



# Penguine

Lance Townsend, Carla Kolze,  
Carson Gustavel, Amanda Board, Patrick Du,  
Owen McDaniel, Emily Smith

# Doki Doki Library Club: Revolutionary Dating Sim

- You are stuck in a library with Sonic, Charlie Brown, and Scooby Doo characters
- There is only one way out of this nightmarish library: Dating
- You must successfully woo one of the various characters in the game to leave
- If you are unsuccessful in your advances on ALL characters, you are stuck there forever and lose
- Dating involves talking to the characters and playing different mini games
- Library is located in Miami, FL.



Penguine



# Story Board



Player enters game and gets a start menu. Player can enter their username that LI's will address them by.

The start menu will also give a brief overview of the game.



Penguine

# Story Board

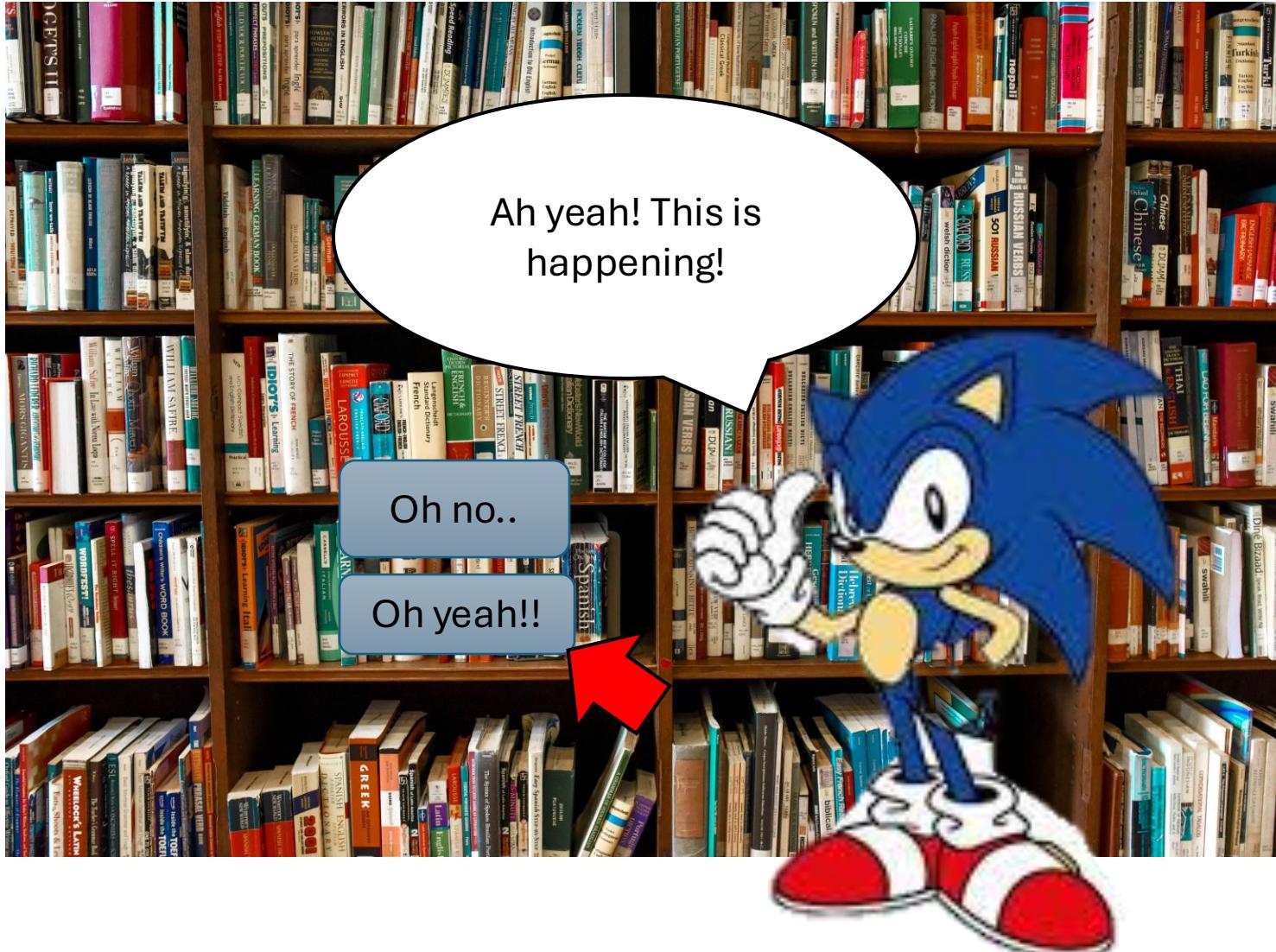


Player enters the library and can choose who they talk to. In this example, player chooses to talk to Sonic.

Carla



# Story Board



Player talks to Sonic LI.

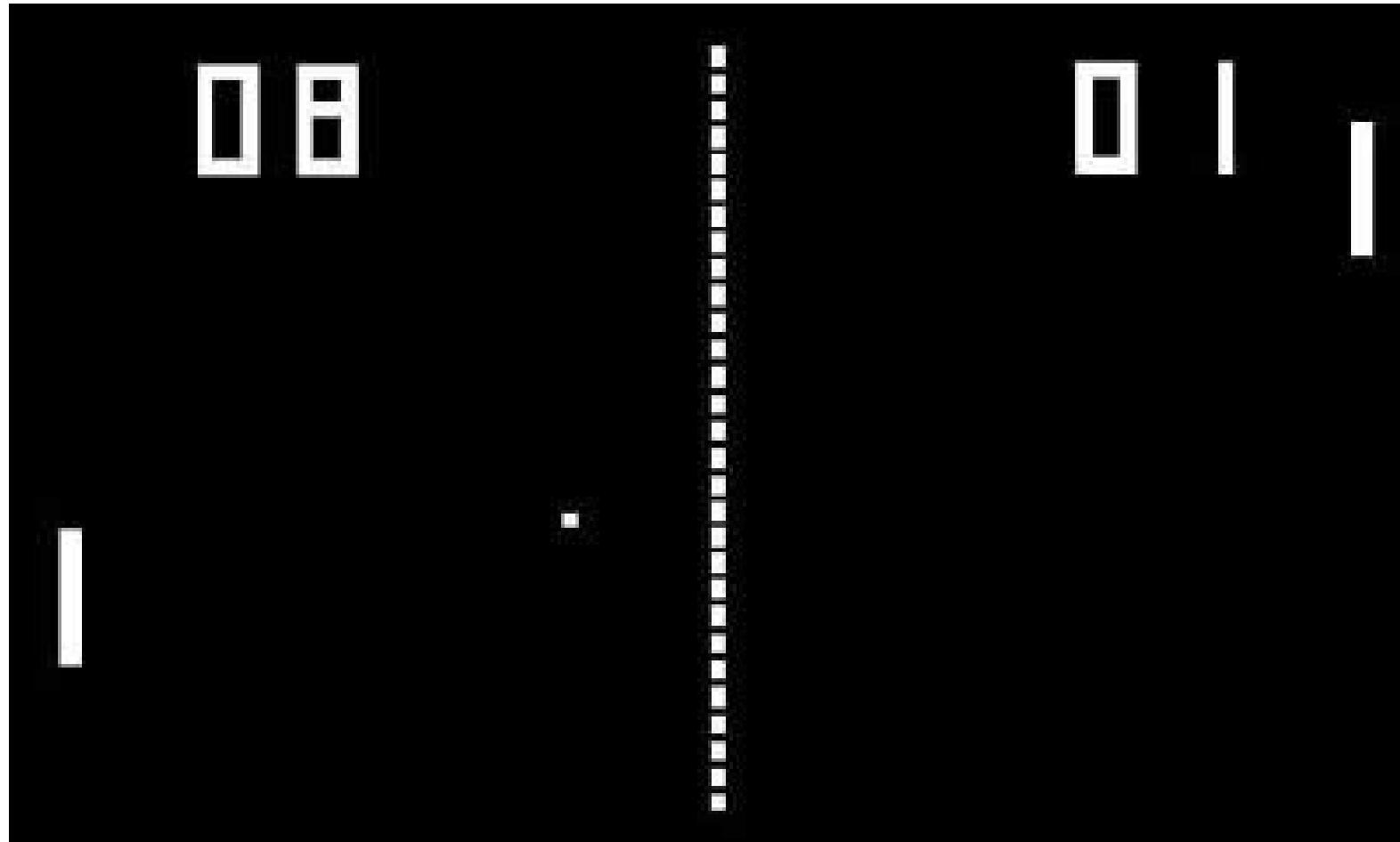
They have the option of at least two dialogues to respond with. Depending on their response, Sonic's affection points might increase. If Sonic really likes the responses, he will ask to go on a minigame date.

Carla



Penguine

# Story Board



Player is on a minigame date with Sonic. They play a short game (pong in this case) with Sonic (AI).

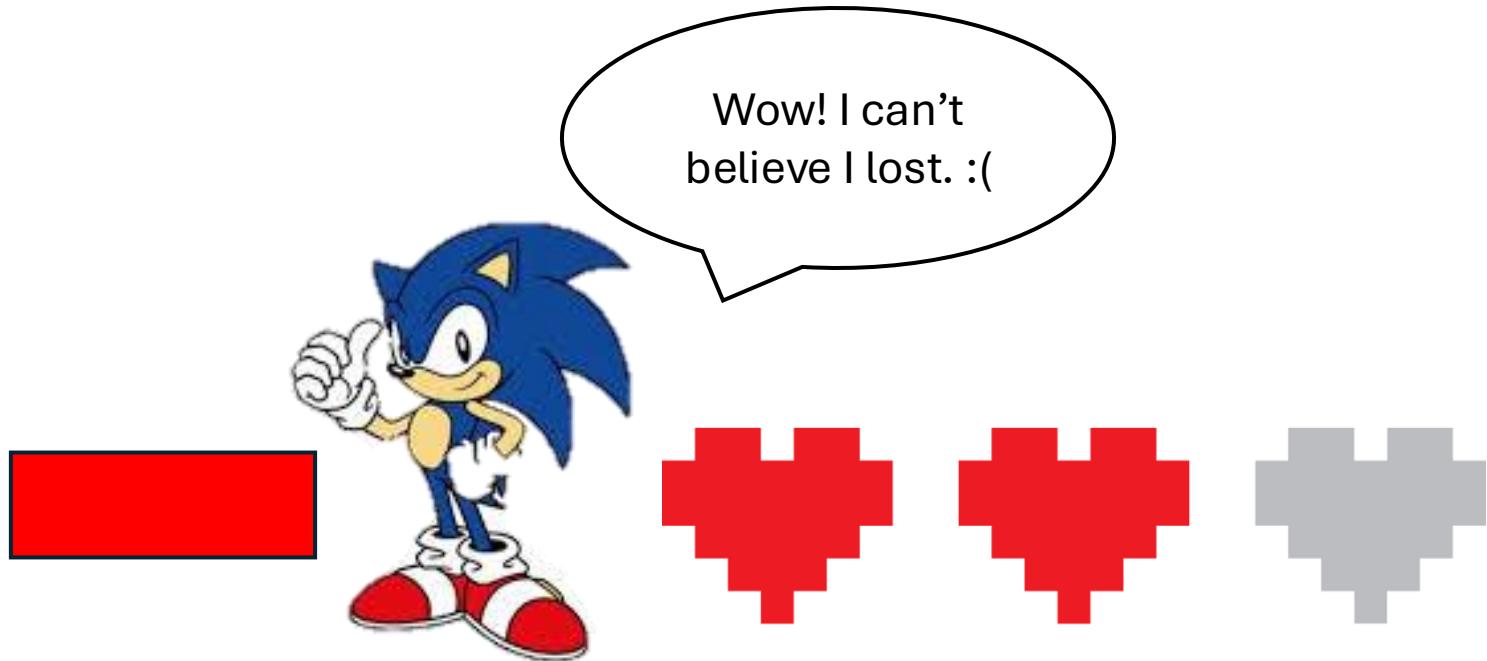
Certain LI's will like winning or losing the games. Sonic likes to win.

Carla



Penguine

# Story Board



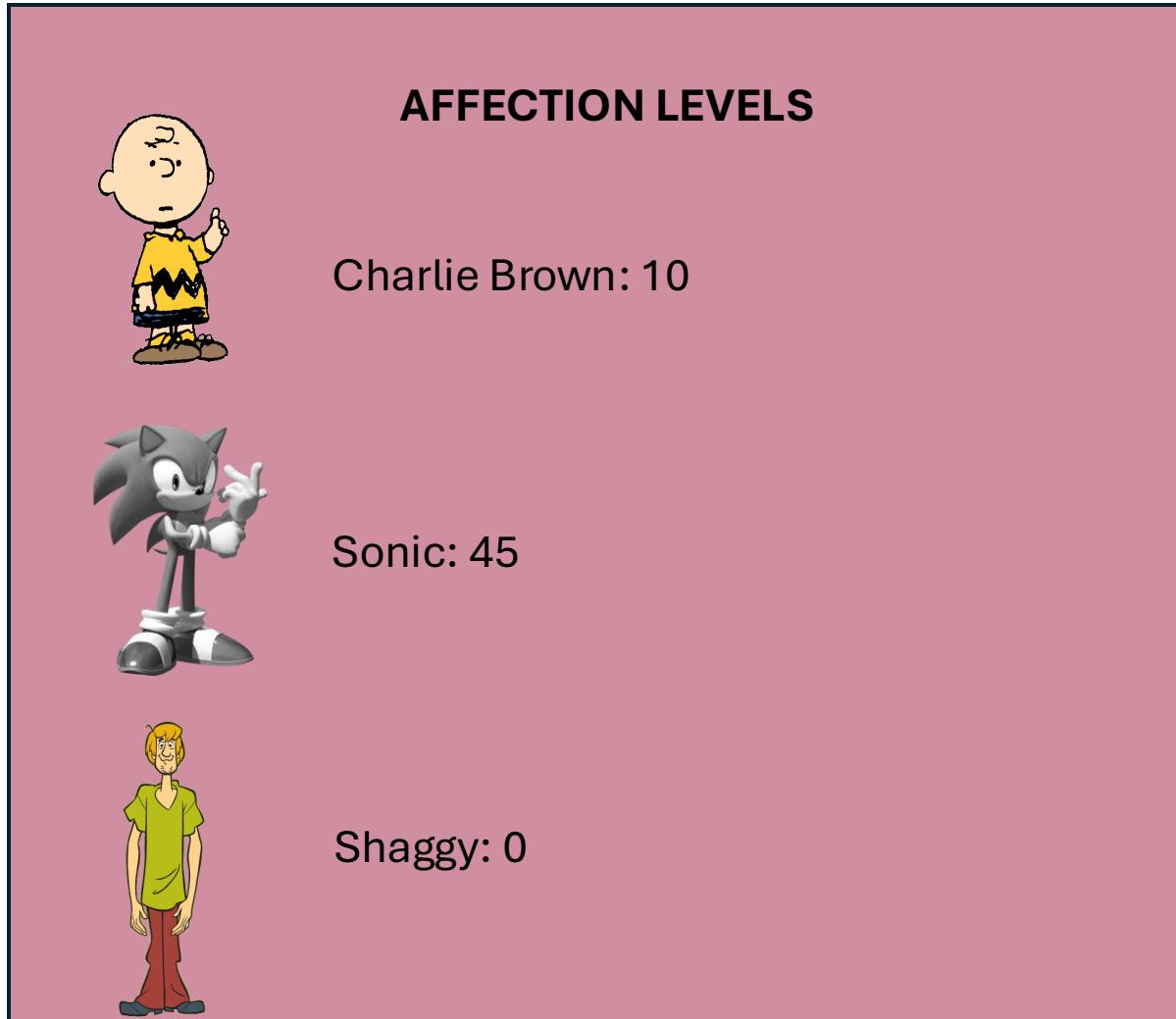
You won the game, and hence lost some of Sonic's affection points.

