

Architecture Specification

Isabel Giang edited this page on Apr 26 · 19 revisions

Architecture Specification

IntroOutroHandler

A handler which manages start screen and end screen user interaction and interface behavior for the Level application. Creates and updates Level's user interface for start and end behavior. Receives and manages all user input, connecting this input to other components such as `LevelRecipe` and `RecipeBox`.

Properties

Name	Type	Description
userClass	string	The Dungeons & Dragons class, stored from user input.
userLevel	number	The Dungeons & Dragons level, stored from user input.
done	boolean	True if level up process is complete. Otherwise, default state is false.

Functionality

Name	Parameters	Return	Behavior
displayDefault()	None	None	Renders webpage title "Level" in the top center of the screen. Renders webpage slogan "Updating D&D 5e Characters Made Easy" centered below the title.
displayChoices()	None	None	Retrieves list of available classes from <code>RecipeBox</code> and builds a dropdown menu from this list to display possible class choices. Builds a dropdown menu with corresponding choices for <code>number s</code> ranging from 1-20 to display possible level choices
levelUpStart()	None	None	If chosen <code>userClass</code> is contained in <code>RecipeBox.classList</code> and <code>userLevel</code> is a <code>number</code> from 1-20, retrieves a <code>LevelRecipe</code> from <code>RecipeBox</code> with the class and level that matches <code>userClass</code> and <code>userLevel</code> . Begins the level-up process when user clicks on "Let's Begin" button. Calls <code>endBehavior()</code> when the level up process is complete.

▼ Pages 7

- [Home](#)
- [Architecture Specification](#)
- [Design Specification](#)
- [Process](#)
- [Quality Prioritization](#)
- [Requirement Specification](#)
- [Verification Plan](#)

Clone this wiki locally

<https://github.com/I461-Bri>

Clone in Desktop

Name	Parameters	Return	Behavior
endBehavior()	None	None	Creates and displays the end-screen interface. Renders a congratulations message with chosen <code>userClass</code> and <code>userLevel</code> . (i.e, "Congratulations! You are now a Level 3 Bard.") Creates "Level Again" button.
resetUI()	None	None	When user clicks on "Level Again" button, returns interface to default state.

Connections

Inputs

- `RecipeBox` - Calls to retrieve an `Array<string>` of available class names
- `LevelRecipe` - Calls to retrieve `done` which indicates whether or not the level-up process is complete
- Event handler: `onclick` from user input - User indicates choice for `userClass` and `userLevel`

Outputs

- `LevelRecipe` - Calls to initialize level-up process

RecipeBox

Stores all `LevelRecipe` for every combination of levels and classes. Can return an `Array<string>` of classes available and a specific `LevelRecipe` available to use.

Properties

Name	Type	Description
<code>class</code>	<code>string</code>	The current Dungeons & Dragons class, provided by <code>InputOutputHandler</code> .
<code>level</code>	<code>number</code>	The current Dungeons & Dragons level, provided by <code>InputOutputHandler</code> .
<code>classList</code>	<code>Array<string></code>	A list of Dungeons & Dragons 5e classes. Represents the appropriate user input for classes.
<code>recipeList</code>	<code>Array<LevelRecipe></code>	A list of <code>LevelRecipe</code> , with a corresponding recipe for each supported class and level combination.

Functionality

Name	Parameters	Return	Behavior
<code>getClassList()</code>	None	<code>Array<string></code>	Returns an <code>Array<string></code> of Dungeons & Dragons 5e classes.
<code>getRecipe()</code>	<code>string</code> , <code>number</code>	<code>LevelRecipe</code>	Returns a <code>LevelRecipe</code> that corresponds to the given <code>string</code> class and <code>number</code> level.

Connections

Inputs

- `InputOutputHandler` - Provides the `class` and `level` chosen by the user.

Outputs

- `InputOutputHandler` - Returns a `LevelRecipe` that corresponds with the given `string` class and `number` level upon being called.
- `InputOutputHandler` - Returns a `Array<string>` of Dungeons & Dragons 5e classes that represent the classes that are supported by Level.

LevelRecipe

Stores all the `StepDescriptions`, preordered by Dungeons & Dragons rule specifications, necessary for a specified class and level. Manages iteration through `StepDescriptions` and provides notice to `InputOutputHandler` when level-up process is complete.

Properties

Name	Type	Description
<code>recipeClass</code>	<code>string</code>	The Dungeons & Dragons class.
<code>recipeLevel</code>	<code>number</code>	The Dungeons & Dragons level.
<code>done</code>	<code>boolean</code>	True if iteration through <code>StepDescription</code> is complete. Otherwise, set to false as default.
<code>hitPointInstruction</code>	<code>string</code>	Represents the hit point modifier. Varies depending on class. (i.e, For Bards, "1d8 + Constitution Modifier")
<code>stepList</code>	<code>Array<StepDescription></code>	A list that stores the appropriate <code>StepDescriptions</code> for <code>LevelRecipe</code> particular class and level. Ordered by Dungeons & Dragons rule specifications (i.e, Hit Points, Proficiency Bonus (if $\text{level} \% 4 = 1$), Class-specific characteristics, Features, Magic)

Functionality

Name	Parameters	Return	Description
<code>isDone()</code>	none	<code>boolean</code>	Returns true if level-up process is complete. Otherwise, returns false.
<code>stepInitialization()</code>	none	none	Iterates through <code>stepList</code> and initializes all stored <code>StepDescriptions</code> .
<code>exit()</code>	none	none	Exits out of <code>StepDescription</code> display and returns to <code>IntroOutroHandler</code> .

