NOVEMBER 2021

# ANOMALOUS

## A NOSTALGIC ADVENTURE GAME

WITH A TWIST

PREPARED BY MICHELLE GORDON

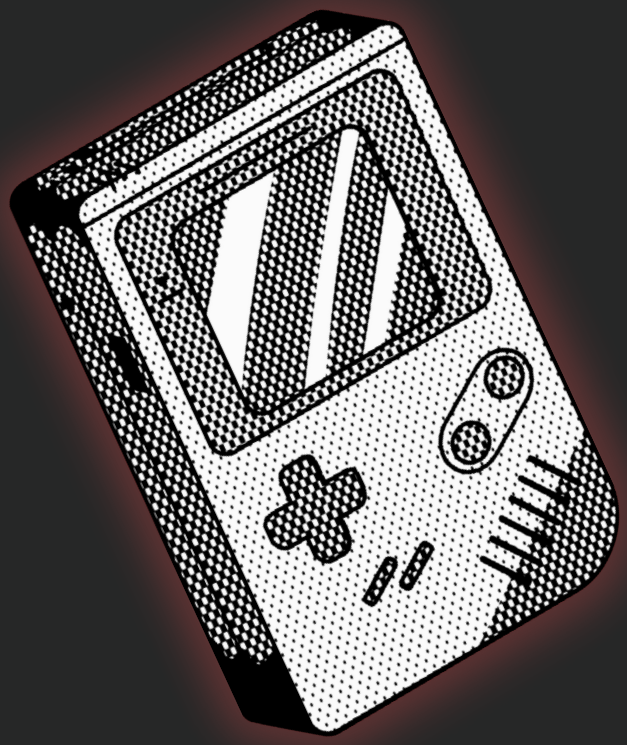
CSIS 2420 - FALL 2021

# DESCRIPTION

Anomalous is an interactive fiction game where the player moves through a spell-binding story by making decisions at each pivot point of the plot. The aesthetic feel of interface will be a throw-back to a nostalgic computing era where the World Wide Web was still new, and ads didn't plaster every corner of the internet.

The gameplay of Anomalous will begin with the user being asked for their name.Upon inputting their name, Anomalous will auto-generate a 90's themed username that the user will be referred to throughout the game. The player will receive the story a piece at a time and then will be presented with four choices. Whatever the player chooses to do will affect how the story plays out and determine their final scores. This is where the twist comes in.

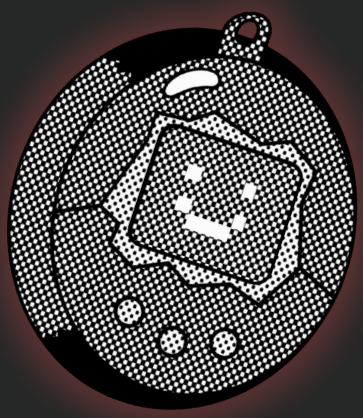
Upon completing the game, the player will be presented with their final scores and the true purpose of Anomalous will be revealed. The scores that are tallied throughout gameplay are actually scores for personality traits that match a member of the Belcher family from the animated sitcom, "Bob's Burgers." Finally, the user will be presented with a report stating which family member they are most like.



# REQUIREMENTS

## FEATURES/FUNCTIONALITY:

- Username generator
  - Prompt the player to enter their name and then look up the ASCII character numeric value of the first three letters their name. Use the three numbers to randomly choose words for the three portions of the username that will be formatted as follows: word1+word2+number. An example of a username that would be generated is: kittyluvr92.
- Evolving Avatar
  - Prompt player to choose from a small list of avatars.As they progress through the game, the avatar's expression will change in accordance with the player's evolving scores.
- Changing Storyline
  - There will be four versions (tracks) of the story that are aligned with one of the four different personality traits. As the player makes choices throughout the game, their changing scores will inform the program to stay on the same track or jump to a different track while traversing forward through the story.
- Decision-Making
  - There will be four buttons below the story text that contain the different actions available to choose from.
- Scoring
  - Keep track of four scores (one for each personality trait) throughout the game.
- Final Report
  - At the end of the game, present user with a final scores report which includes:
    - Final score of each personality trait
    - Information about which Belcher they are most like and why
    - A list of other players and their final scores



# REQUIREMENTS CONTINUED...

## TECHNOLOGY:

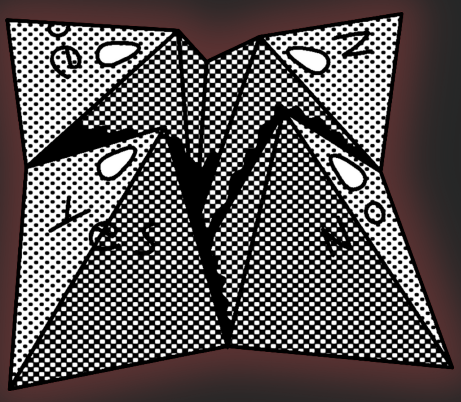
- Programming Language: Java
- Libraries used: Java's Scanner API, WindowBuilder
- IDE: Eclipse
- Data Management: MySQL

## DATA STORED:

- Player's name
- Player's username
- Player's personality scores (four individual scores)
- Historical Player data:
    - All of the above data for each past player along with the date they played the game.

