CEN4725/5726 Natural User Interaction

Course Project Final Paper

Group Members:
Carlson Joseph
Alec Shears
Jaeyong Sin
Munish Tanwar

Section: Undergraduate

Gator Path

1 Iterative Design and Development Process

1.1 Initial Design

Our initial design process revolved around the creation of an Alexa-based adventure game grounded in the use of story related voice commands. A player's voice inputs will be used to guide their interaction and progression through our narrative, which continues based on unique choices the player can make. This decision for players to use voice commands was made with the intent of achieving a deeper level of immersion into the game since it will both remove the need for any physical interaction (i.e. button pressing on a controller), and allow the player to feel part of the game since they are directly conversing with the narrator and making choices that alter their storyline. Initial features included in our first prototype were rudimentary and were selected primarily based on their utility to the player. These included options that allowed a player to know their position in the story, the ability to stop or restart the game, or to repeat the last voice prompt the game delivered to the player. Because this was our first time creating an adventure game with multiple branches for players to take, it was important for us to use storyboarding to keep track of the narrative as it grew and to efficiently make alterations where needed. Our prior combined experience of playing open-ended narratives ranging from HTML games to AAA titles did make it easier for us to anticipate the desires and feelings of our players, however, and this made it easier for us to know where our narrative may be stronger or fall flat with the average gamer.

1.2 Developing the First Prototype

Our development process of the first prototype was dependent on frequent group meetings in order to brainstorm ideas for the storyline, discuss limitations, and collaborate on writing code and setting up our environments and dependencies. The process of setting up the Alexa Skills Kit and getting acclimated to its various APIs was especially a challenge for our group, as we had not yet had much experience in the department of developing voice command projects. The collaboration was essential to this step because we all needed an adequate grasp of these functionalities in order to efficiently work together, especially when assigned story sections were separated. As mentioned earlier, we mainly utilized the Alexa Skills Kit's collection of APIs to ease the process of building the project. However, we also found great use in Twine, an

open-source tool that allowed us to easily develop our non-linear storyline and convert sections of it to HTML when needed. Collaboration through Twine was very important to the development of our story because it allowed us to maintain a cohesive narrative with our individual ideas working synergistically, rather than it being a patchwork of opposing ideas being thrown together.

1.3 Developing the Second Prototype

In our second prototype development process, we considered adding sound to the game and focused our attention on creating a more engaging experience for the users. Due to the COVID-19 outbreak, instead of meeting in-person, we had zoom call meetings to brainstorm ideas for additional storyline, discuss limitations and collaborate on adding sound features to the game. Since we were already familiar with developing on Alexa Developer Console, the process was smooth. Twine was used to converting our story into HTML when needed, and allow us to grasp how the story would be branched out into multiple paths. Communication was key because we needed to know who was working on what part, and what kind of progress we were making.

1.3.1 Change 1

a. Initially, there were some options that the users could say.
b. We added more varied options that users can say when selecting what they want to do in the story.
c. We got this idea from testing our game, we were running into recognition errors
d. We decided to include this change by adding different ways of saying an option to make the game recognize the users smoothly.

1.3.2 Change 2

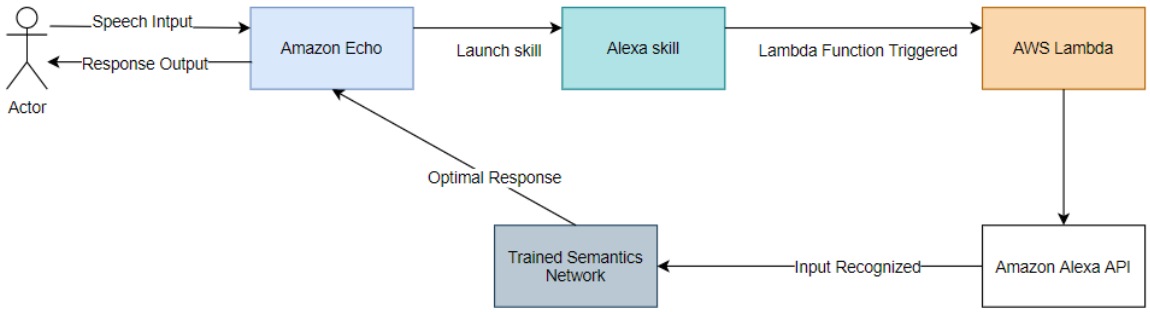
a. Initially, there were no sound effects in the game.
b. We added sound effects to key actions and events.
c. We got this idea from the peer review and suggestions from friends.
d. We decided to include this change by including an SSML tag in the code.

1.3.3 Change 3

a. Initially, all the characters had the default Alexa voice.
b. We added different voices and emotions for characters
c. We got this idea from the peer review and suggestions from friends
d. We decided to include this change by implementing different voices from Alexa voice library to give them more character and emotion.