Now you will reenter the opening library shot and choose who to talk to next.



Penguine

# Story Board



Player pulls up the HUD menu to see all the other LI affection levels for them.

In this case, Sonic has been unenchanted enough by the player that the player has no chance with him, so he is greyed out.

Note: Some characters already have affection points for the player without meeting them, some don't.



# Story Board



Player is in the library again and can choose who to talk to next. Say player chooses Charlie Brown.

Notes: Sonic is greyed out because player has disappointed him and exhausted any further romantic chances.



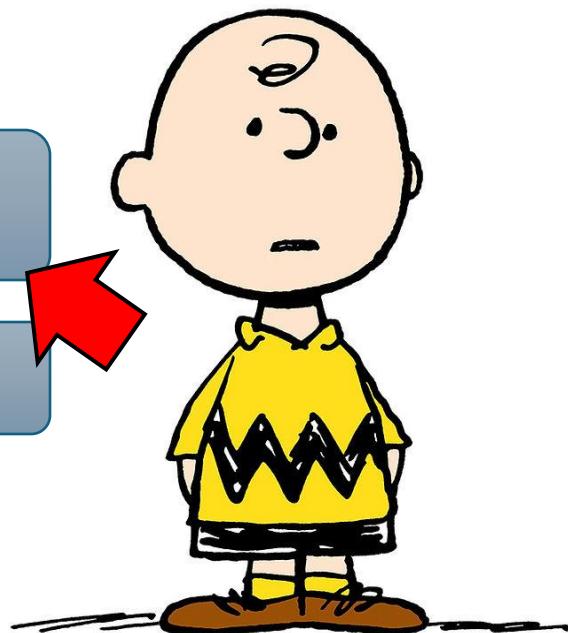
Penguine

# Story Board

Good grief. Don't you  
know you can't talk  
in a library?

Sorry Charlie! I just  
wanted to talk to you :)

Don't be a downer,  
Charlie Brown

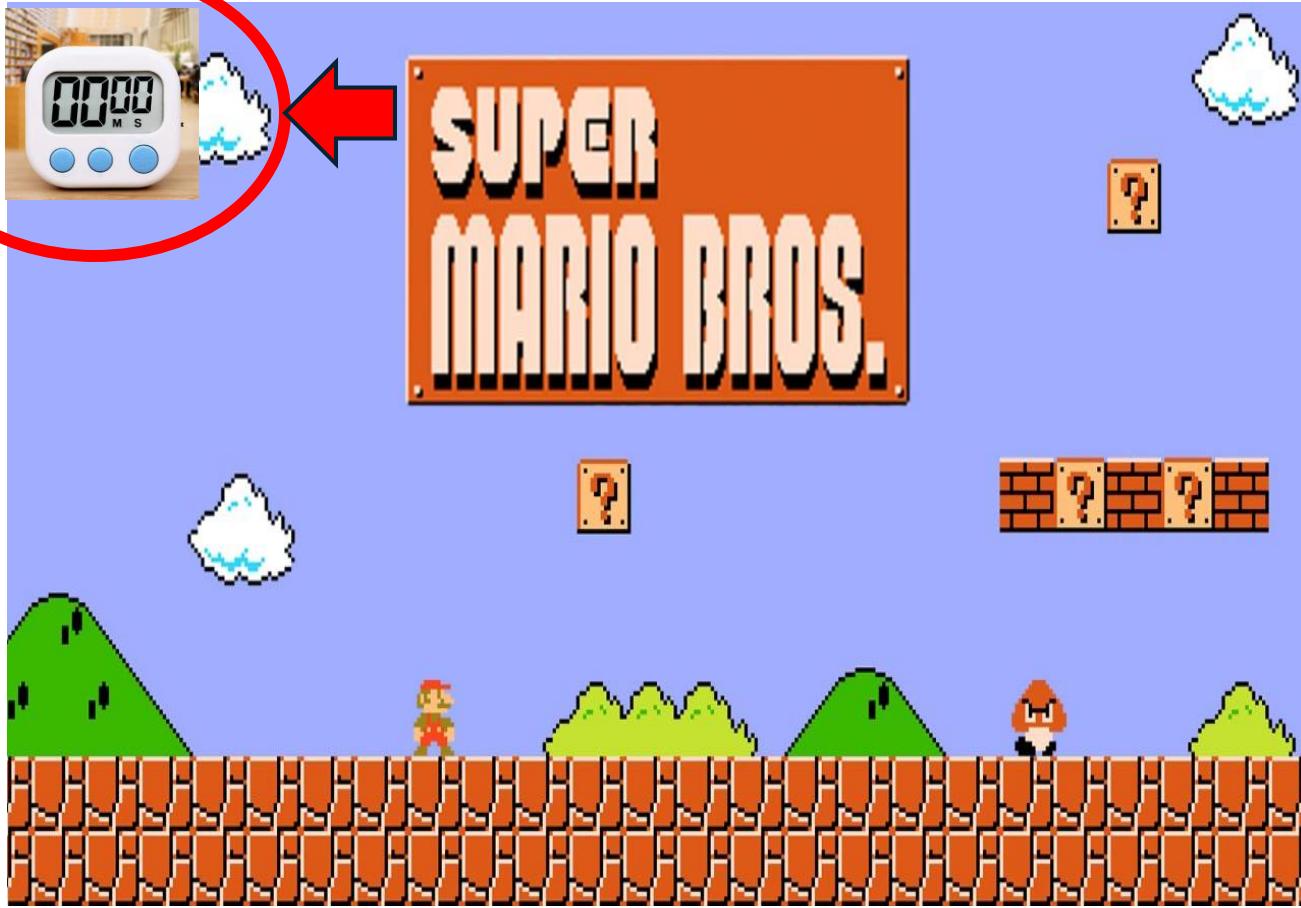


Player is talking to Charlie Brown, who has his own dialogue, with different dialogue response options. Similar to Sonic, his affection levels and further dialogue will depend on which responses the player chooses.



Penguine

# Story Board



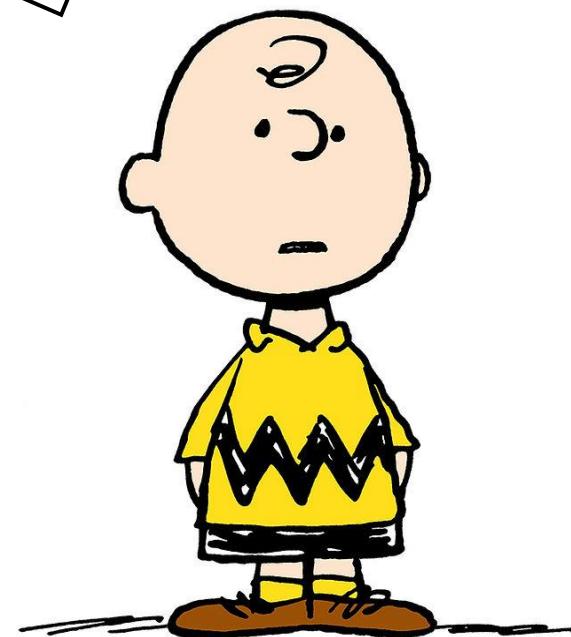
Player is on a minigame with Charlie Brown because he likes you. In this game, he is not playing against you, but watching you play a timed game. His affection levels are dependent on if you win in time. You won't know his response until the game is over.



Penguine

# Story Board

Wow, you're worse  
than me! I sure like  
playing an equal.



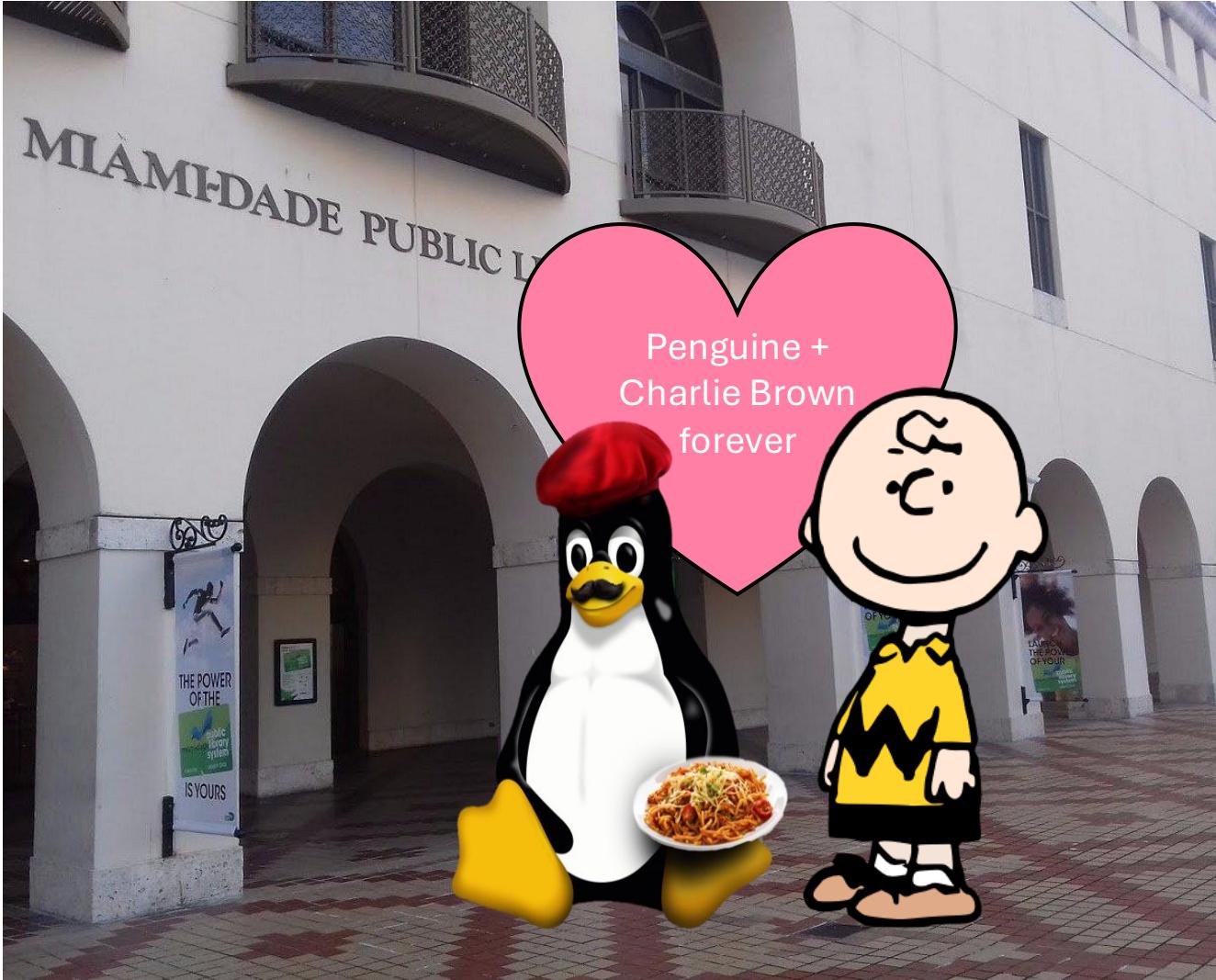
You lost the game, and  
Charlie Brown liked that.  
His affection levels go up.

Carla



Penguine

# Story Board



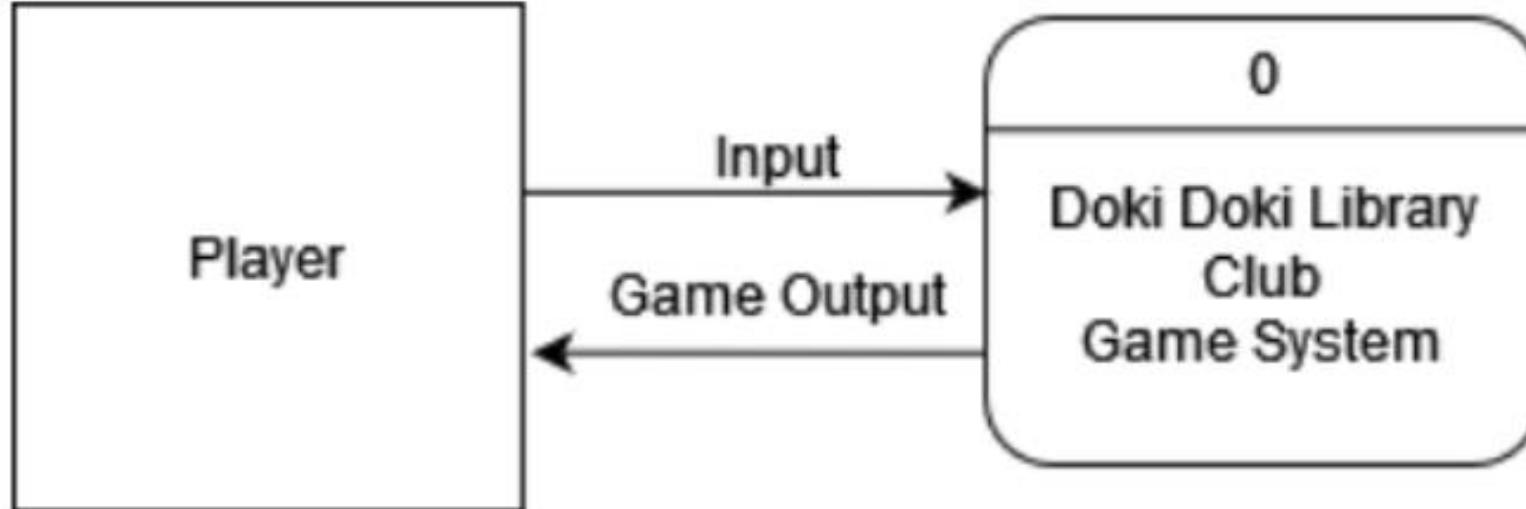
Charlie Brown's affection levels for you are high enough that you have won his heart, and having found a partner to leave the Miami-Dade Public Library with, you won the game. The final screen is a cute heart with your chosen partner.

