001220703125</Height>  <LocX>9.5666418075561523</LocX>  <LocY>129.25</LocY>  <Width>333.00000858306885</Width>  </value>  </entry>  </InformationBoxes>  </value>  </entry>  <entry>  <key>LESSON\_STORY\_INTRO</key>  <value>  <Background>Backdrops\Medieval.png</Background>  <Buttons>  <entry>  <key>BACK</key>  <value>  <Height>1</Height>  <LocX>1</LocX>  <LocY>1</LocY>  <Width>1</Width>  </value>  </entry>  <entry>  <key>NEXT</key>  <value>  <Height>48.0999755859375</Height>  <LocX>571.96661376953125</LocX>  <LocY>335.2166748046875</LocY>  <Width>128.88336181640625</Width>  </value>  </entry>  </Buttons>  <Characters>  <entry>  <key>PLAYER</key>  <value>  <Height>91.26666259765625</Height>  <LocX>135.98329162597656</LocX>  <LocY>313.01666259765625</LocY>  <Width>104.21666717529297</Width>  <CharacterAssetType>RIGHT\_POINT\_NO</CharacterAssetType>  <CharacterType>PLAYER</CharacterType>  <Movements>  <Movement>  <AnimationSequences>  <AnimationSequence>WALK\_RIGHT\_OPEN</AnimationSequence>  <AnimationSequence>WALK\_LEFT\_BEHIND</AnimationSequence>  </AnimationSequences>  <EndX>136</EndX>  <EndY>313</EndY>  <MovementType>WALK\_ONTO\_SCREEN</MovementType>  <Speed>1</Speed>  <StartX>1</StartX>  <StartY>313</StartY>  </Movement>  <Movement>  <AnimationSequences>  <AnimationSequence>WALK\_RIGHT\_BEHIND</AnimationSequence>  <AnimationSequence>WALK\_LEFT\_OPEN</AnimationSequence>  </AnimationSequences>  <EndX>1</EndX>  <EndY>313</EndY>  <MovementType>WALK\_OFF\_SCREEN</MovementType>  <Speed>1</Speed>  <StartX>136</StartX>  <StartY>313</StartY>  </Movement>  </Movements>  </value>  </entry>  <entry>  <key>HERO</key>  <value>  <Height>133.816650390625</Height>  <LocX>295.08334350585938</LocX>  <LocY>277.25</LocY>  <Width>83.249984741210938</Width>  <CharacterAssetType>LEFT\_POINT\_UP</CharacterAssetType>  <CharacterType>HERO</CharacterType>  <Movements>  <Movement>  <AnimationSequences>  <AnimationSequence>WALK\_RIGHT\_OPEN</AnimationSequence>  <AnimationSequence>WALK\_LEFT\_BEHIND</AnimationSequence>  </AnimationSequences>  <EndX>136</EndX>  <EndY>313</EndY>  <MovementType>WALK\_ONTO\_SCREEN</MovementType>  <Speed>1</Speed>  <StartX>1</StartX>  <StartY>313</StartY>  </Movement>  <Movement>  <AnimationSequences>  <AnimationSequence>WALK\_RIGHT\_BEHIND</AnimationSequence>  <AnimationSequence>WALK\_LEFT\_OPEN</AnimationSequence>  </AnimationSequences>  <EndX>1</EndX>  <EndY>313</EndY>  <MovementType>WALK\_OFF\_SCREEN</MovementType>  <Speed>1</Speed>  <StartX>136</StartX>  <StartY>313</StartY>  </Movement>  </Movements> </value>  </entry>  <entry>  <key>VILLIAN</key>  <value>  <Height>111.61668395996094</Height>  <LocX>444.31666564941406</LocX>  <LocY>129.86666870117188</LocY>  <Width>98.050003051757812</Width>  <CharacterAssetType>LEFT\_EVIL</CharacterAssetType>  <CharacterType>VILLIAN</CharacterType>  <Movements>  <Movement>  <AnimationSequences>  <AnimationSequence>WALK\_RIGHT\_OPEN</AnimationSequence>  <AnimationSequence>WALK\_LEFT\_BEHIND</AnimationSequence>  </AnimationSequences>  <EndX>136</EndX>  <EndY>313</EndY>  <MovementType>WALK\_ONTO\_SCREEN</MovementType>  <Speed>1</Speed>  <StartX>1</StartX>  <StartY>313</StartY>  </Movement>  <Movement>  <AnimationSequences>  <AnimationSequence>WALK\_RIGHT\_BEHIND</AnimationSequence>  <AnimationSequence>WALK\_LEFT\_OPEN</AnimationSequence>  </AnimationSequences>  <EndX>1</EndX>  <EndY>313</EndY>  <MovementType>WALK\_OFF\_SCREEN</MovementType>  <Speed>1</Speed>  <StartX>136</StartX>  <StartY>313</StartY>  </Movement>  </Movements>  </value>  </entry>  </Characters>  <GameObjects>  <!