॥ त्वं ज्ञानमयो विज्ञानमयोऽसि ॥

Interactive Game Development for Language Literacy

Contributor(s):Aatif Ahmad (B.Tech. AiDE, 2nd Year),
Anushk Gupta (B.Tech. AiDE, 2nd Year)
Supervisor(s): Dr. Rachel Philip (Assistant Professor, SoLA),
Dr. Riby Abharam Boby (Assistant Professor, ME)

Semester-1, AY 2023-2024

Abstract

In our B.Tech third-semester project, Anushk Gupta and Aatif Ahmad, under the guidance of Dr. Riby Abraham Boby and Dr. Rachel Philip, developed an interactive game to boost language skills. Our aim was to create a fun and accessible way for learning.

We delved into existing language games, crafting a system that's not only user-friendly but also easily expandable for future improvements. Our contribution includes modular code snippets with comments, and the depth sensor integration, facilitating collaboration for further development.

The real-world testing phase involved gathering valuable data through participant feedback and performance metrics. Overcoming challenges during testing allowed us to refine the game for better results.

The outcomes reveal a positive impact on language skills, affirming the effectiveness of our developed game. This abstract provides a balanced overview of our project's goals, methods, and key findings within the realm of educational technology.

Division of Work and Contribution

Aatif Ahmad:

- Implemented the A and T modules of the game, adding significant depth to interactive features.
- Integrated voice recognition into the S module, enhancing the overall user experience.
- Added code to read player timings and generated insightful statistical graphs using matplotlib.
- Enhanced the project's online presence with a comprehensive project description and a contributors page.

Anushk Gupta:

- Crafted the opening slide and the S module of the game, contributing to its engaging design.
- Introduced the time module to enable tracking of player timings.
- Designed the landing page for the project website, showcasing creative skills.
- Developed the points collection system after each slide of a module, enhancing user engagement.

Acknowledgment

Our sincere gratitude extends to all those who played a pivotal role in the successful completion of our project, "Interactive Game Development for Language Literacy."

Special acknowledgement to Arun V B and Barun Mahaldar, Digital Humanities students at IIT Jodhpur, who designed the game modules, adding significant value to the project.

Dhruv Agrawal, B.Tech Final year student at IIT Jodhpur, joined the team during the final phase, contributing to coding for the sensor and developing the coordinate shifting code, ensuring seamless functionality of the game.

Our sincere thanks to Kendriya Vidyalaya, IIT Jodhpur, for granting us permission to conduct field testing with their students from first to fifth grade. The collaboration provided invaluable real-world feedback and insights into the game's effectiveness in an educational setting.

We express our deepest gratitude to our mentors, Dr. Riby Abraham Boby and Dr. Rachel Philip, for their unwavering support, guidance, and expertise throughout the project. Their insights and constructive feedback were instrumental in shaping the project's trajectory.

Each member's unique contribution has played a crucial role in the project's success, and we acknowledge and appreciate their efforts.

Table Of Contents

1. Introduction

1.1 Background

- Sakshar is a game based learning methodology initially developed at Indian Institute of Technology, Delhi in association with Waseda University, Japan. It was originally developed with the initiative to help stroke patients but then gradually progressed to make learning fun for students.

- There have been several versions of the game like the KINECT version which is a cost-effective solution and the latest Intel Real-Sense Camera based version.

- Our effort has been in the direction of creating a similar game with updated content and some more features added alongside creating analytics for the player's performance so that a customized learning path could be developed for the individual players.

1.2 Objectives and Goals

- The primary objective is to create an updated version of the game in Python programming language instead of C# which was the earlier language of the game.

- The selection of python as the language of choice for the game was based on the fact that python has several relevant libraries and modules like the pygame, mixer, matplotlib, numpy which could be used to make interactive games and track the player's performance data.

- The goal of this project is to let kids learn words based on phonetics in order to ensure a holistic learning approach for the kids.

- It is also emphasized to include different activities in the game, unlike simply selecting words of a particular character to give kids an enjoyable experience of the game.