Carla



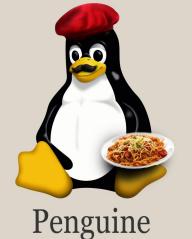
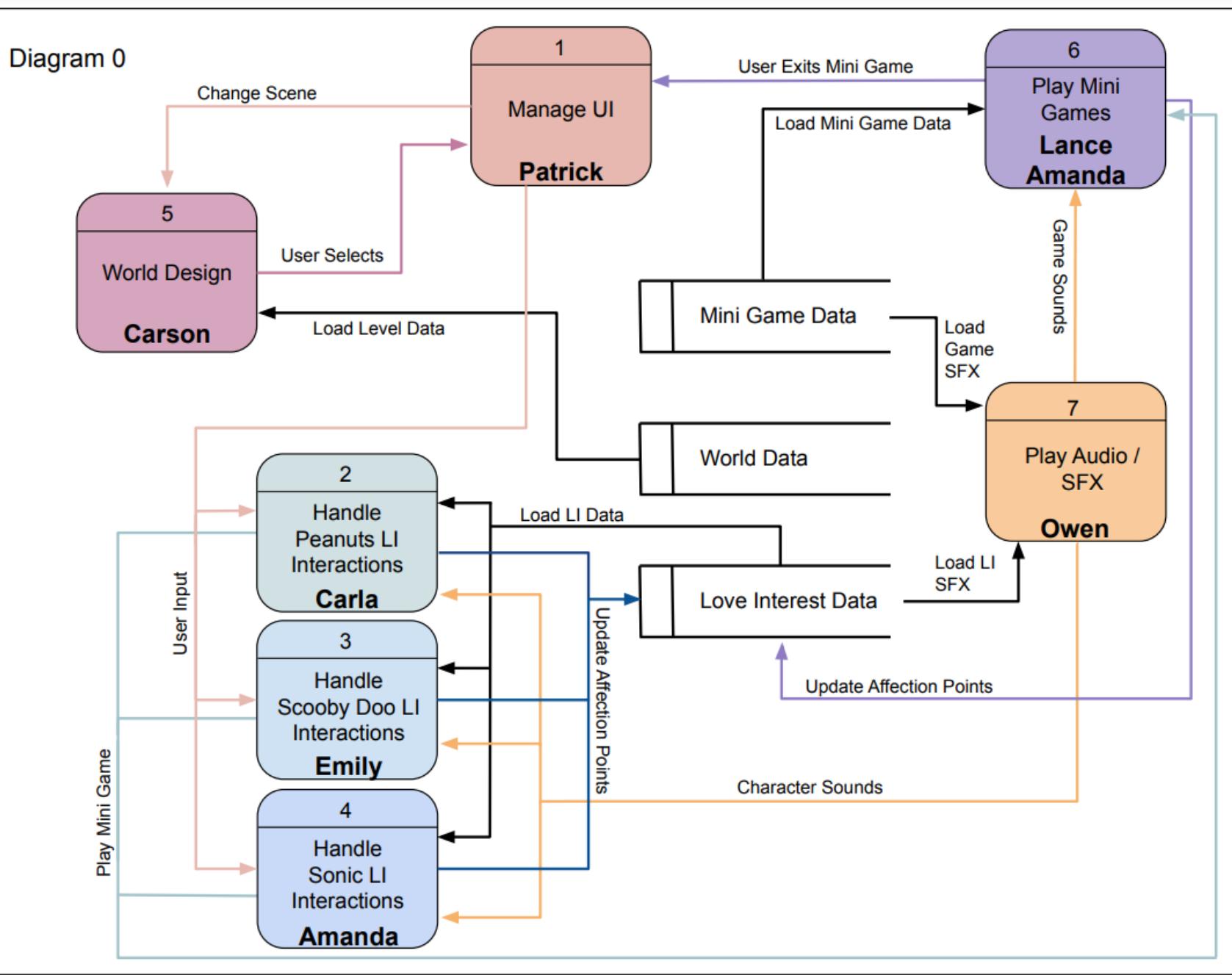
Owen

# CONTEXT DIAGRAM SLIDE



Carson

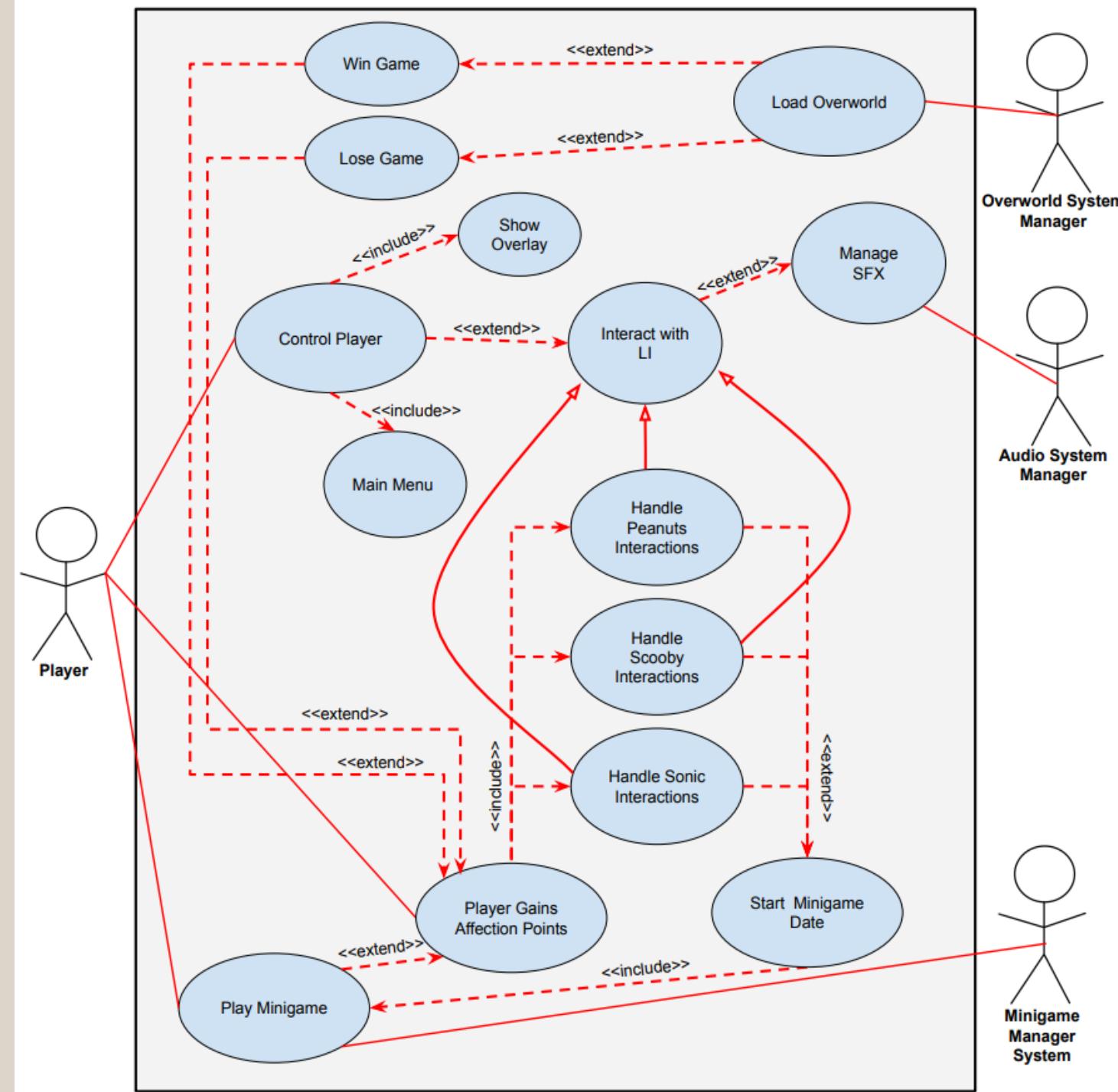
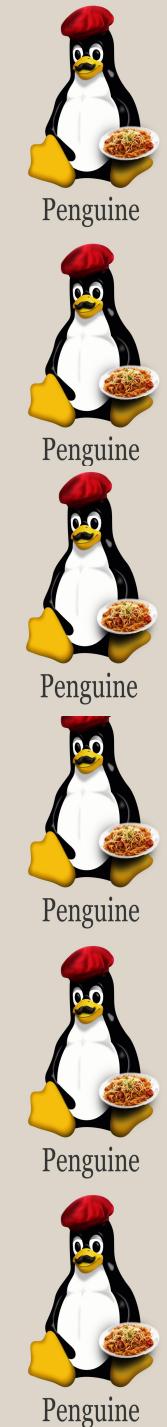
# DIAGRAM 0



Penguin

# GLOBAL USE CASE

- Patrick: Control Player – Show Overlay, Main Menu, Interact with LI
- A/C/E: Handle Each LI Interaction
- Owen: Audio/SFX
- Owen: Player Gains Affection Points (eventually enough to initiate a date)
- Lance: Start Minigame and Play
- Owen: Player Gains Affection Points (updated)
- Carson: Win/Lose the Game and Manage Overworld





# Process 1: UI Management

---

Patrick

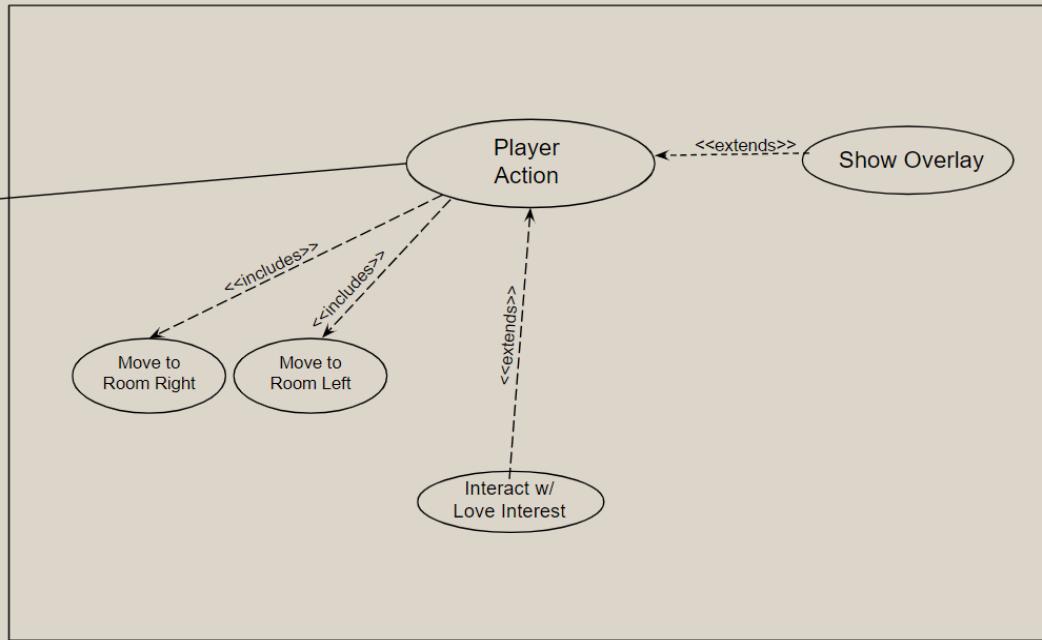
- My feature: UI management
  - Room Movement
  - Love Interest Interaction
  - Main Menu
  - Show Overlay
- Priority: 1
  - Room changes and character interaction input are essential to functionality of our game.
- Relatively Low Complexity than other features



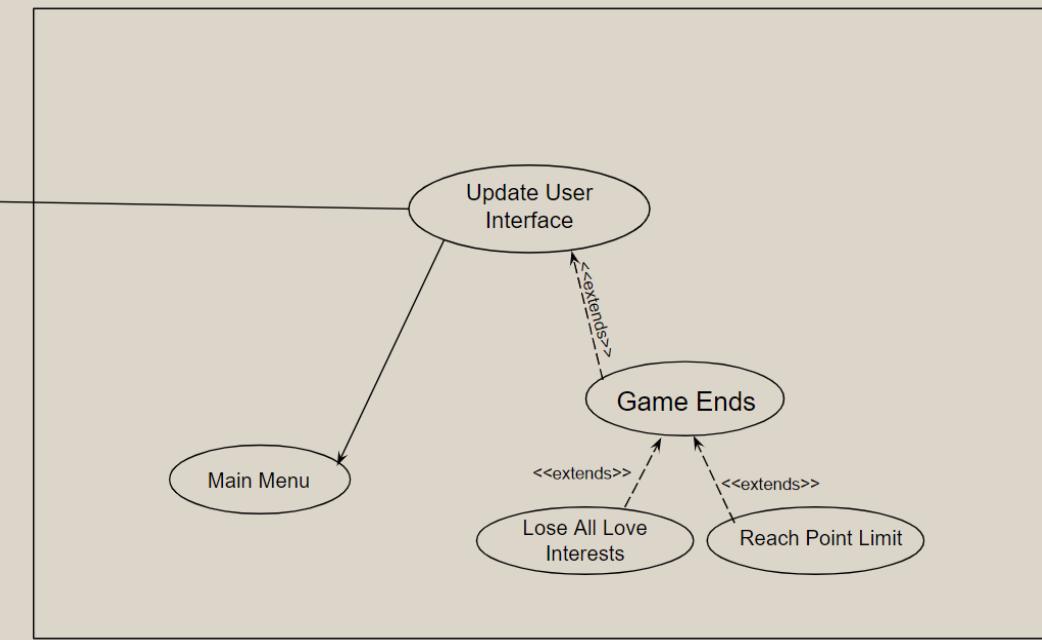
# Use Case Diagrams:

Process 1: UI Management

player



player



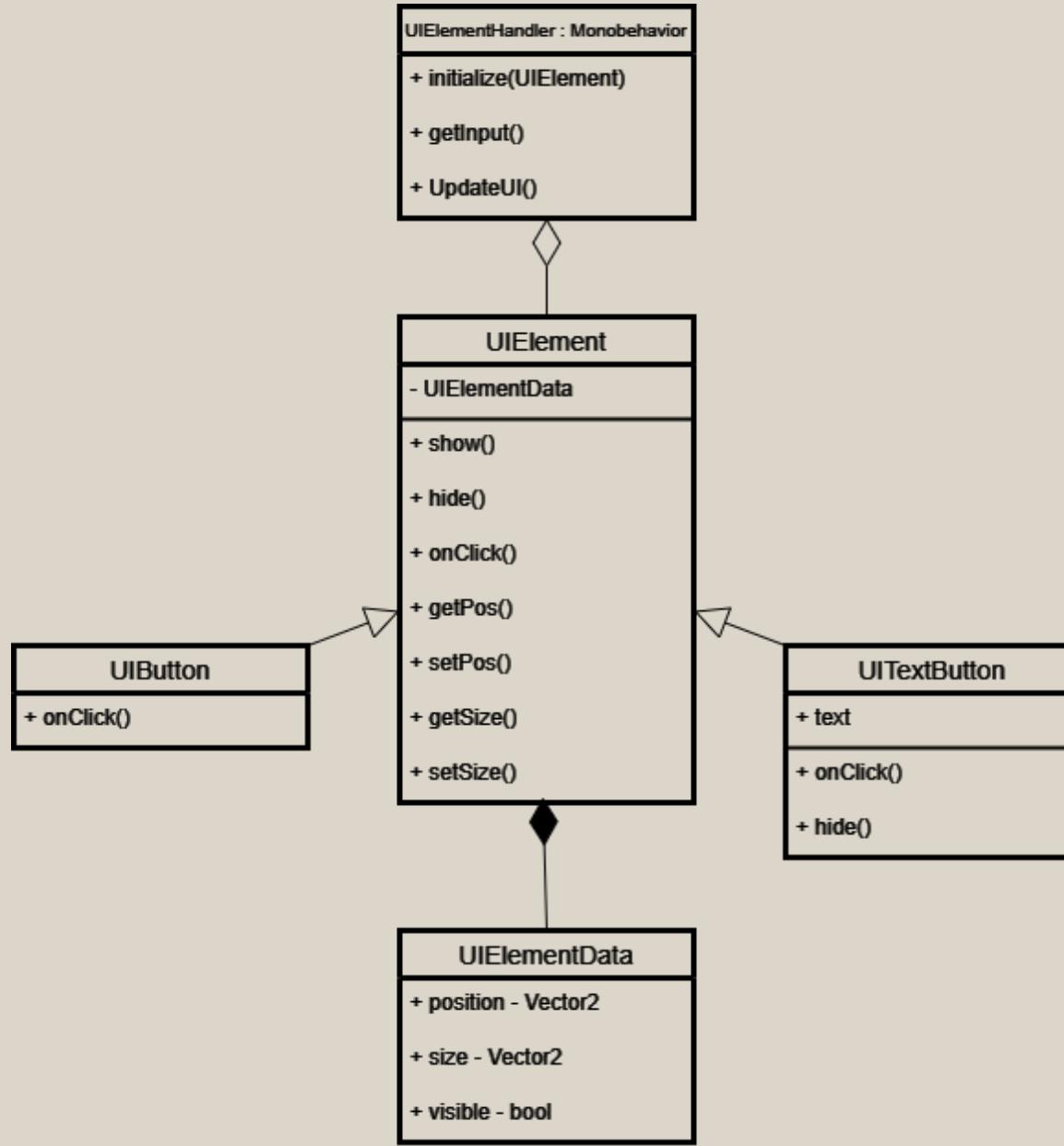
Patrick

# Class Diagram

Process 1: UI Management



Penguine



Patrick



Carla

# Process 2: Peanuts Characters LI Interactions

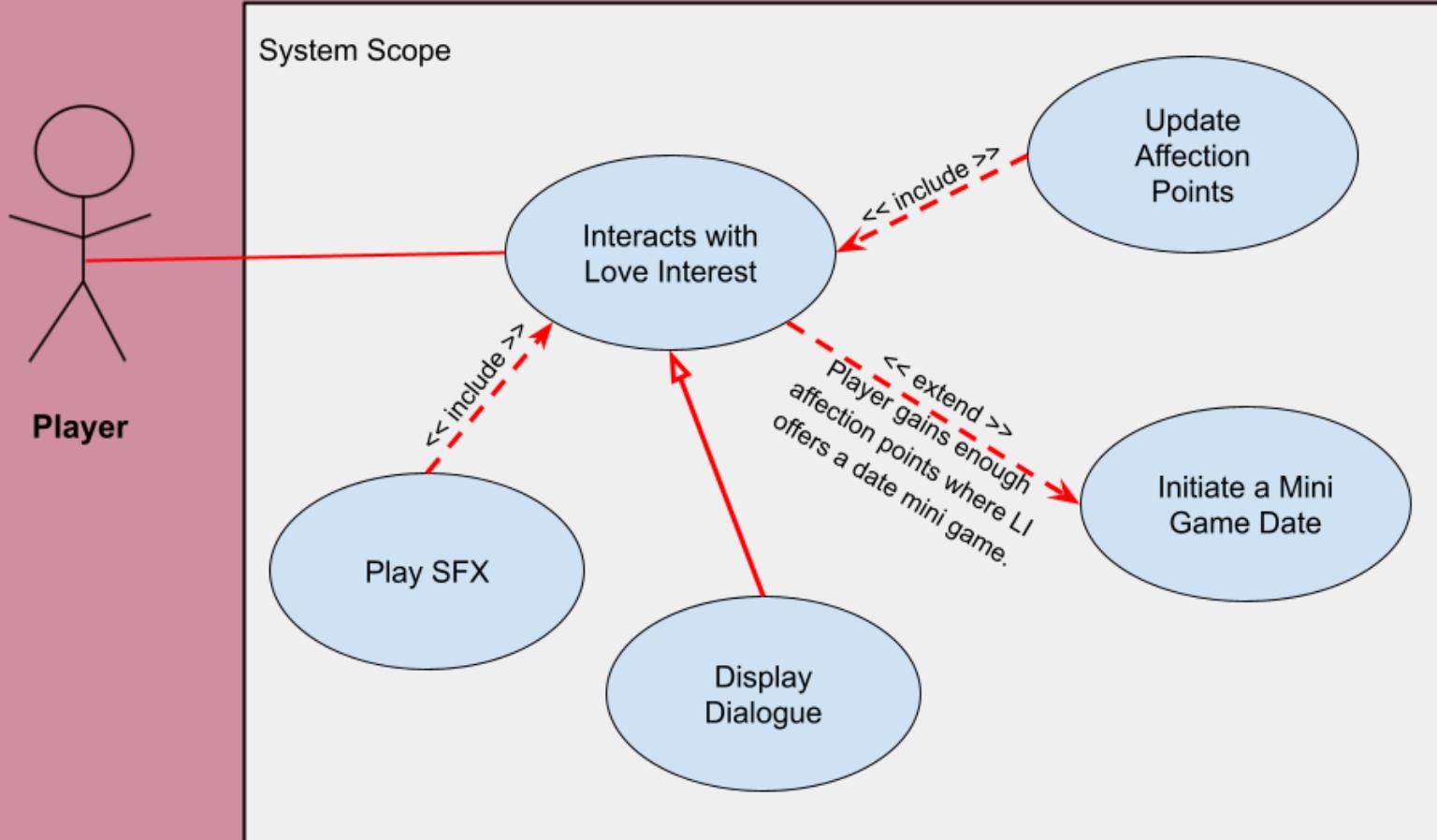
- My feature: designing all Peanuts-inspired Love Interest characters
- Includes:
  - Preferred dialogue answers unique to each character
  - A favorite minigame for each character
  - Visual character designs and representations
- Priority: 1\* (essential)
- Complexity: Less Complex



Penguine

# Peanuts LI Interactions

## Use Case

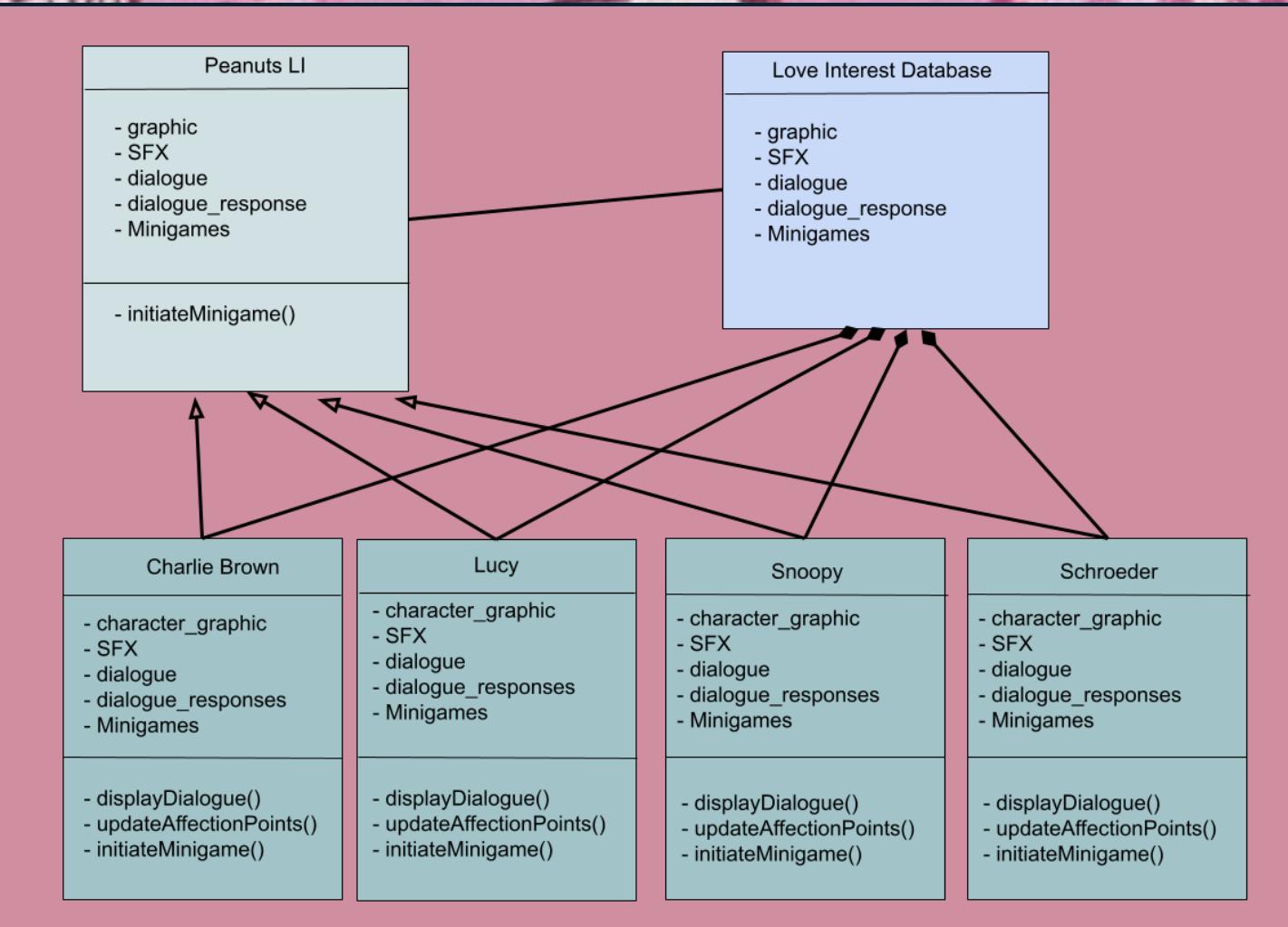


Carla



# Peanuts LI Interactions

## Class Diagram



Carla



Emily

## Process 3: Scooby Doo LI Interactions

My feature: designing all Scooby Doo-inspired Love Interest characters

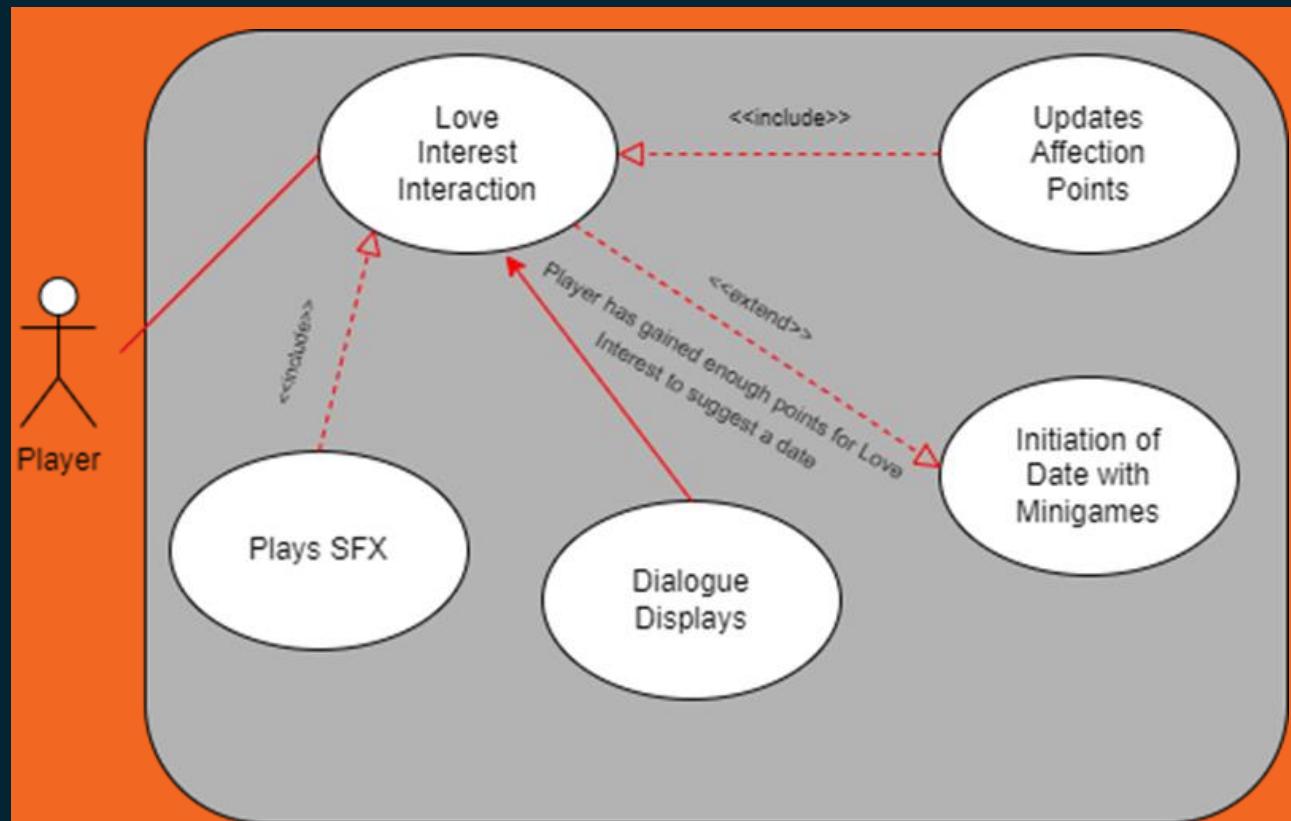
Includes:

- Preferred dialogue answers unique to each character
- A favorite minigame for each character
- Visual character designs and representations
- Complexity: Less Complex

Priority: 1\* (essential)