-- Player text box -->  <GameObject>  <Height>143.68331909179688</Height>  <LocX>36.083301544189453</LocX>  <LocY>194.00001525878906</LocY>  <PathToAsset>Props\GenericInteraction\Regular Speech 2.png</PathToAsset>  <Width>155.40000534057617</Width>  </GameObject>  <!-- Hero text box -->  <GameObject>  <Height>79.550003051757812</Height>  <LocX>278.43331909179688</LocX>  <LocY>251.35002136230469</LocY>  <PathToAsset>Props\GenericInteraction\TextBubble.png</PathToAsset>  <Width>235.56668090820313</Width>  </GameObject>  <!-- Villian text box -->  <GameObject>  <Height>146.15000915527344</Height>  <LocX>68.149978637695312</LocX>  <LocY>205.71669006347656</LocY>  <PathToAsset>Props\GenericInteraction\TextBubble.png</PathToAsset>  <Width>219.5333251953125</Width>  </GameObject>  </GameObjects>  <InformationBoxes>  <entry>  <key>PLAYER</key>  <value>  <Height>46.866683959960938</Height>  <LocX>61.36663818359375</LocX>  <LocY>224.21665954589844</LocY>  <Width>84.48333740234375</Width>  </value>  </entry>  <entry>  <key>HERO</key>  <value>  <Height>51.183334350585938</Height>  <LocX>292</LocX>  <LocY>261.83332824707031</LocY>  <Width>207.816650390625</Width>  </value>  </entry>  <entry>  <key>VILLIAN</key>  <value>  <Height>107.91666436195374</Height>  <LocX>-0.30003738403320312</LocX>  <LocY>-1.4833300113677979</LocY>  <Width>772.0666389465332</Width>  </value>  </entry>  <entry>  <key>ALT1</key>  <value>  <Height>151.70001220703125</Height>  <LocX>9.5666418075561523</LocX>  <LocY>129.25</LocY>  <Width>333.00000858306885</Width>  </value>  </entry>  </InformationBoxes>  </value>  </entry>  <entry>  <key>LESSON\_STORY\_OUTRO</key>  <value>  <Background>Backdrops\Medieval.png</Background>  <Buttons>  <entry>  <key>BACK</key>  <value>  <Height>1</Height>  <LocX>1</LocX>  <LocY>1</LocY>  <Width>1</Width>  </value>  </entry>  <entry>  <key>NEXT</key>  <value>  <Height>48.0999755859375</Height>  <LocX>571.96661376953125</LocX>  <LocY>335.2166748046875</LocY>  <Width>128.88336181640625</Width>  </value>  </entry>  </Buttons>  <Characters>  <entry>  <key>PLAYER</key>  <value>  <Height>91.26666259765625</Height>  <LocX>135.98329162597656</LocX>  <LocY>313.01666259765625</LocY>  <Width>104.21666717529297</Width>  <CharacterAssetType>RIGHT\_POINT\_NO</CharacterAssetType>  <CharacterType>PLAYER</CharacterType>  <Movements>  <Movement>  <AnimationSequences>  <AnimationSequence>WALK\_RIGHT\_OPEN</AnimationSequence>  <AnimationSequence>WALK\_LEFT\_BEHIND</AnimationSequence>  </AnimationSequences>  <EndX>136</EndX>  <EndY>313</EndY>  <MovementType>WALK\_ONTO\_SCREEN</MovementType>  <Speed>1</Speed>  <StartX>1</StartX>  <StartY>313</StartY>  </Movement>  <Movement>  <AnimationSequences>  <AnimationSequence>WALK\_RIGHT\_BEHIND</AnimationSequence>  <AnimationSequence>WALK\_LEFT\_OPEN</AnimationSequence>  </AnimationSequences>  <EndX>1</EndX>  <EndY>313</EndY>  <MovementType>WALK\_OFF\_SCREEN</MovementType>  <Speed>1</Speed>  <StartX>136</StartX>  <StartY>313</StartY>  </Movement>  </Movements>  </value>  </entry>  <entry>  <key>HERO</key>  <value>  <Height>133.816650390625</Height>  <LocX>295.08334350585938</LocX>  <LocY>277.25</LocY>  <Width>83.249984741210938</Width>  <CharacterAssetType>LEFT\_POINT\_UP</CharacterAssetType>  <CharacterType>HERO</CharacterType>  <Movements>  <Movement>  <AnimationSequences>  <AnimationSequence>WALK\_RIGHT\_OPEN</AnimationSequence>  <AnimationSequence>WALK\_LEFT\_BEHIND</AnimationSequence>  </AnimationSequences>  <EndX>136</EndX>  <EndY>313</EndY>  <MovementType>WALK\_ONTO\_SCREEN</MovementType>  <Speed>1</Speed>  <StartX>1</StartX>  <StartY>313</StartY>  </Movement>  <Movement>  <AnimationSequences>  <AnimationSequence>WALK\_RIGHT\_BEHIND</AnimationSequence>  <AnimationSequence>WALK\_LEFT\_OPEN</AnimationSequence>  </AnimationSequences>  <EndX>1</EndX>  <EndY>313</EndY>  <MovementType>WALK\_OFF\_SCREEN</MovementType>  <Speed>1</Speed>  <StartX>136</StartX>  <StartY>313</StartY>  </Movement>  </Movements> </value>  </entry>  <entry>  <key>VILLIAN</key>  <value>  <Height>111.61668395996094</Height>  <LocX>444.31666564941406</LocX>  <LocY>129.86666870117188</LocY>  <Width>98.050003051757812</Width>  <CharacterAssetType>LEFT\_EVIL</CharacterAssetType>  <CharacterType>VILLIAN</CharacterType>  <Movements>  <Movement>  <AnimationSequences>  <AnimationSequence>WALK\_RIGHT\_OPEN</AnimationSequence>  <AnimationSequence>WALK\_LEFT\_BEHIND</AnimationSequence>  </AnimationSequences>  <EndX>136</EndX>  <EndY>313</EndY>  <MovementType>WALK\_ONTO\_SCREEN</MovementType>  <Speed>1</Speed>  <StartX>1</StartX>  <StartY>313</StartY>  </Movement>  <Movement>  <AnimationSequences>  <AnimationSequence>WALK\_RIGHT\_BEHIND</AnimationSequence>  <AnimationSequence>WALK\_LEFT\_OPEN</AnimationSequence>  </AnimationSequences>  <EndX>1</EndX>  <EndY>313</EndY>  <MovementType>WALK\_OFF\_SCREEN</MovementType>  <Speed>1</Speed>  <StartX>136</StartX>  <StartY>313</StartY>  </Movement>  </Movements>  </value>  </entry>  </Characters>  <GameObjects>  <!-- Player text box -->  <GameObject>  <Height>143.68331909179688</Height>  <LocX>36.083301544189453</LocX>  <LocY>194.00001525878906</LocY>  <PathToAsset>Props\GenericInteraction\Regular Speech 2.png</PathToAsset>  <Width>155.40000534057617</Width>  </GameObject>  <!-- Hero text box -->  <GameObject>  <Height>79.550003051757812</Height>  <LocX>278.43331909179688</LocX>  <LocY>251.35002136230469</LocY>  <PathToAsset>Props\GenericInteraction\TextBubble.png</PathToAsset>  <Width>235.56668090820313</Width>  </GameObject>  <!-- Villian text box -->  <GameObject>  <Height>146.15000915527344</Height>  <LocX>68.149978637695312</LocX>  <LocY>205.71669006347656</LocY>  <PathToAsset>Props\GenericInteraction\TextBubble.png</PathToAsset>  <Width>219.5333251953125</Width>  </GameObject>  </GameObjects>  <InformationBoxes>  <entry>  <key>PLAYER</key>  <value>  <Height>46.866683959960938</Height>  <LocX>61.36663818359375</LocX>  <LocY>224.21665954589844</LocY>  <Width>84.48333740234375</Width>  </value>  </entry>  <entry>  <key>HERO</key>  <value>  <Height>51.183334350585938</Height>  <LocX>292</LocX>  <LocY>261.83332824707031</LocY>  <Width>207.816650390625</Width>  </value>  </entry>  <entry>  <key>VILLIAN</key>  <value>  <Height>107.91666436195374</Height>  <LocX>-0.30003738403320312</LocX>  <LocY>-1.4833300113677979</LocY>  <Width>772.0666389465332</Width>  </value>  </entry>  <entry>  <key>ALT1</key>  <value>  <Height>151.70001220703125</Height>  <LocX>9.5666418075561523</LocX>  <LocY>129.25</LocY>  <Width>333.00000858306885</Width>  </value>  </entry>  </InformationBoxes>  </value>  </entry>  </LocaleScreens>  </Locale> |

# Appendix F. Subject XML

Table 20 Subject XML Example

|  |
| --- |
| <Subject>  <IntroText>Algebra</IntroText>  <Subject>Algebra Adventures</Subject>  </Subject> |