- The target audience of the game are mostly the kids under 12 years of age across Rajasthan's demography covering different ethnicities and cultural backgrounds.

2. Methodology

2.1 Project Approach

- At first we reviewed the originally developed game to get ideas for our own version of the game.

- We received slides for the different modules of the game which were skillfully created by Digital Humanities students who were cooperating with us on the project.

- They gave us the slides of the S,A and T modules of the game one after the other and then we collected voiceovers from https://voicemaker.in/ , other supporting images from https://www.freepik.com/. These resources were linked with our code during the development of the game.

2.2 Development Methods

- We started with python's pygame library and created the window of the game which spans over 1000px by 700px. Next, we added the landing page of the game and added cheerful music as the background taken from https://www.chosic.com/ .

- The beginning page of the S module was later added. It was then followed by multiple scenes of the S module where a player has to select words starting from letter 'S' of the English alphabet. When the beginning page pops up, a voiceover starts, which instructs the player to choose the words starting with 'S' in both languages, namely English and Hindi.

- The game also includes slides where players have to count objects appearing on the screen. It was added to get rid of the monotonous approach of the former version of the game which was based only on selecting objects.

- We have also added a speech recognition slide where a player is instructed to count some objects on the screen and respond accordingly. It is created with Python's Speech Recognition Library. We checked using multiple words and concluded that the system needs at least two words to detect clearly what the player intends to say. Based on this

observation, we added several options which a player might speak in order to respond to the question on the slide. It is to be noted that the question is not written on the slide, rather it is asked as a voiceover in both English and Hindi languages to test the player's listening skills.

2.3 Game Architecture

- Every module starts with a beginning page which displays a letter of the English alphabet and a voiceover with an instruction.

- The players then move to several scenes of the game by clicking on ‘NEXT’ buttons which are present on multiple slides of the game.

- After successfully completing a slide, a victory image is popped up which shows the scores of the player. Apart from scores, we have added stars on every victory image to give players a playful experience.

- This template is followed for all the letters namely ‘S’, ‘A’ and ‘T’ which have been developed so far. The emphasis has been placed on making students well versed with multiple words of selected phonetic sounds.

- The player is also given an option of moving from one module to the other by pressing on the letter buttons constantly displayed on the right side of the screen. The right palette also hosts a ‘END GAME’ button which can be pressed anytime to clear the game.

- The game ends when player reaches a score of 12 covering 5 slides of ‘S’ module, 4 slides of ‘A’ module and 3 slides of ‘T’ module.

- At every slide of the game, code has been added using python’s time module to track the player’s timings for the analytics. When the game ends, all the analytics are saved in an ‘analytics.txt’ file. This is managed through file handling.

- After the analytics file is updated, we have also created a ‘data.py’ file using python’s matplotlib library which reads the data from ‘analytics.txt’ file and adds name of the player from a ‘players.txt’ file which is concurrently maintained by adding name of the player once he/she has finished playing the game.

- Once the ‘data.py’ file is run, it saves the game statistics of all the players who have played the game and whose names have been entered in the ‘players.txt’ file. The statistics include timings of the player on the slides of the game in the form of bar graphs whose heights represent time in seconds.

- After all the graphs are created, they are clubbed together in a pdf file which is uploaded on the analytics website of the project which could be accessed by : https://aatifxarchie.github.io/saksharanalytics/index.html.

- The website is created using HTML, CSS and uploaded using Github Pages.

- The website has a landing page from where one could go to the description page of the project, contributors page and the analytics page which has a downloadable link of the statistics pdf.