# Scooby Doo LI Interactions



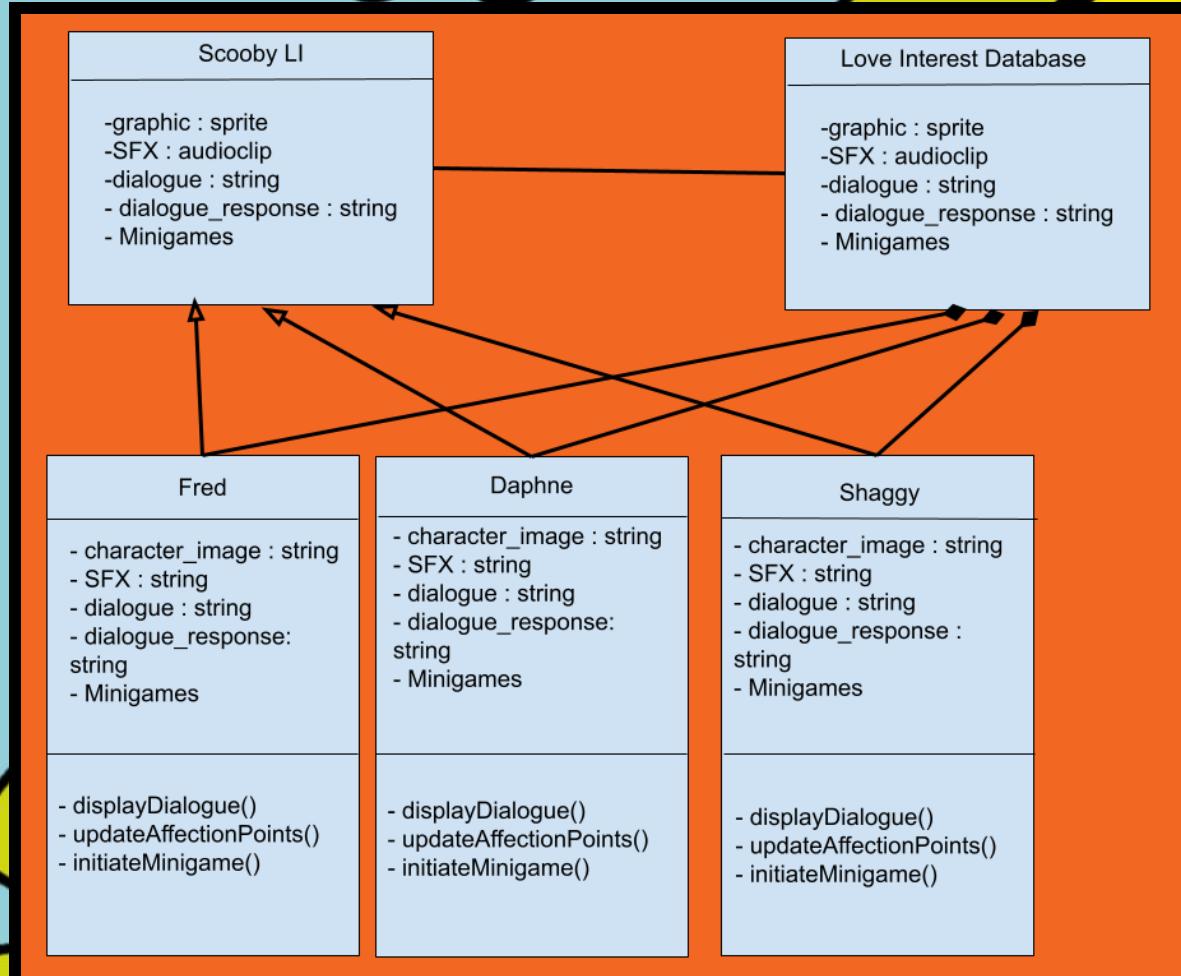
## Use Case Explained:

- Player initiates interaction with Love Interest
- Plays SFX and displays dialogue and response options
- LI can initiate a minigame date if they like the player
- Results of likes/dislikes regarding dialogue responses and minigame are used to update affection points

Emily



# Scooby Doo LI Interactions



## Class Diagram:

- Superclass: **Scooby LI**
- Subclasses: **Fred**, **Daphne**, **Shaggy**
- Relation: **Scooby LI** and **Dialogue Database**
- Composition: **Dialogue Database** and all subclasses

Emily



# Process 4: Sonic LI Interactions

My feature: developing all Sonic-themed Love Interests (LI)

Includes:

- Creating a script for main player and LI conversations
- Preferred dialogue answers for each LI
- A favorite minigame for each LI
- Visual representations for each LI

**Priority:** 1\* (*essential*)

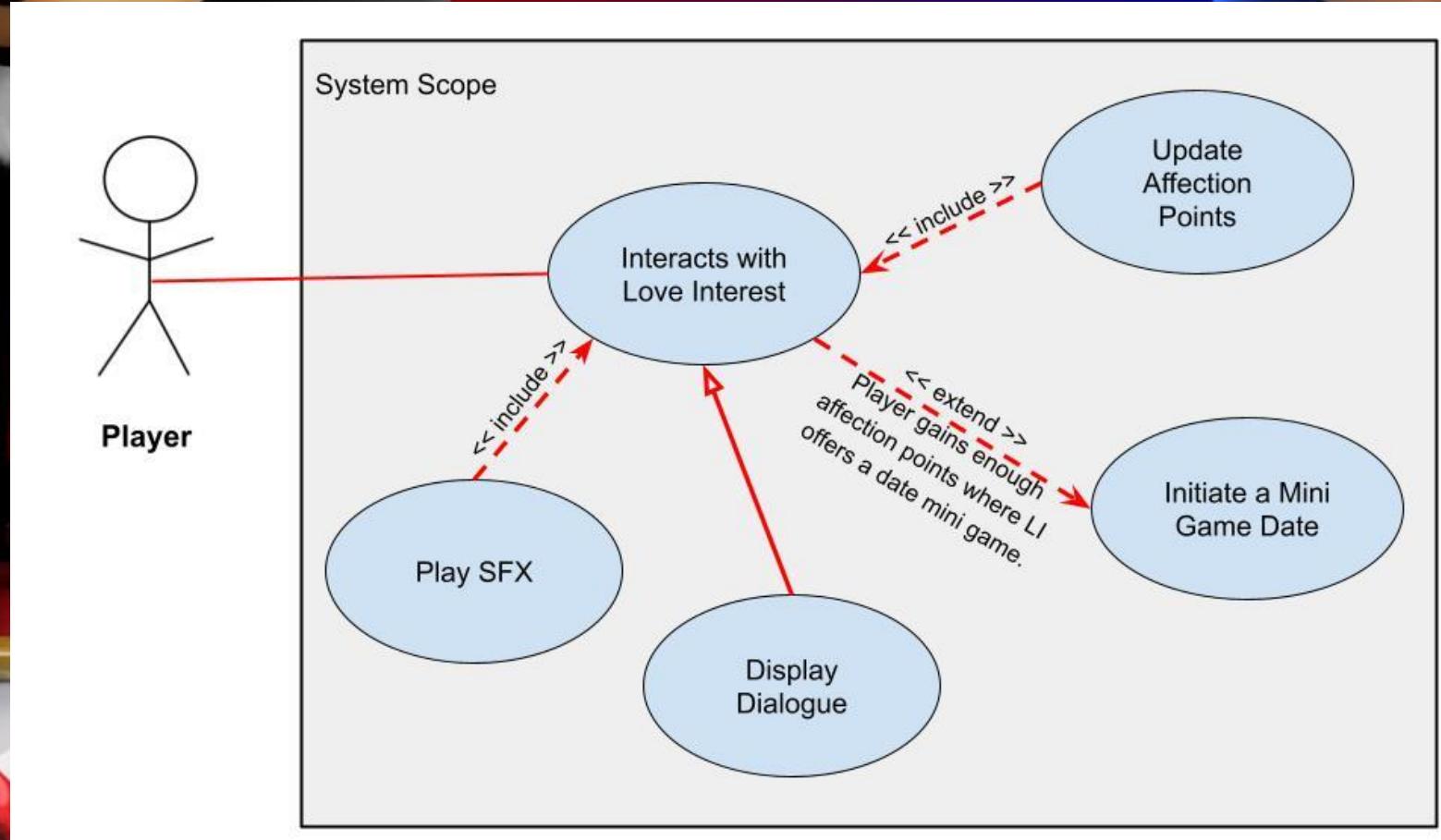


Amanda



# Sonic LI Interactions

Use Case Diagram

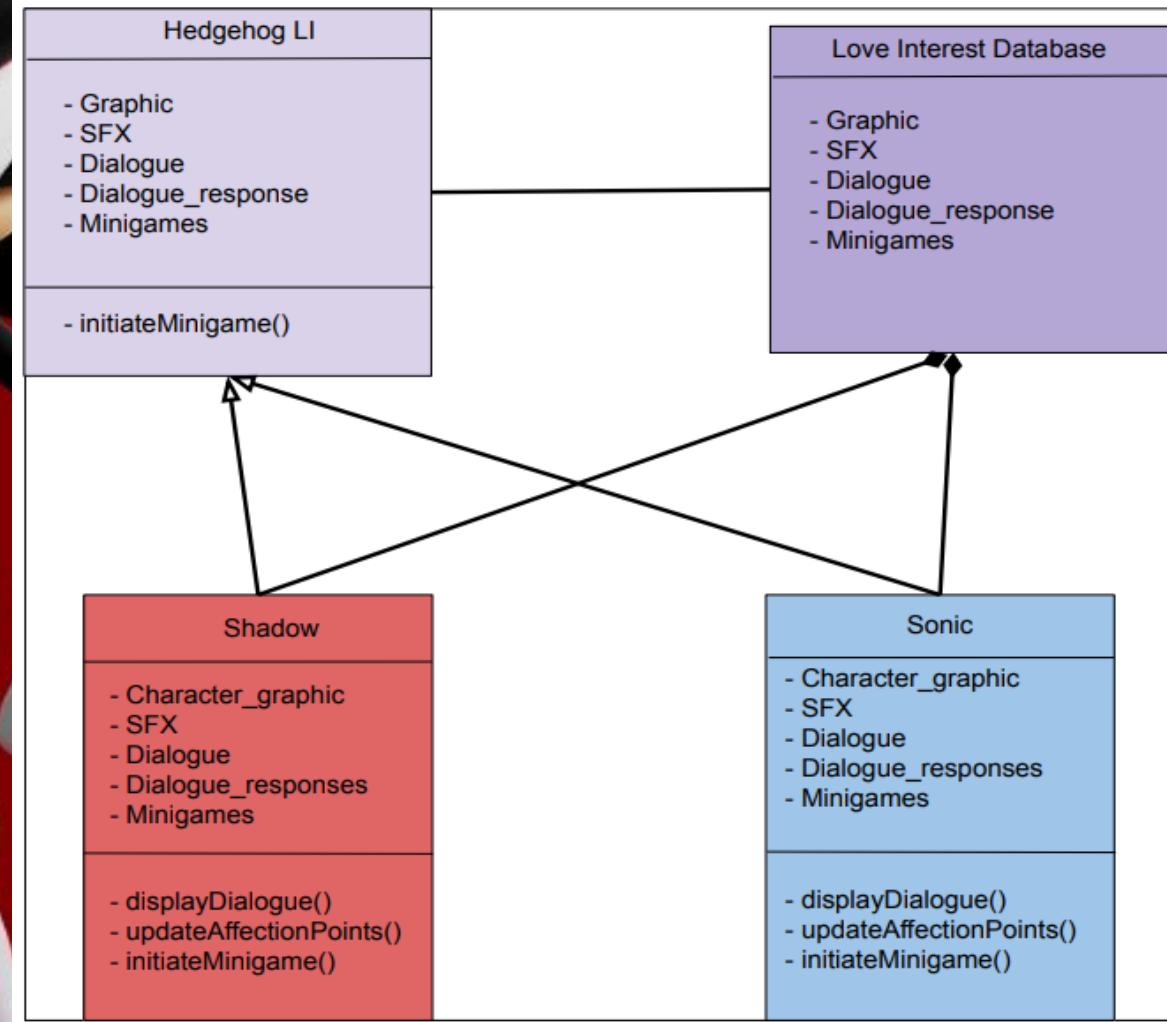
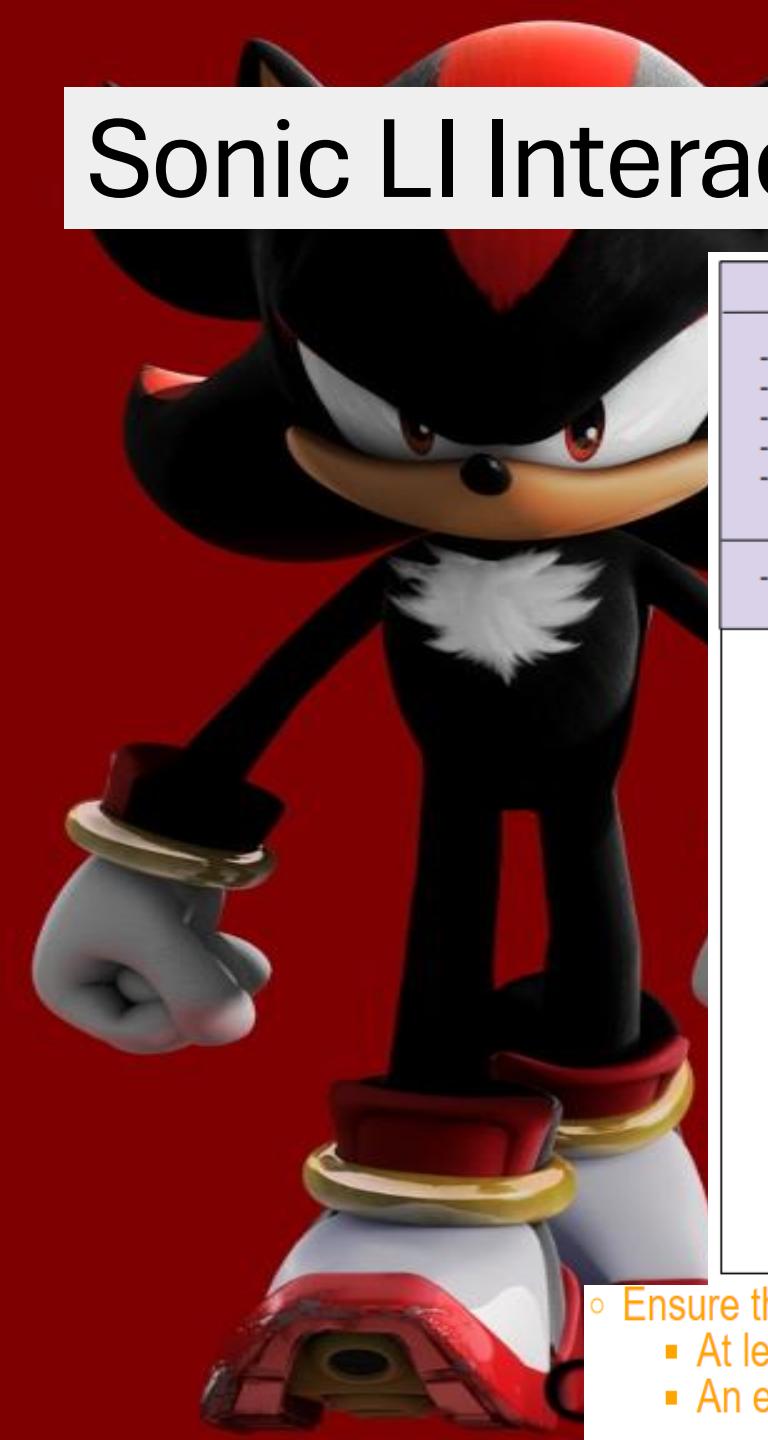


Amanda



Penguinie

# Sonic LI Interactions



- Ensure that your class diagram includes:
  - At least one example of a subclass/superclass (entirely within your code) and
  - An example of aggregation.