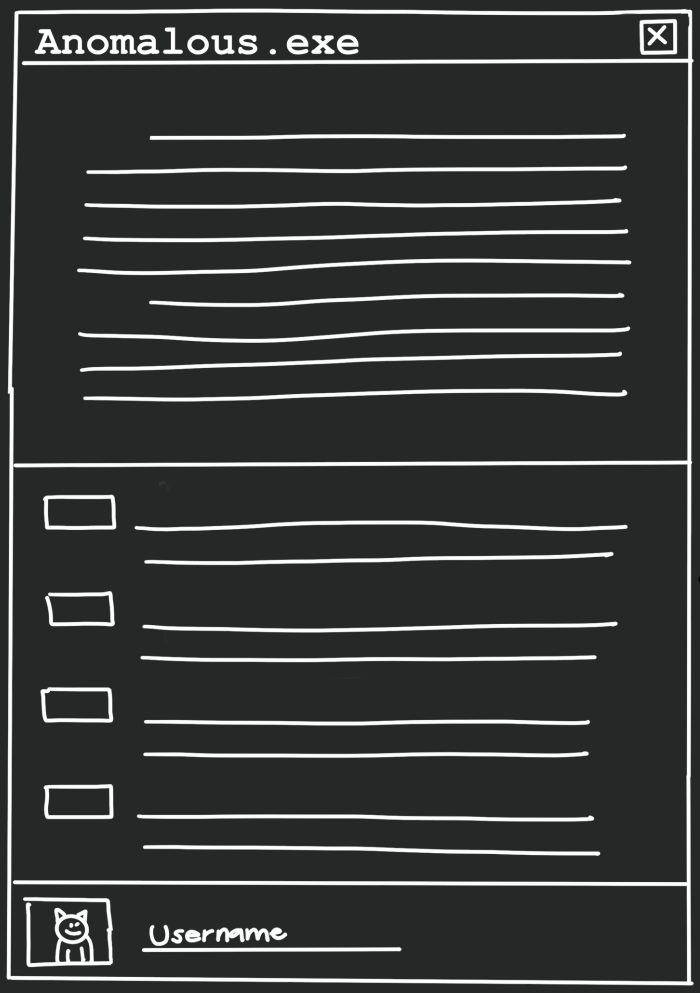
## INTERFACE/GUI:

- Center panel that holds the story's text
- Player's info - avatar and username
- Four buttons at the bottom
    - each contains an action to take in response to the portion of the story in the center panel.
- A "quit" button



