



# Building a Discord Game with Postgres

# Agenda

- About us
- Motivation behind the Project
- Tools we used
- How our game works
- What we experienced & learned
- Conclusion
- Demonstration

# About us

Caspar Klein

Student at the HU Berlin, majoring in  
Computer Science (4th Semester)

embedded software engineer (working  
student)

Xenia Kukushkina

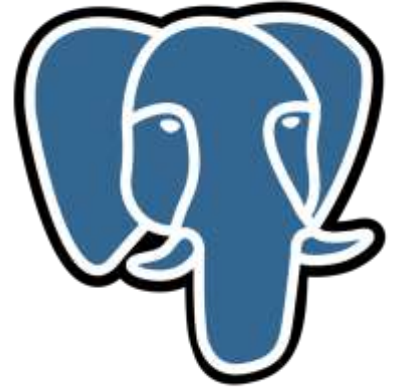
Student at the HTW Berlin majoring in  
Computer Science in Culture and Health (4th  
Semester)

working student in Data Analytics

# Motivation behind the project

- originally a Uni-Project for the *Databases Class*
  - what was needed was an ERD (Entity-Relationship-Diagram) and to build the database schema and show some sample SQL-queries
  - shared interest in gaming
  - personal interest in developing games
- 
- so we build a text based RPG (Role Play Game) with basic functionality
    - player combat
    - item gathering

# PostgreSQL



- it was given as the database system of choice by the university

I also asked our Professor why they decided to use PostgreSQL

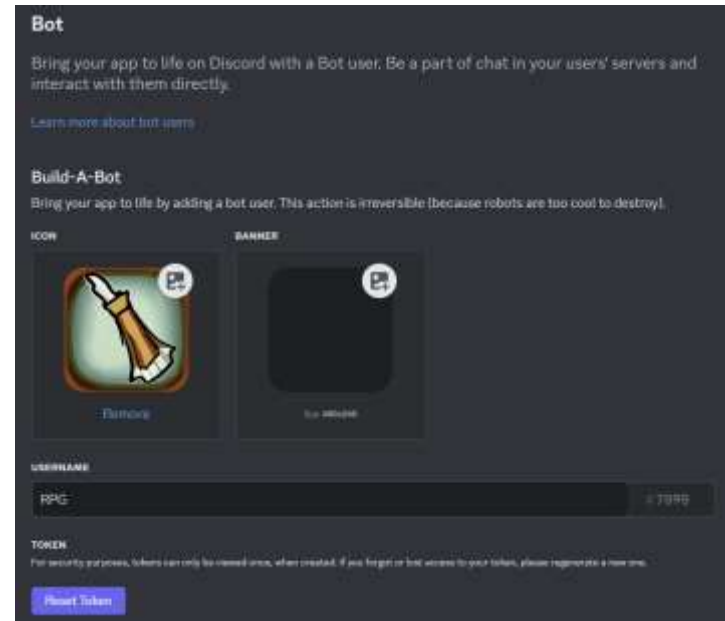
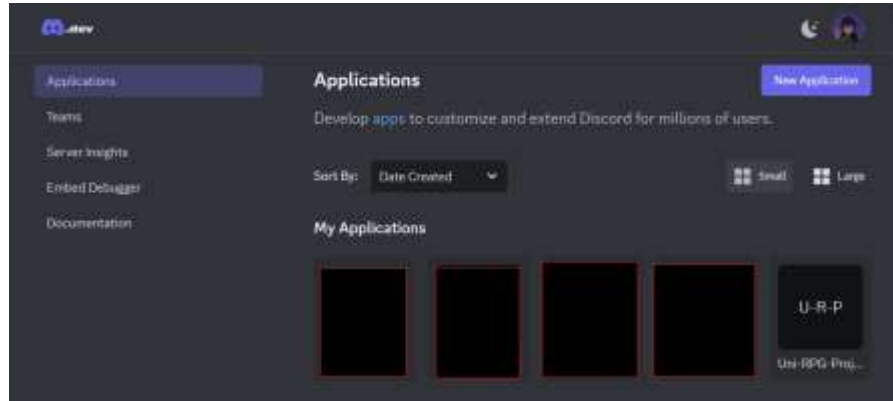
- Broadly used; many use cases thus nice to have for students
- Open Source, no licensing fees
- Can be easily hosted on campus

However the University Server has reached EOL

	version text	🔒
1	PostgreSQL 12.18 (Debian 12.18-1.pgdg110+2) on x86_64-pc-linux-gnu, compiled by gcc (Debian 10.2.1-6) 10.2.1 20210110, 6...	