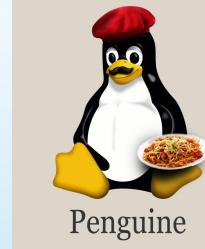
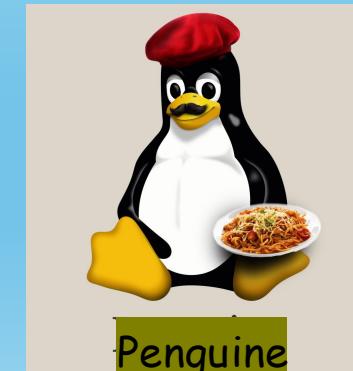


Amanda

# Process 5: Overworld (World Design)

## My feature: World Design

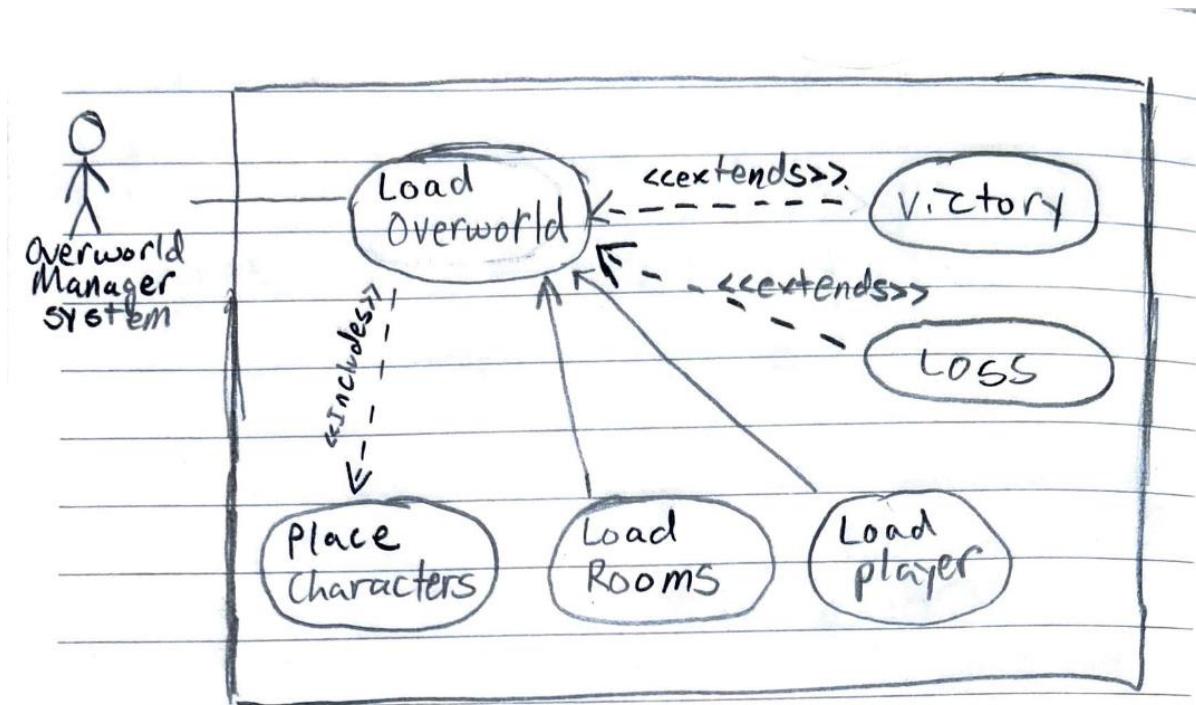
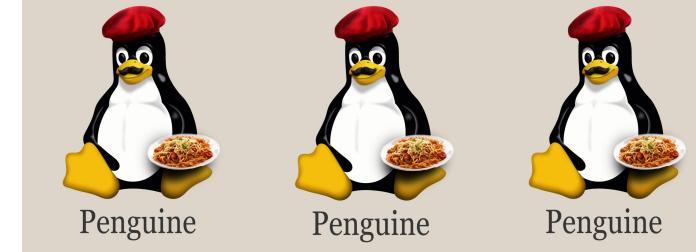
- As the player navigates through the game world, they will interact with characters.
- The game world (library) is made up of many rooms, each with a different appearance and functionality.
- When interacted with, the characters will interact with the player depending on score, lockout status, and other factors retrieved in world data.
- Depending on the character's reaction, the game will enter dialogue, minigames, or remain in the overworld if the character rejects the player's interaction.
- Priority: 2
- Complexity: 2



# Process 5: Overworld

## Use Case 1: Load Overworld

- The game enters the overworld
- Exception: Lose and win conditions are checked. If triggered, victory or loss screens are displayed.
- All rooms are loaded.
- Retrieve locations of the player and the last interacted character. Player and character are placed in these rooms.
- The rest of the characters are loaded in random locations.



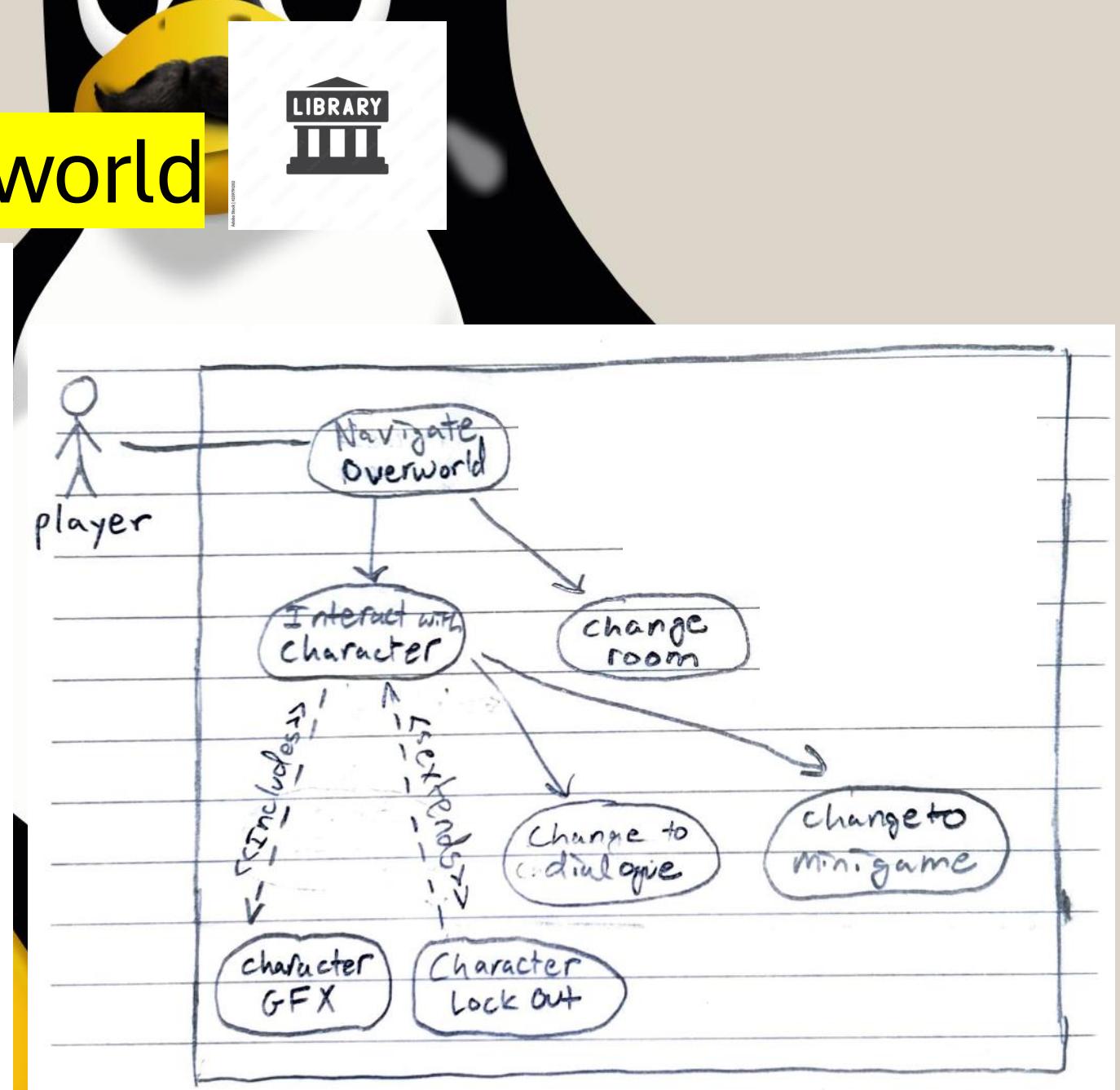


Penguine

# Process 5: Overworld

## Use Case 2:

- The player navigates the library to try and find a character to interact with.
- The player navigates through several rooms.
- The player attempts to interact with a character.
- Exception: If the character is locked out, the character will reject the dialogue. The player will remain in the overworld.
- Otherwise, the player goes on a date with the character (dialogue, minigame)



