# **Team Amalthea Cyber-poetry Slam Use Case document**

# 1. ProtectWord

**Actors**: Player

**Entry Condition**: Have at least one unprotected word

**Exit condition**: Protected words updated to include unprotected word **and** unprotected words updated to remove that word

**Flow OF Events:**

1. Player requests to protect word
2. System updates protected and unprotected words

# 2. ReleaseWord

**Actors:** Player

**Entry Condition**: Have at least one protected word

**Exit Condition**: Unprotected words updated to include protected word **and** protected words updated to remove that word

**Flow of Events**:

1. Player requests to release a protected word

2. System updates unprotected and protected words

# 3. PublishPoem

**Actor:** Player

**Entry Condition:** Have a poem

**Exit Criteria:** Poem is published on Wall **and** all words in that poem are released to unprotected area

**Flow of Events:**

1. The player selects to Publish Poem
2. System publishes poem and unprotects the words used in the poem

# 4. MoveProtectedWord

**Actors**: Player

**Entry condition**: Have at least one word in protected area

**Exit condition**: Word is moved to the location specified by the player without overlapping another word or poem

**Flow Of Events**:

1. Player selects to move a word to a different location
2. System updates protected area to reflect the move

# 5. MovePoem

**Actors**: Player

**Entry Condition**: Have at least one poem in protected area

**Exit Condition**: Poem is moved to the location specified by the player without overlapping another word or poem

**Flow Of Events**:

1. Player selects to move a poem to a different location
2. System updates the protected area to reflect the move

# 6. ConnectWord

**Actors**: Player

**Entry Condition**: Have at least two protected words

**Exit condition**: One poem consisting of the chosen word connected to the left or right of existing word or poem without overlapping another word or poem

**Flow Of Events**:

1. Player selects a protected word to connect to another word or existing poem
2. System updates protected area to show poem status

# 7. ConnectPoem

**Participating Actor:** Player

**Entry Condition:** Have at least two poems

**Exit Criteria:** One poem consisting of the chosen poem connected to the top or bottom of another existing poem without overlapping another word or poem

**Flow Of Events:**

1. Player selects a poem to connect to another existing poem
2. System updates protected area to reflect the poem’s status

# 8. DisconnectWord

**Actors**: Player

**Entry condition**: Have at least one poem

**Exit condition**: Word is disconnected and removed from poem without overlapping another word or poem in the protected area

**Flow Of Events**:

1. Player selects an edge word in the poem to disconnect
2. System updates the protected area to reflect the disconnected word and poem status

# 9. DisconnectRow

**Actors**: Player

**Entry Condition**: Have at least one poem with at least two rows

**Exit Condition**: Target row is disconnected and moved away from the existing poem without overlapping another word or poem

**Flow Of Events**:

1. Player selects a row to within a poem to disconnect
2. System updates the protected area to reflect the updated poems

10. ShiftRow

**Actors**: Player

**Entry Condition**: Have at least one poem which has at least two rows

**Exit condition**: Poem structure is unchanged but one row is shifted to the left or right without overlapping an existing word or poem

**Flow Of Events**:

1. Player selects a row from a poem to shift to the left or right
2. System updates protected area to reflect poem with the shifted row

# 11. UndoMove

**Actors:** Player

**Entry Condition:** A move has been made within the bounds of protected or unprotected area

Exit Criteria: The system reverts the protected and unprotected area to a state before the move was made

**Flow of Events:**

1. The player selects to “undo” their last move
2. The system returns to a state before the previous move was made

# 12. RedoMove

**Actors:** Player

**Entry Condition:** Player has undone at least one move

**Exit Criteria:** The system reverts to state before the Undo was performed

**Flow of Events:**

1. The player selects to “Redo” their last Undo
2. The system returns to a state before the previous Undo was made

# 13. ReleasePoem

**Actors**: Player

**Entry condition**: Have at least one poem in protected area

**Exit condition**: All words in poem are released to unprotected area

**Flow Of Events**:

1. Player selects to release poem
2. System updates protected and unprotected area to reflect all words in the poem being released

# 14. RequestSwap

**Actors**: Player

**Entry Condition**: Have at least one word in the unprotected area

**Exit Condition**: Swap requested

**Flow Of Events**:

1. Player requests a word swap
2. System presents Swap Request Submission screen
3. Player submits swap request
4. System requests broker for a match

# 15. RevokeSwap

**Actors:** Player

**Entry Condition:** Player has requested a swap that has not yet been carried out

**Exit Criteria:** There is no longer a swap being processed

**Flow of Events:**

1. The player selects to revoke a swap request
2. The system ceases to process the swap request

# 16. SearchUnprotected

**Actors**: Player

**Entry condition**: Have at least one word in unprotected area

**Exit condition**: Selected word is brought to the front of any overlapping words in unprotected area and the selected words are shown in the table

**Flow Of Events**:

1. Player enter the word(s) to search within the unprotected words
2. System searches the unprotected words for desired word(s) and displays the results to the user

# 17. SortUnprotected

**Actors**: Player

**Entry condition**: At least one word exists in the unprotected area

**Exit condition**: The table of words is sorted in the unprotected area

**Flow of Events**:

1. Player selects a way to sort unprotected words in the table
2. System updates the table to reflect the desired sorting

# 18. SelectWordFromTable

**Actors:** Player

**Entry Condition**: Have at least one word in unprotected area

**Exit Condition:** Selected word is brought to the front of any overlapping words in unprotected area

**Flow of Events:**

1. Player selects a word from the table
2. System updates unprotected area to bring the selected word to the front of any overlapping words