---

3. Implementation

The following are the code snippets along with descriptions used in the game development:

1. The modules needed for the game along with creation of the pygame window.

```
import pygame
from pygame import mixer
import time
import pygame.font
import speech_recognition as sr
from pygame.locals import *

start_time = pygame.time.get_ticks()

#Initialization
pygame.init()

#Screen creation
screen = pygame.display.set_mode((1200,700))

#landing page image
land_image = pygame.image.load("land_image.jpg")

'''Background sound'''
mixer.music.load("smilee.mp3")
mixer.music.play(-1)
```

2. This is an example of the implementation of a slide of the game. We have created several buttons of nearly the same size as the object. They have been placed on the lowermost layer. Using collision technique, when a player happens to click on a button, a colorful image is blitted on the topmost layer thus illusioning the player as if the image just got coloured.

```
if night_scene_image_status == 1:
    if time_1_2_1_status == 1:
        time_1_2_1 = pygame.time.get_ticks()
        time_1_2_1_status = 0
    sun_image_status = 2
    screen.blit(night_scene_image,(0,0))
    input_8 = my_button_6.check_click()
    if input_8:
        stone_image_status = 2

    input_9 = my_button_5.check_click()
    if input_9:
        swing_image_status = 2

    input_10 = my_button_7.check_click()
    if input_10:
        spider_image_status = 2

    input_11 = my_button_8.check_click()
    if input_11:
        stars_image_status = 2

if stone_image_status == 2:
    screen.blit(stone_image,(8,527))
    screen.blit(stone_text,(280,600))
    if night_stones_sound_status == 1:
        night_stones_sound.play()
        night_stones_sound_status = 0
```

3. Saving all the recorded timings in the analytics file.

```
with open('analytics.txt','a') as file:
    file.write(str(time_1_1) + " " + str(time_1_2) +
               " " +str(time_1_3) + " " +str(time_1_4) + " "
               +str(time_2_1) + " "
               + str(time_2_2) + " " +str(time_2_3)
               + " " +str(time_2_4) + " " + str(time_3_1)+
               " " + str(time_3_2)+ " " + str(time_3_3)+ " "
               + str(total_game_time) + "\n")
```

4. This is the analytics file that stores the player's timings.

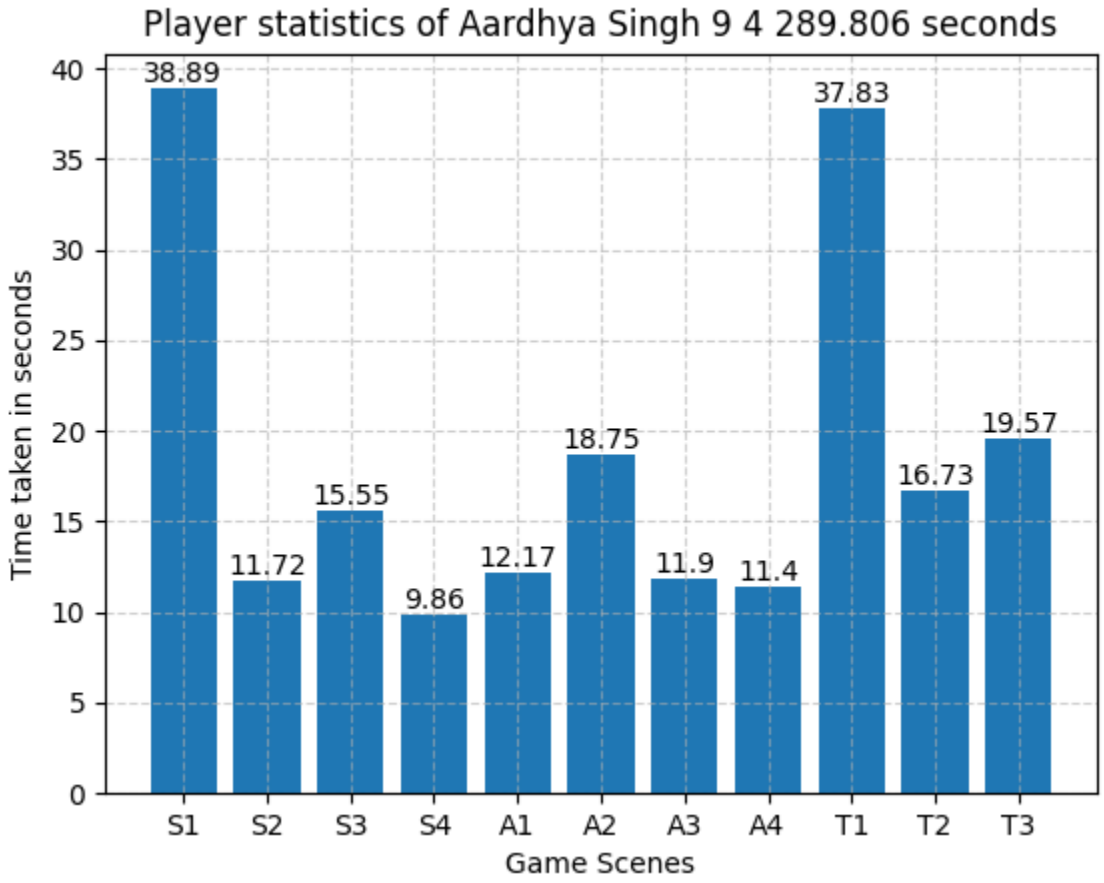
```
45798 14462 13594 8131 35999 14351 42169 12730 36871 12625 31978 419429
11385 18388 23013 6490 8892 26095 8747 17429 22149 12907 23421 243817
42020 17766 18942 7763 9189 16603 12776 12426 23955 9066 11750 275096
55797 40055 19933 19829 17000 53740 10642 11169 22532 14125 19135 350619
21585 12188 14175 6008 7721 17332 8456 19816 15014 15483 23551 254597
58618 21505 43193 9064 15913 16299 12748 19052 18080 13809 27466 357332
20562 15709 12398 6887 10024 17025 11938 18493 23054 19682 23559 264414
38886 11721 15546 9860 12167 18746 11896 11403 37834 16729 19573 289806
57543 42380 49049 28648 35438 63578 13886 14713 105987 15909 50504 690566
79044 43991 26795 6470 58715 131400 14134 26704 99971 24554 83529 740585
63413 24504 16707 17430 0 0 0 0 0 0 0 107054
0 0 0 0 22275 48404 21797 55656 50748 29966 55004 284350
20562 15709 12398 6887 0 0 0 0 0 0 0 56556
0 0 0 0 10024 17025 11938 18493 0 0 0 58480
0 0 0 0 0 0 0 0 23054 19682 23559 67745
11385 18388 23013 6490 8892 26095 8747 17429 22149 12907 23421 243817
55797 40055 19933 19829 17000 53740 10642 11169 22532 14125 19135 350619
```