2 Final Architecture

2.1 System Architecture



2.2 Code Modules

The functions used in the code:

- LaunchRequestHandler:
    - This function handles the launch intent which should include the name of the skill. This function also handles the startOverIntent. Upon recognizing the intent, the function initializes the speakOut variable to the output that welcomes the player to the game, sets the stages for them, and asks the player what street they would want to go. The output for this function is a speakOut variable that the Alexa prompts the user.
- StreetIntentHandler:
    - This function handles the intents about streets that correspond to the player's response to speakOut from LaunchRequestHandler. If the player chooses Thirteenth street, the boolean variable thirteenthStreet is set to true, and prompt

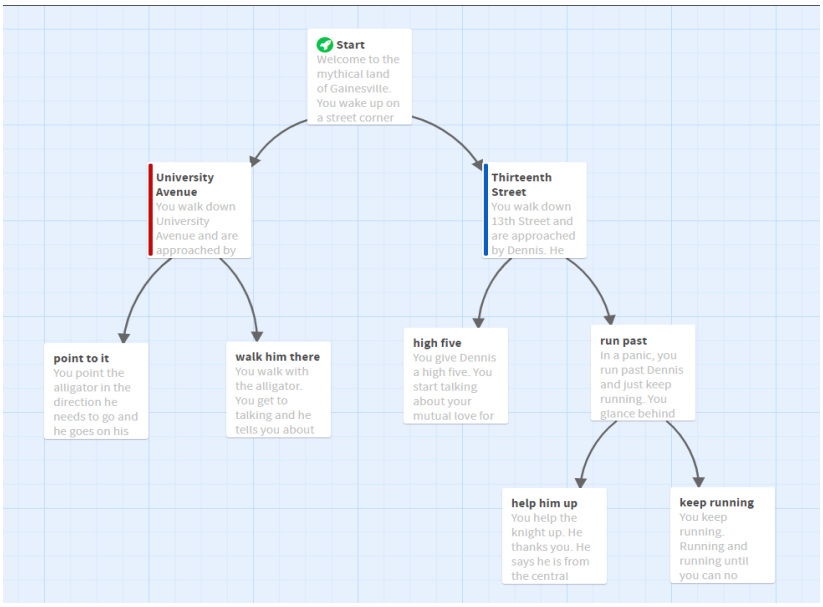
further instruction about Denis, an non-player character. If the player chooses University Street, the boolean variable universityAvenue is set to true and prompts the player further instruction about Alligator, another non-player character. The output for this function is a speakOut variable that the Alexa prompts the user.

- AlligatorIntentHandler:
    - This function handles the intents about Alligator. This function is triggered if the universityAveune boolean is true and thirteenthStreet boolean is false. Based on the player's input, a boolean variable that corresponds to that input is set to true, and speakOut variable is set to the output response player should get. The user is prompted with speakOut and prompt is get the player's responses.
- DennisIntentHandler:
    - This function handles the intents about Dennis. This function is triggered if the thirteenthStreet boolean is true and universityAvenue is false. Based on the player's choice and response, a boolean variable that corresponds to that choice is set to true, and speakOut is set to the output response player should get. The user is prompted to speak Out and re-prompted for the response.
- KnightIntentHandler:
    - This function is similar to DennisIntent function. This function is triggered if the player chooses to continue with the story without Dennis and encounter another non-player character, Knight. Based on the player's response and choices, a boolean variable that corresponds to that choice is set to true, and speakOut is set to the output response the player should get. The user is prompted to speak Out and re-prompted for the response.
- WhereAmIIntentHandler
    - This function is triggered when the user responds with "where am I" which is part of the whereAmIIntent. The function outputs the current location in the game. The output is based on the boolean values which are set during other intent handler functions.
- HelpIntentHandler

    - This function is a default function provided by Amazon. This function handles built-in intent that assists the user.
- CancelAndStopIntentHandler:
    - This function is a default built-in function provided by Amazon. This function handles StopIntent or CancelIntent. The users are prompted with a farewell response.
- SessionEndedRequestHandler:
    - This function is the default built-in function provided by Amazon. Nothing was altered with this function.
- IntentReflectorHandler
    - This function is the default built-in function provided by Amazon. This function is used for interaction model testing and debugging. It repeats what used just said.
- ErrorHandler
    - This function is the default built-in function provided by Amazon and it is used for error handling.
- All the intent handler functions and helper functions are exported as Alexa.SkillBuilders.

2.3 Third-Party Tools

- Twine
    - This is an open-source tool for building non-linear stories.
    - https://twinery.org/



2.4 Source Code

- https://github.com/jackeysin/Gator-Path

3 Future Work

In the future we plan to add more features to make the game more dynamic. We have talked about implementing some sort of combat system in the game whether that be a simple dice roll, or an approach where the user can choose between multiple moves. Either way this could make the game more fun to play, and it will add another element to the gameplay. Another feature we talked about was linking our game to the Amazon database system dynamo DB. This would allow the state of the game to be saved so that the user can return to the game at any time and continue. Another interesting feature would be creating an inventory system for the user, where the user can store and use items. I think our game is fully playable as of right now, but adding some of these features would make the game more immersive and enjoyable.