Penguine



Penguine



# Process 5: Class diagram

Superclass: OverworldObject

Subclass: InteractableObject

Subclass: OverworldCharacter

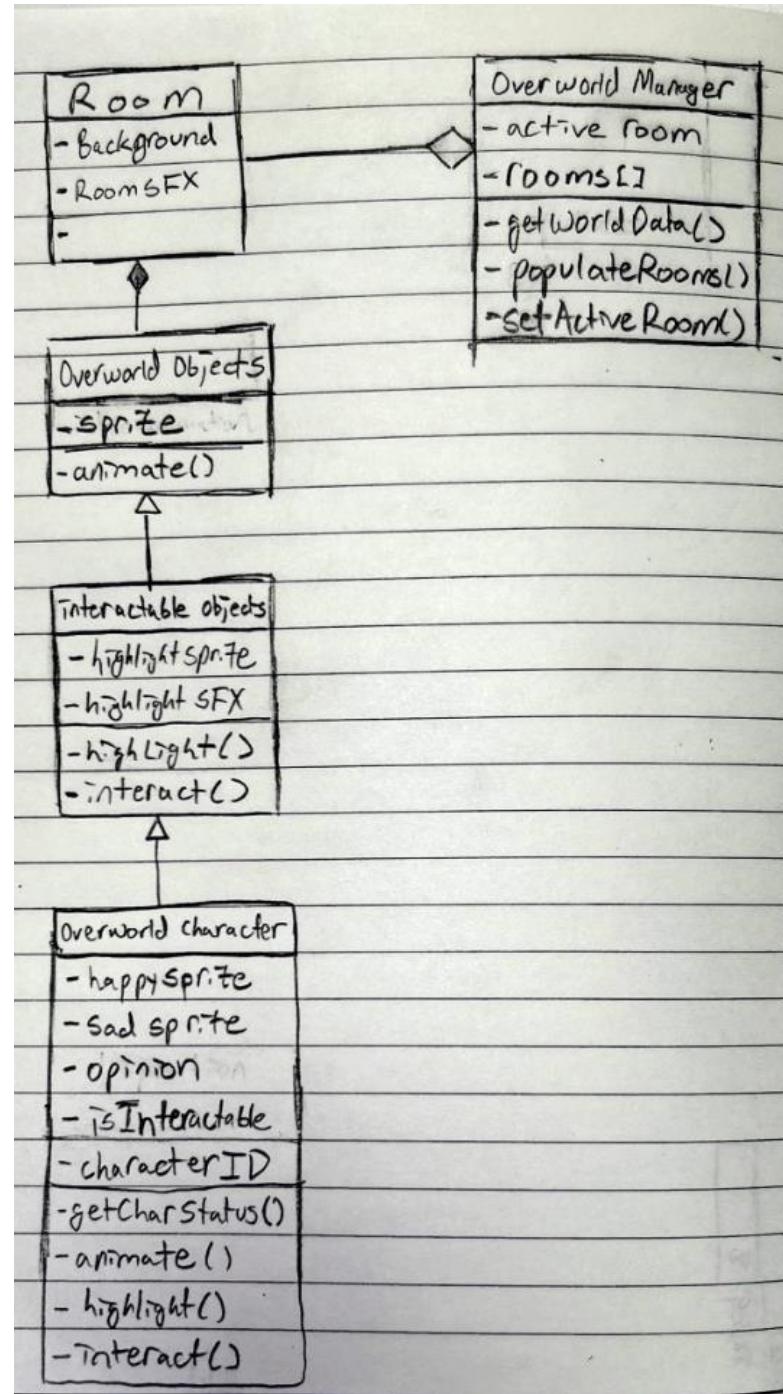
Composition/Aggregation



Carson Gustavel



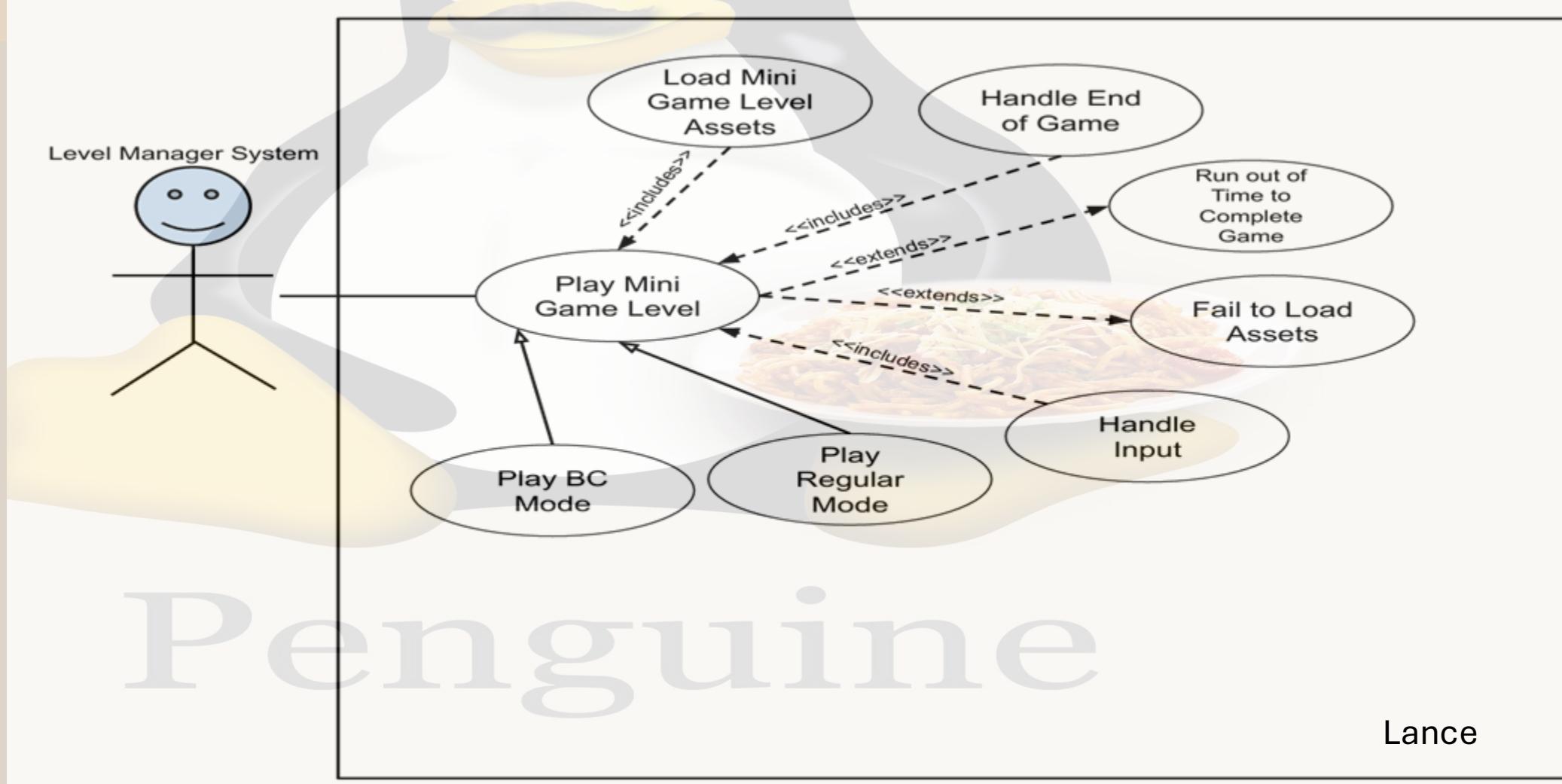
Adobe Stock | carson202



# Complexity: 3

# Priority: 2

# Process 6: Use Case



# Process 6: Continued

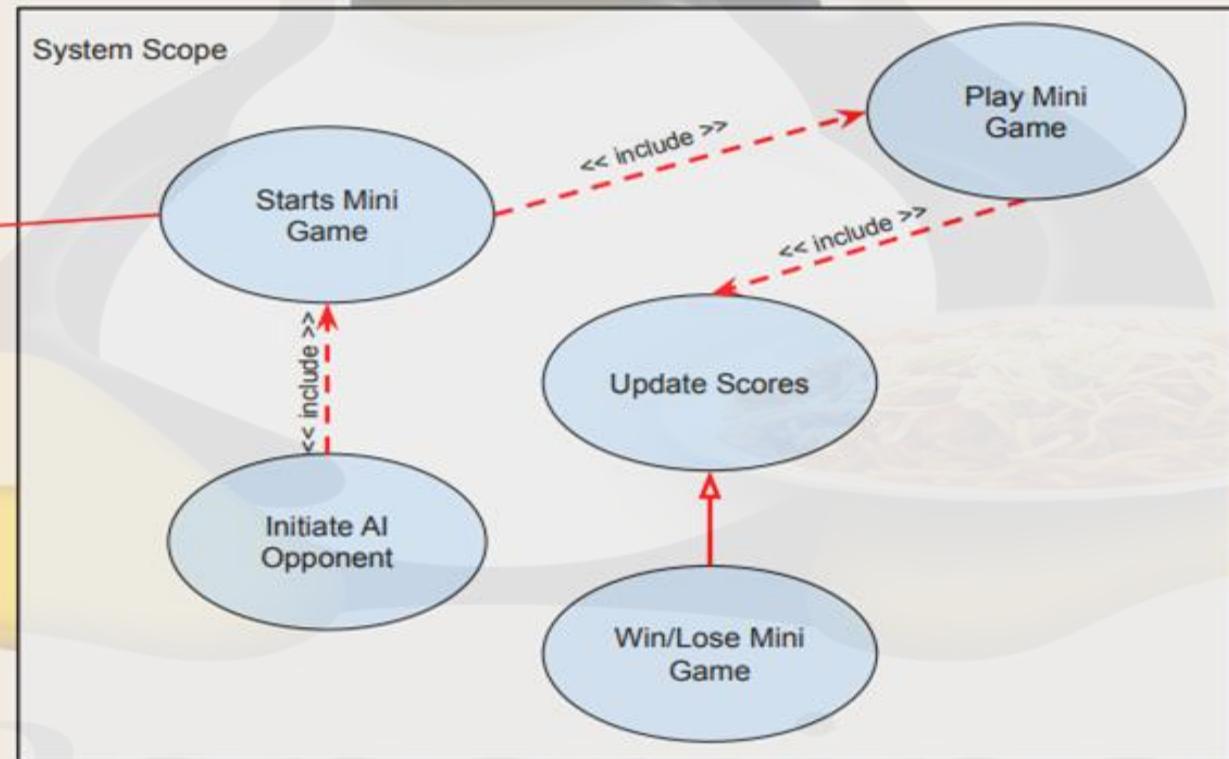
AI Use Case:

Complexity: 3

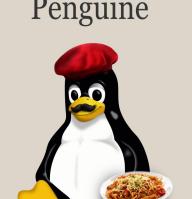
Priority: 2



Player



Penguin



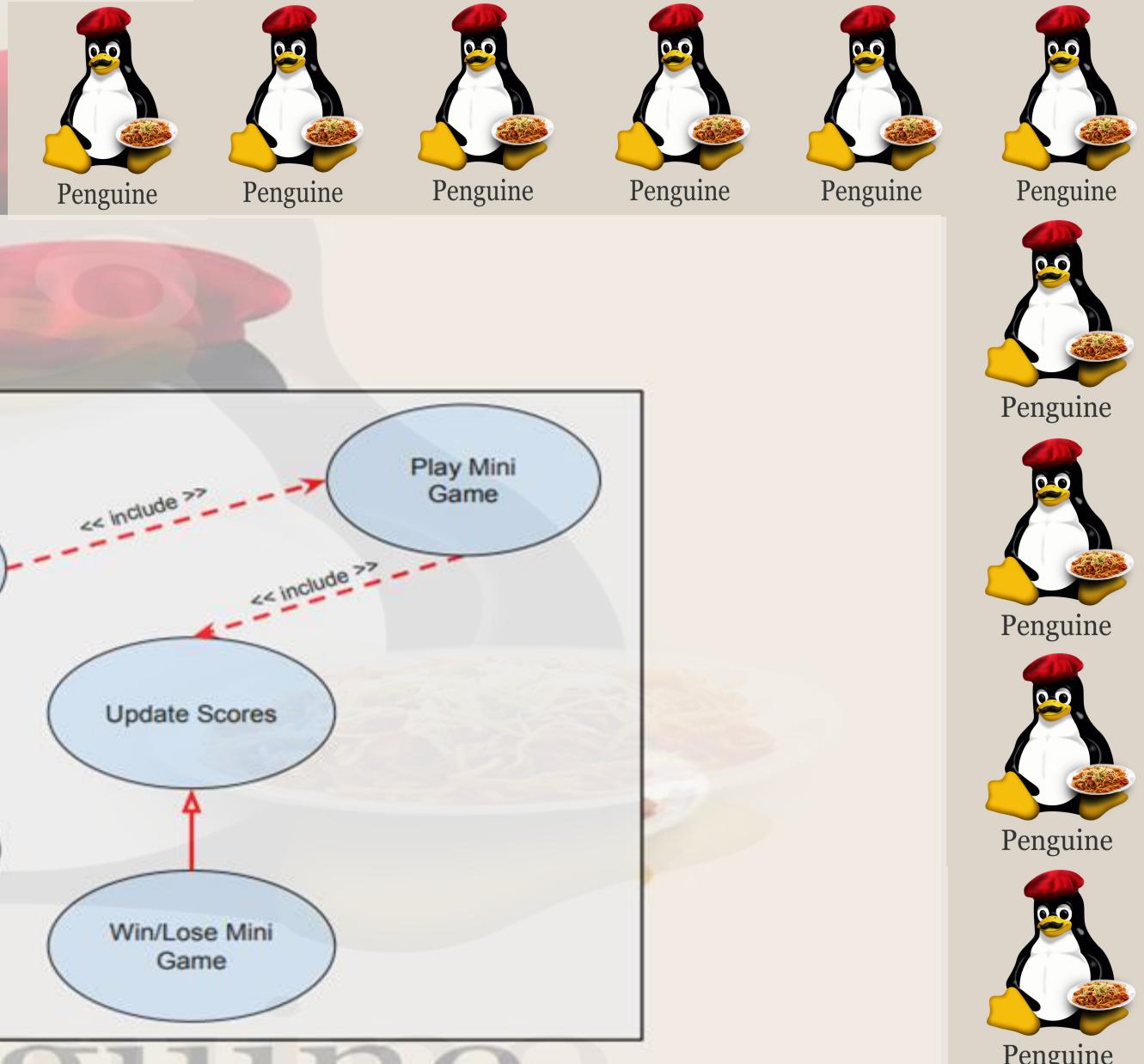
Penguin



Penguin



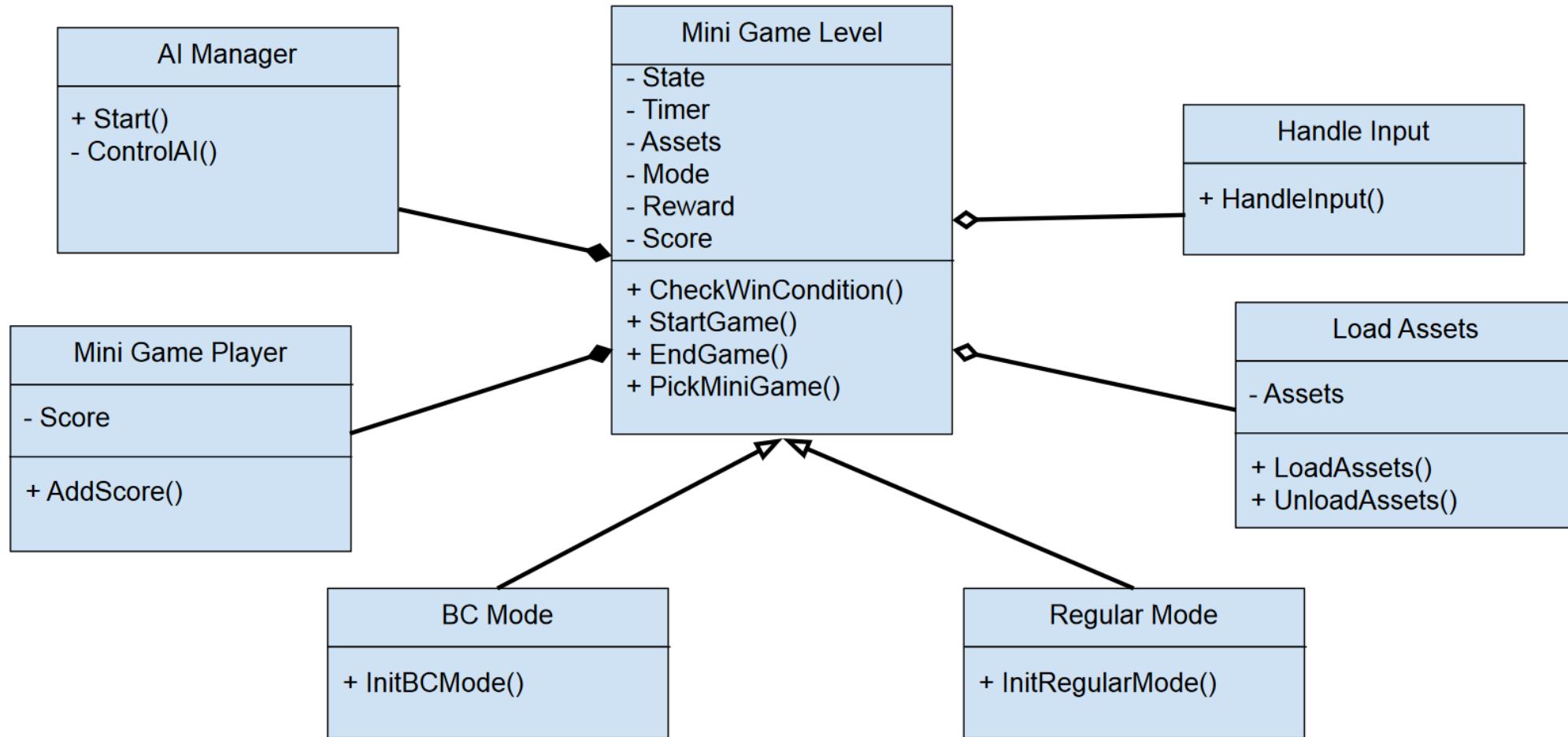
Penguin



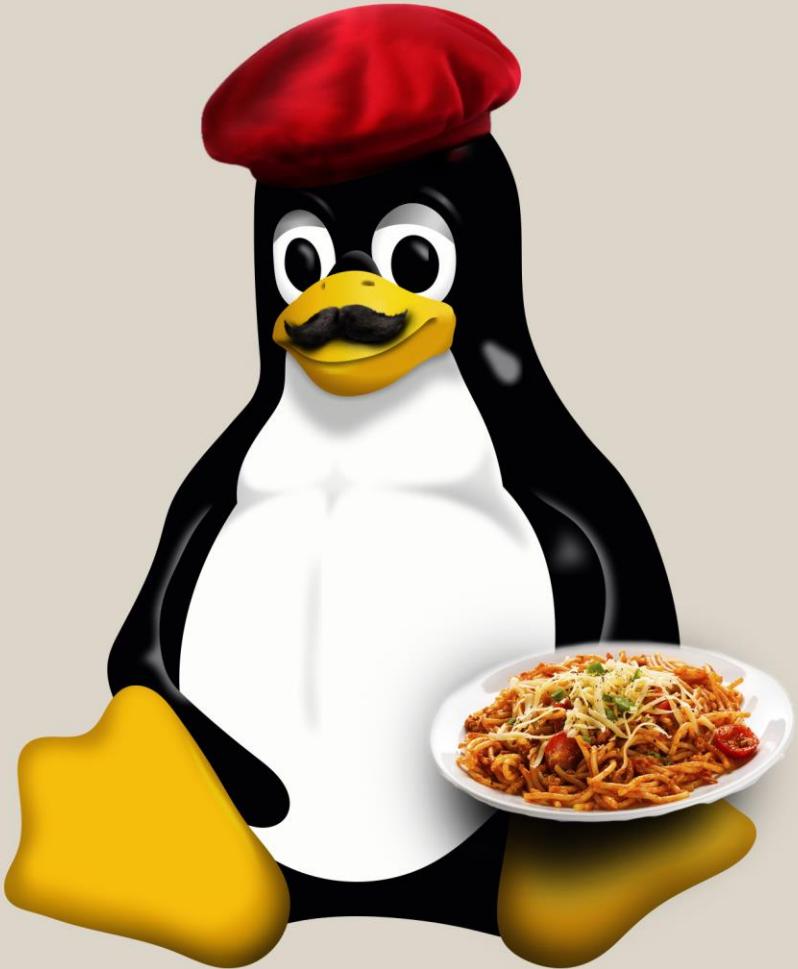
# Process 6: Class diagram



Penguin



LanCe

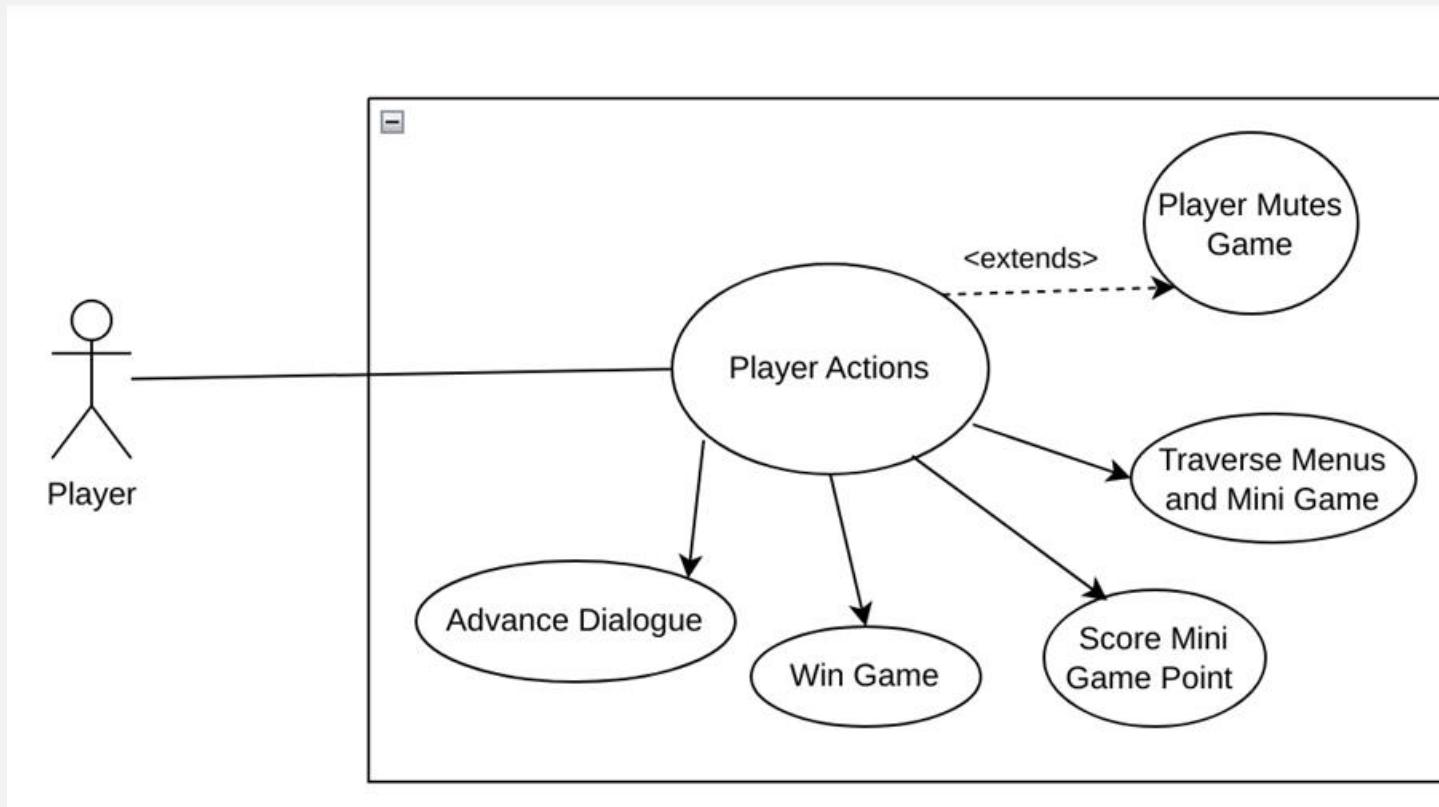


# Penguine

## Process 7: Audio (Owen McDaniel)

- My Feature:
  - Sound Management
    - Background Music
    - Dialogue Sounds
    - Menu and Button Sounds
    - Mini Game Audio
  - Scoring System
- Priority: 3
  - Audio isn't necessary for the game to function, but it does make the game more responsive and polished.
- Complexity: 2
  - Audio and Scoring shouldn't be too complicated besides the interaction with the databases

# Process 7: Audio Case Diagram



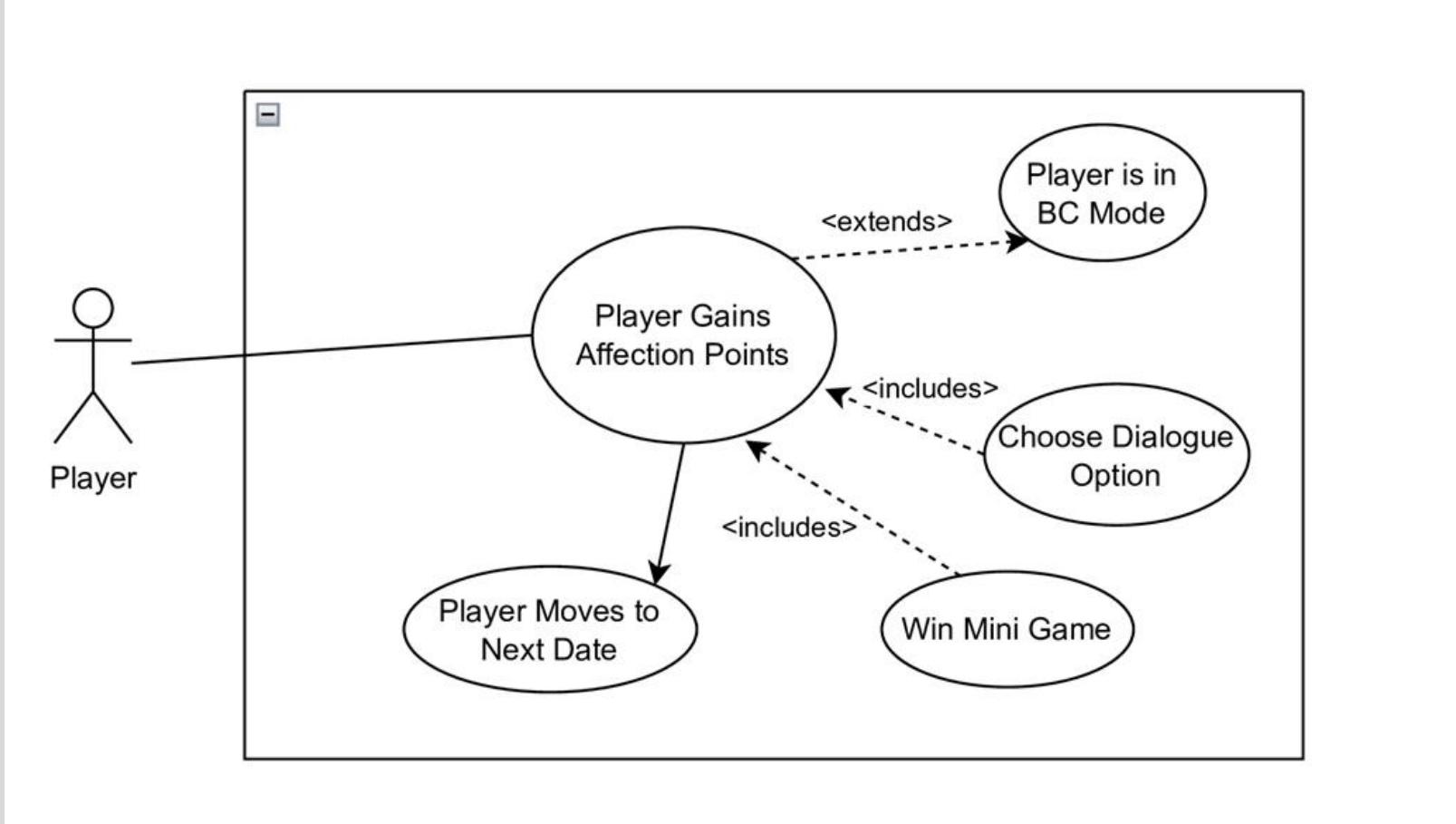
Penguine



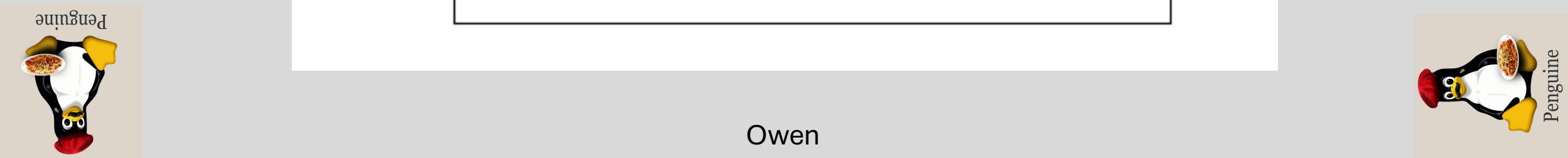
Penguine



# Process 7: Scoring Case Diagram



Owen



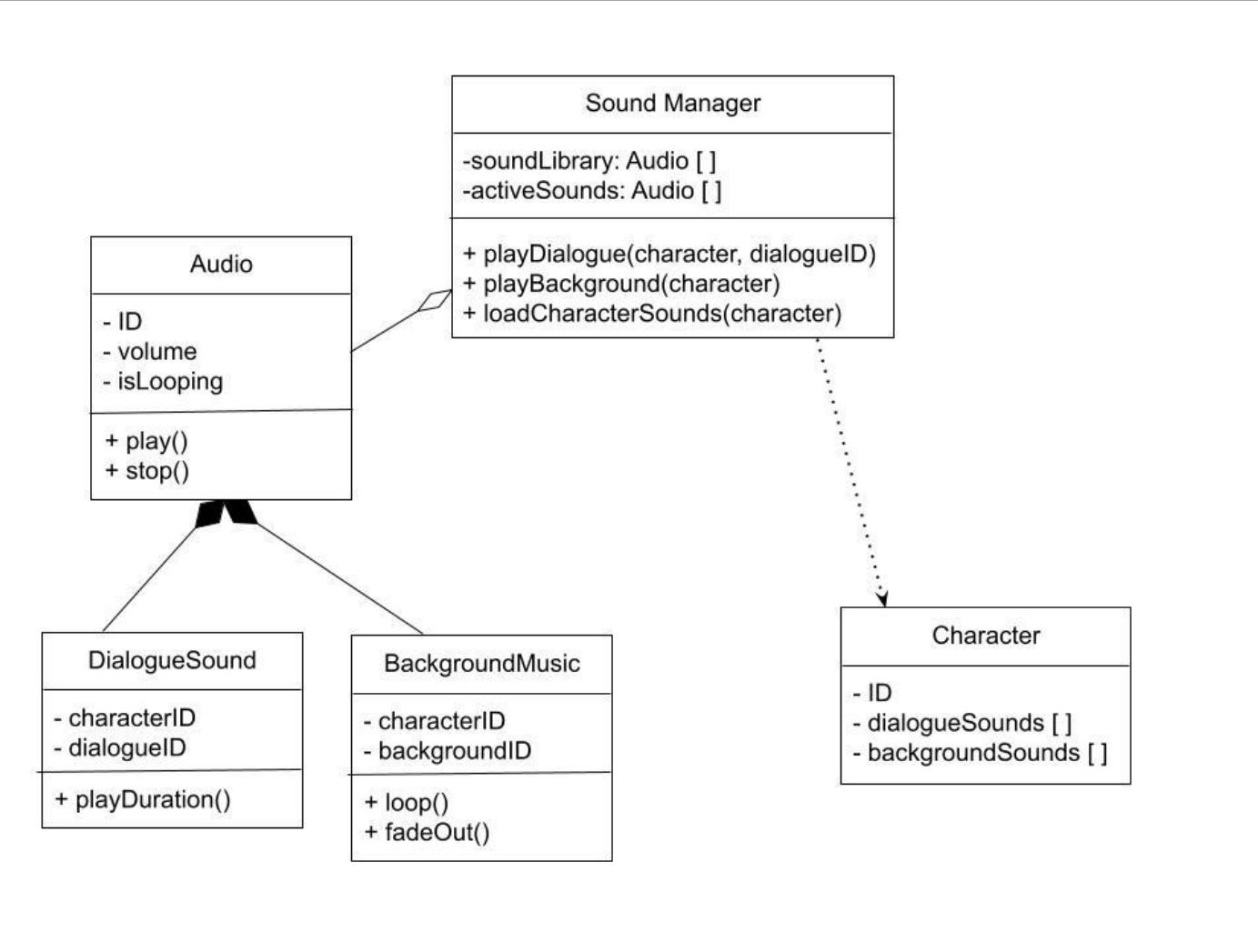


Penguine

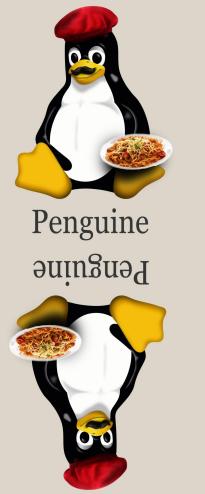
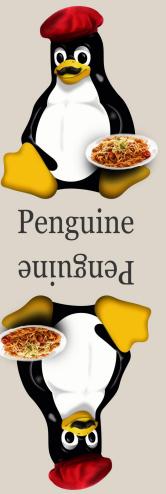
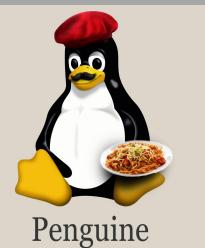


Penguine

# Process 7: Audio Class Diagram



Owen

Penguine  
PenguinePenguine  
Penguine

Penguine



Penguine

Any  
Questions?



Penguine