# Sample Rubric for IP.1 (Max Score: 91)

This is the sample rubric for HW1. I will accept any name the student comes up with as a name for this puzzle, as long as it is consistent and is not just “System”. It could be “Syllablast” or “ClassProject” or “Application” or “App” or whatever.

If you use different words across different use cases, then you will lose a point for each subsequent use. For example, if you use “App” in the first use case, but “Syllablast” in all other use cases, you will lose one point for each subsequent use case.

In general, each use case is graded as shown below (with minimum of 11 points).

**Grading Rubric: (11 points minimum but there can be more based on individual use case)**

+1 Participating Actor: Initiated by player (+1)

+1 Entry Condition: <each use case has its own valid points>

+1 Exit Criteria: <each use case has its own valid points>

+1 Flow of Events (+1 for numbered)

1. Player requests Something Valid (+1 Player, +1 Request)
2. **Syllablast** SomethingValid and refreshes Display (+1 App, +1 Valid, +1 Refresh)

Student gets +1 just for identifying the different parts (ParticipatingActor, EntryCondition, ExitCriteria, FlowOfEvents)

Note: Actor should always be **Player** (and not **User** or anything else) since in my original description that is the term I used. It should not be “actor” either since that is too vague.

Whether Entry Condition or Exit Condition (vv. For Criteria) it is ok.

**If a student makes a consistent mistake across multiple use cases, the most points to be lost is “-4”**

## Use Cases

The following use cases are identified for this application domain.

**UC 1: Choose Configuration (+15 points)**

**Participating Actor**: Initiated by **Player**

**Entry Condition**: None **(+1)**

**Exit Criteria**: Board initialized to chosen configuration **(+1)** andnumber of moves set to zero **(+1)** andscore is set to zero **(+1)**

**Flow of events:**

1. **Player** requests a configuration to play
2. **Syllablast** loads configuration and refreshes display

**UC 2: Swap Syllables (+14 points)**

**Participating Actor**: Initiated by **Player**

**Entry Condition**: Puzzle has not been completed **(+1)**

**Exit Criteria**: Chosen two syllables are exchanged **(+1)** and number of moves incremented **(+1)**

**Flow of events:**

1. **Player** requests to swap two syllables
2. **Syllablast** exchanges syllables and refreshes display

**UC 3: Undo Swap (+14 points)**

**Participating Actor**: Initiated by **Player**

**Entry Condition**: At least one swap has occurred **(+1)**

**Exit Criteria**: Most recent swap undone **(+1)** and number of moves decremented **(+1)**

**Flow of events:**

1. **Player** requests to undo a swap
2. **Syllablast** undoes swap and refreshes display

**UC 4: Reset Puzzle (+14 points)**

**Participating Actor**: Initiated by **Player**

**Entry Condition**: None

**Exit Criteria**: Board is reset to its initial state **(+1)**, the score is set to zero **(+1)**, and number of moves set to zero **(+1)**

**Flow of events:**

1. **Player** requests to reset puzzle
2. **Syllablast** resets board and refreshes display

**UC 5: Complete Puzzle (+16 points)**

**Participating Actor**: Initiated by **Player**

**Entry Condition**: Fourteen syllables are in their proper location **(+1)**

**Exit Criteria**: number of moves incremented **(+1)**, all syllables are in their proper location **(+1)**, and the score is 16 **(+1)**

**Flow of events:**

1. **Player** requests to swap final two syllables
2. **Syllablast** completes board,shows congratulatory message **(+1)**, and refreshes display

## Story Board (18 points)

Students must provide a graphic showing the essential layout of the application and provide sufficient documentation so you know how the player should interact with the application. A simplified GUI offers the player a limited number of ways to interact. I will always advise building a simple interface that is 100% functional—this is much better than trying to make an overly sophisticated and complicated interface that you can't get working. Here is a storyboard that aligns with the use cases above.

|  |  |
| --- | --- |
|  | **Points:**  (+2) Where board is drawn  (+2) Ability to choose three configurations  (+2) Show score  (+2) Show move counter  (+2) Ability to choose swap  (+2) Ability to undo swap  (+2) Ability to reset puzzle  (+2) Show where congratulatory message will appear upon completion  (+2) Must verbally explain or visually show how player selects syllables to be exchanged |

Given this GUI, consider how the user will interact with the system. The Game Window will show the board and syllables. The user uses mouse to select/deselect a syllable. No more than two syllables can be selected. Clicking on **Swap** does nothing if exactly two syllables are not selected. *It should be possible to only enable the swap button when two syllables are selected.* Once swapped, #moves increases and possibly the score is adjusted (could lose points)

After the game is completed, the Game Window is updated to congratulate the user.

With each move, the move counter is increased. Selecting **Reset** will reset the board to its original configuration (**Reset Puzzle**) and the score and move counter is set to zero. Choosing one of the three configurations will reset board to that initial state (**Choose Configuration**).

Note that coloring correct syllables is an optional part of this assignment.