5. This is the ‘players.txt’ file that stores players information in the format: name-class-age.

6. This is the implementation of the ‘data.py’ file that creates bar graphs and then saves them as png images.

```python
import matplotlib.pyplot as plt

with open('players.txt', 'r') as players_file:
    player_names = [line.strip() for line in players_file]

with open('analytics.txt', 'r') as file:
    lines = file.readlines()

for player_index, line in enumerate(lines):
    numbers_as_strings = line.split()
    numbers = [float(num) for num in numbers_as_strings]
    times = [float(num / 1000) for num in numbers]
    times_new = times[:11]

    scenes = ['S1', 'S2', 'S3', 'S4', 'A1', 'A2', 'A3', 'A4', 'T1', 'T2', 'T3']

    bars = plt.bar(scenes, times_new)

    plt.xlabel('Game Scenes')
    plt.ylabel('Time taken in seconds')
    player_name = player_names[player_index] if player_index < len(player_names)
    aggregate_time = times[11]
    plt.title(f'Player statistics of {player_name} {aggregate_time} seconds')
```

7. This is an example image of a player's timings of the game slides.



8. HTML code snippet written for the home page of the website.

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>SAKSHAR</title>
    <link rel="stylesheet" type="text/css" href="styleindex.css">
</head>
<body>
    <header>
    <h1>SAKSHAR ANALYTICS</h1>
    </header>
    <main class="commands">
        <p><a href="project.html">About the Project</a></p>
        <hr>
        <p><a href="contributors.html">Contributors</a></p>
        <hr>
        <p><a href="analytics.html">Latest Analytics Data</a></p>
        <hr>
    </main>
</body>
</html>
```

9. Styling the website using CSS.