# THE INTERFACE

## A NOD TO CLASSIC SOFTWARE FROM THE 90'S

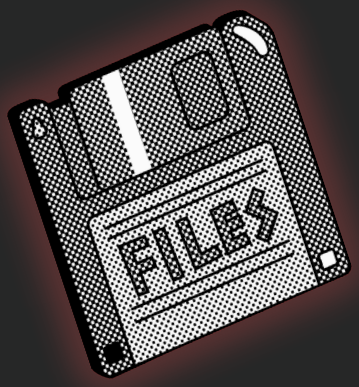


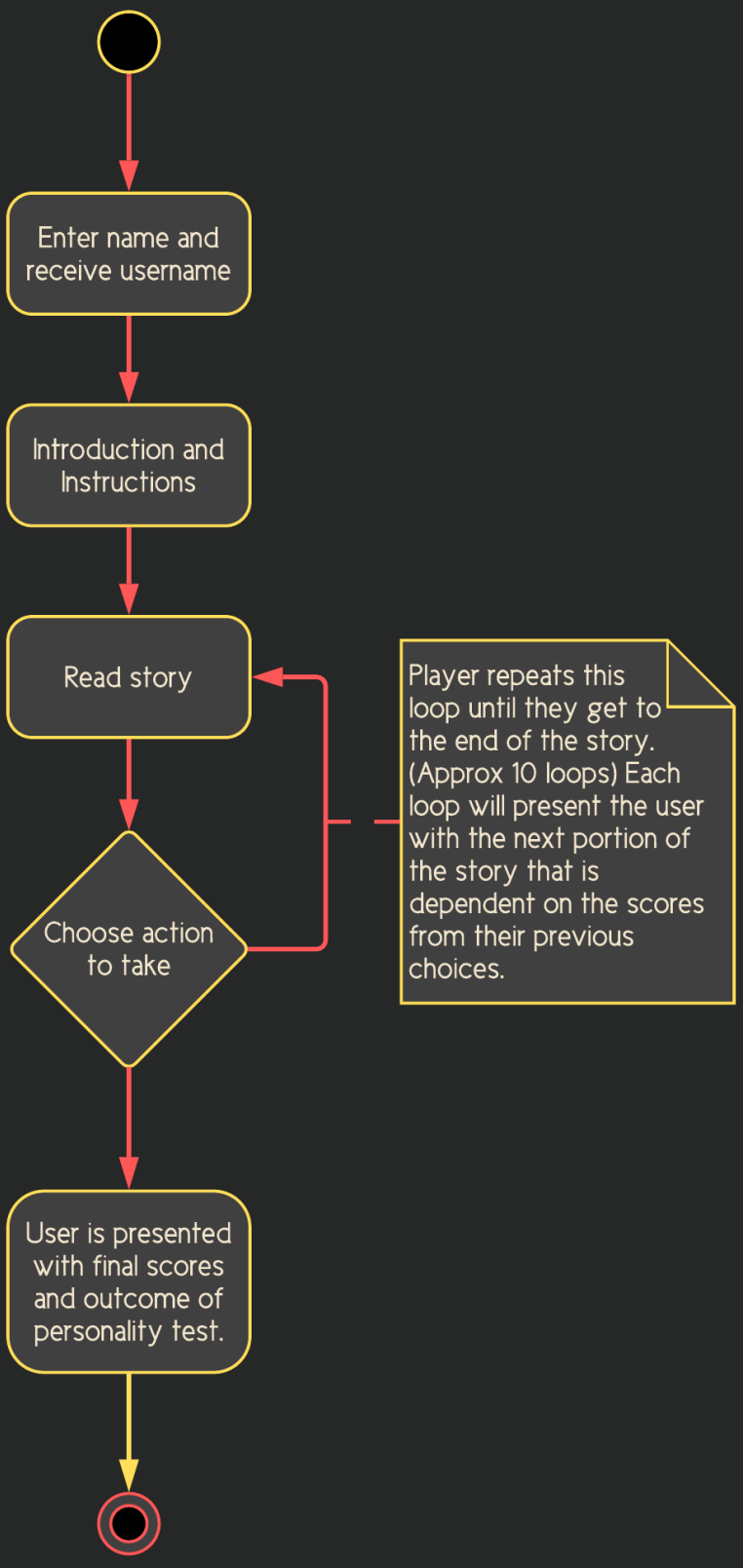

The interface of Anomalous features four distinctive sections:

- The title bar with the name of the game and the "X" button to exit the game at any time. It will be created in the style of a Windows95 window.
- The text for the current portion of the storyline. There will be roughly one to two paragraphs. Just enough to keep the story moving and present the player with a choice to make.
- The first part of the player's dashboard. It will feature four buttons with the description of an action written next to it. The player chooses how to proceed by clicking on one of the buttons.
- The second part of the player's dashboard. It will house the player's evolving avatar as well as their username.

# UML

## ACTIVITY DIAGRAM



# TESTING

## DESCRIPTION OF OUR PROCESS

01

**Performance Testing**
Anomalous' performance metrics will be tested to make sure it does not lag long enough to entice a user to quit or get distracted by their phone (as much).

02

**Validation Testing**
We want to avoid the old adage of "Garbage in, garbage out" by using validation testing to make sure only valid data types will be accepted when the user enters their name.

03

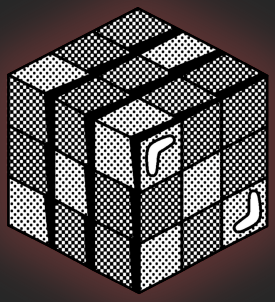
**Unit Testing**
A test-suite will be created in JUnit for unit testing Anomalous' logic - we want the complexities of the scoring effects to work correctly to preserve the integrity of the outcome.

## METRICS USED

As part of the performance testing, two metrics will be used. Time will be captured to test the length of time it takes to run the automatic username generator method. Additionally, tests will be run to capture the number of nodes the program traverses through when searching for a piece of the username, or the next portion of the story in accordance with the player's current scores. This is to ensure that the most efficient algorithms are being used throughout the program.

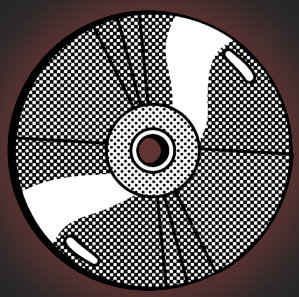
# DATA STRUCTURES AND ALGORITHMS

## A DESCRIPTION OF THE DIFFERENT DATA STRUCTURES AND ALGORITHMS USED TO MANAGE DATA AND FACILITATE PROCESSES



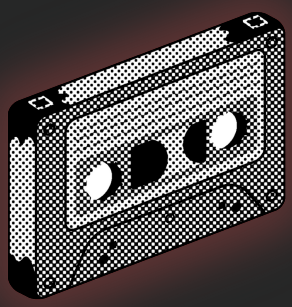
### HASHING

Much like encryption methods that are used to obfuscate a password or other type of secure data, a hashing algorithm will be used to automatically generate a username based off of the player's name.



### NODES

Nodes will be used to store each portion of the story as well as possible actions the player can take, in addition to the point values for each of those actions. They will function as cells of the story.



### LINKED LISTS

An array of linked lists will house the nodes in different "tracks" to keep each version of the storyline separate and organized. This will make hopping from one track to the next easier to manage.