Connections

Inputs

- `InputOutputHandler` - Initialized by `InputOutputHandler` to begin iteration through the level-up process.
- `StepDescription` - Receives `boolean` indicating that user is finished with current `StepDescription` and is ready to iterate to the next `StepDescription`.

Outputs

- `StepDescription` - Provides current number in level-up process based on progress in iteration, (i.e, Step 5 if current `StepDescription` is 5th in storage), user class and user level for display. Initializes `StepDescription` in order to begin display of interface.

- `InputOutputHandler` - Returns `done` to `InputOutputHandler` upon being called to indicate that level-up process is complete.

StepDescription

Stores all relevant information (title, explanation of Step, choices) for a specific Step (i.e, Expertise, gained by Bards at Level 3 and Rogues at Level 6). Can be shared by multiple levels, multiple classes, and multiple class-level combinations.

Properties

Name	Type	Description
<code>stepClass</code>	<code>string</code>	The Dungeons & Dragons class.
<code>stepLevel</code>	<code>number</code>	The Dungeons & Dragons level.
<code>stepDone</code>	<code>boolean</code>	Default state is false. True if user indicates that Step display is no longer needed.
<code>title</code>	<code>string</code>	Name of specific Step. Unique identifier to each step. (i.e "Increase Spell Slots")
<code>explanation</code>	<code>string</code>	Natural language explanation of Step.
<code>choiceList</code>	<code>Array<Array<String>></code>	Default state is null unless Step has choices (certain Features). Will contain spell names if learning a new spell, or spell slots if increasing spell slots.
<code>spellSlots</code>	<code>Array<string></code>	Only present if <code>title</code> is "Learn New Spells & Cantrips". Stores a list of spell slots that corresponds with <code>StepDescription</code> 's <code>stepClass</code> and <code>stepLevel</code> .
<code>spellNames</code>	<code>Array<Array<string></code>	Only present if <code>title</code> is "Increase Spell Slots". Stores a list of spells that are mapped to every available Dungeons & Dragons level.

Functionality

Name	Parameters	Return	Description
<code>getStepClass()</code>	<code>none</code>	<code>string</code>	Returns <code>string</code> representing a Dungeons & Dragons class.
<code>getStepLevel()</code>	<code>none</code>	<code>number</code>	Returns <code>string</code> representing a Dungeons & Dragons class level.
<code>getStepDone()</code>	<code>none</code>	<code>boolean</code>	Returns true if user is no longer interested in viewing current Step. Otherwise, returns false.
<code>getTitle()</code>	<code>none</code>	<code>string</code>	Returns <code>string</code> representing title of current Step, which represents a unique identifier for each Step.
<code>getExplanation()</code>	<code>none</code>	<code>string</code>	Returns <code>string</code> representing explanation of current Step.

Name	Parameters	Return	Description
getChoiceList()	none	Array<Array<string>	Only implemented if relevant. (i.e, if Step is a Feature that has choices). Returns an Array<Array<string> of choices that the user can make for their character for the specific Step.
getSpellSlots()	none	Array<string>	Only implemented if relevant. (i.e, if Step's title is "Increase Spell Slots" or "Learn New Spells & Cantrips".) Returns an Array<string> of spell slots that correspond with StepDescription 's stored stepClass and stepLevel .
getSpellNames()	none	Array<Array<string	Only implemented if relevant. (i.e, if Step's title is "Learn New Spells & Cantrips".) Returns an Array<string> of spells that are mapped to all available Dungeons & Dragons levels.

Connections

Inputs

- LevelRecipe - Gets stepClass and stepLevel from LevelRecipe .
- SpellManager - if title is "Increase Spell Slots", retrieves Array<Array<string>> of spells slots that correspond with class and level.
- SpellManager - If title is "Learn New Spells & Cantrips", retrieves Array<Array<string>> of spell slots that corresponds with specific class and level. Also retrieves Array<Array<string>> of spell names for each available level.

Outputs

- LevelRecipe - Provides stepDone to LevelRecipe to indicate that user no longer needs this Step displayed.
- StepBuilder - Provides StepBuilder with title , explanation , stepClass , stepLevel and spellSlots and spellNames , if present.
- StepBuilder - Provides StepBuilder with done ` to indicate when user is done viewing current Step.

SpellManager

Stores information about Dungeons & Dragons spells - in particular, which spell slots are available to a character at a particular class and level and which spells can be used at a particular class' spel level.

Properties

Name	Type	Description
class	string	The Dungeons & Dragons class, passed in from StepDescription .
level	number	The Dungeons & Dragons level, passed in from StepDescription .

Name	Type	Description
classSpellSlotList	Array<Array<Array<string>	List which stores all of the spell slots for all Dungeons & Dragons levels and classes (of the classes that have spellcasting available)
classSpellList	Array<Array<Array<string>	List which stores all the names of the spells available to each level of each class (of the classes that have spellcasting available)

Functionality

Name	Parameter	Return	Description
getSpellSlotList()	string , number	<Array<Array<string>	Returns the list of spell slots that corresponds with the given class and level .
getSpellList()	string , number	<Array<Array<string>	Returns the list of spell names that corresponds with the given class and level .

Connections

Inputs

- `StepDescription` - User-chosen class and level are passed from `StepDescription`

Outputs

- `StepDescription` - Returns `Array<Array<string>` of spell slots that correspond with the given class and level when called.
- `StepDescription` - Returns `Array<Array<string>` of spell names that correspond with the given class and level when called.