```css
a {
    text-decoration: none;
    color: aliceblue;
}

a:hover {
    color: rgb(100, 210, 237)
}

/* Style the contributors section */
.contributors {
    text-align: center;
}

/* Style the heading */
.contributors h2 {
    font-size: 28px;
    font-family: Cambria, Cochin, Georgia, Times, 'Times New Roman', serif;
    letter-spacing: 0.1rem;
    color: aliceblue;
}

/* Style each contributor card */
.contributor {
    display: inline-block;
    width: 250px;
    margin: 20px;
    background-color: rgb(2, 42, 58);
    border-radius: 8px;
    box-shadow: 0 0 10px rgba(235, 234, 234, 0.1);
    transition: transform 0.2s;
}
```

10. Complete display of the setup.

SAKSHAR whole set-up at MECHATRONICS lab IIT Delhi

All the supporting files for the project have been pushed in a Github repository which can be accessed from https://github.com/aatifxarchie/saksharproject2023.

The code for the website creation can be accessed from https://github.com/aatifxarchie/saksharanalytics.

---

4. Results and Evaluation

4.1 Outcome of the Project

The field testing phase of "Interactive Game Development for Language Literacy" unveiled crucial insights that will guide the project's future development and improvements.



4.2 Field Testing

4.2.1 Testing Site: Kendriya Vidyalaya, IIT Jodhpur

The game underwent extensive testing at Kendriya Vidyalaya, IIT Jodhpur, on Sunday, 19th November, 2023 with students ranging from first to fifth grade. This testing site provided a diverse and valuable demographic for assessing the game's effectiveness.

4.2.2. Participant Feedback and Performance Metric

Participants' feedback was instrumental in understanding the strengths and weaknesses of the game. Performance metrics, including completion times per game page and the total time taken to complete the game, were collected to quantitatively assess the game's impact.

4.3 Data Collection

Player timings were meticulously recorded to analyze the duration of engagement and identify potential patterns. Statistical graphs, generated using matplotlib, provided visual representations of player progress and performance over time.

Link for the generated graphs:
https://aatifxarchie.github.io/saksharanalytics/index.html.

4.4 Key Findings

- *Game Content Enhancement*: Participants expressed a desire for increased game content or content shuffling to enhance the overall experience and educational outcomes.
- *Word Familiarity and User Progress*: An interesting trend emerged as successive players demonstrated increased familiarity with words and objects, indicating positive learning progression
- *Projector Overheating Issue*: A technical challenge surfaced with the projector overheating after prolonged use.
- User Difficulties and Challenges: Participants faced challenges related to visual difficulty with small-sized objects and word familiarity.

To develop further on the current findings, recommendations have been made in the section 6.0 of the report.

---

5. Future Development

Based on the findings of the Field testing at Kendriya Vidyalaya, here are the recommendations for further development of the game:

- Expand game content to cater to user preferences and improve engagement.
- Investigate and address the projector overheating issue to ensure prolonged usage without disruptions.
- Implement adjustments to mitigate visual difficulties related to small-sized objects.
- Consider incorporating adaptive features to accommodate varying levels of word familiarity among users.

---

6. Conclusion

The development and testing of "Interactive Game Development for Language Literacy" have provided valuable insights. Key learnings include the positive impact on *user progression* and the identification of *technical and user-centric challenges.*

Steps Ahead:
The outlined recommendations will guide future development, focusing on addressing challenges and enhancing user experience. The *iterative nature of this project* ensures ongoing improvement to meet diverse user needs.

Final Thoughts:
As we navigate the dynamic landscape of educational technology, "Interactive Game Development for Language Literacy" remains committed to engaging, inclusive, and impactful learning experiences.

---

7. Resources Used

---

1. Integrated Development Environment - VS Code.
2. Github
3. Python
4. Pygame Library
5. Speech Recognition Library
6. Matplotlib Library
7. Numpy Library
8. https://voicemaker.in/
9. https://www.freepik.com/
10. https://www.chosic.com/free-music/games/
11. https://mp3cut.net/change-volume
12. https://www.speakatoo.com/
13. https://www.resizepixel.com/edit