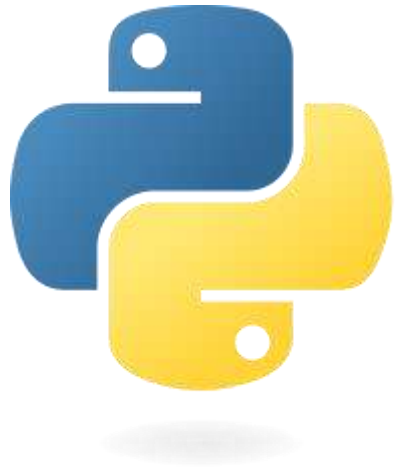
# Discord

- easy to use developer portal
- “free” hosting
- “build-in” multiplayer support



# Python

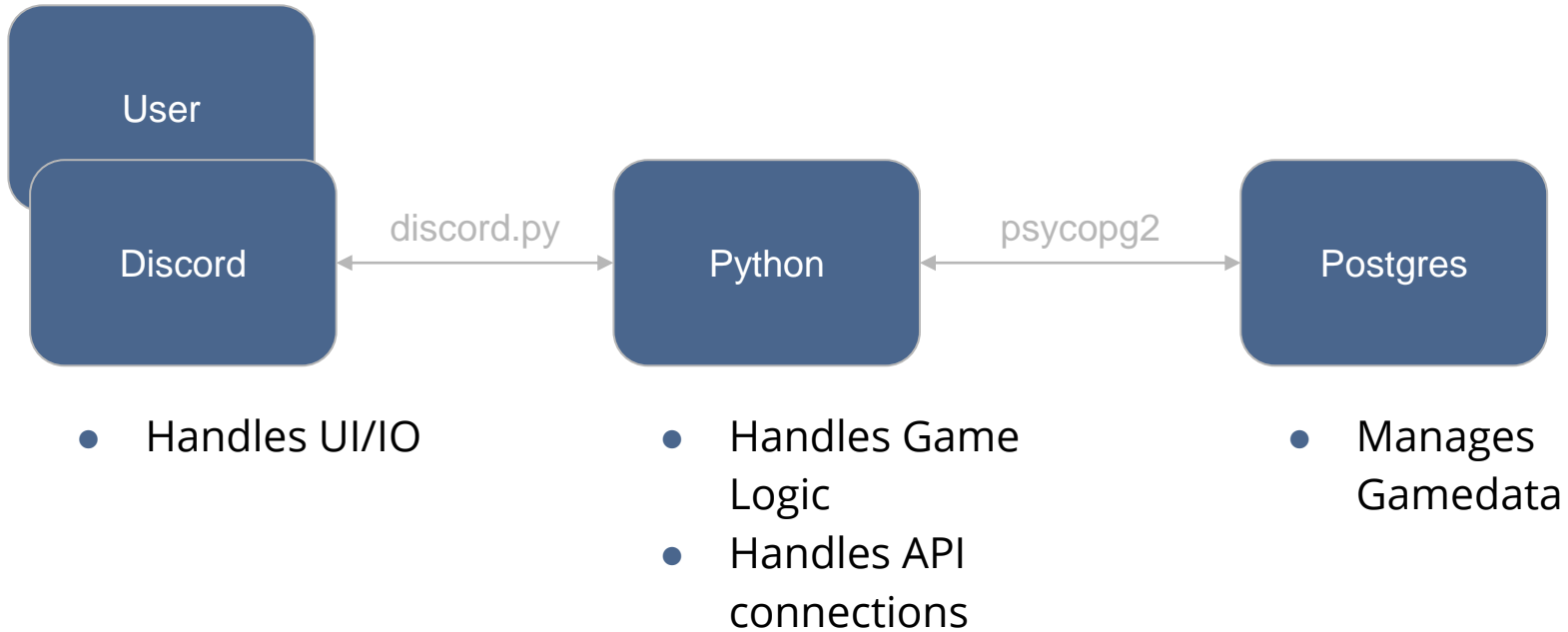
- the programming language most are familiar with
- easy to work with
- nice package library, we used `psycopg2` and `discord.py`



## Why psycopg2?

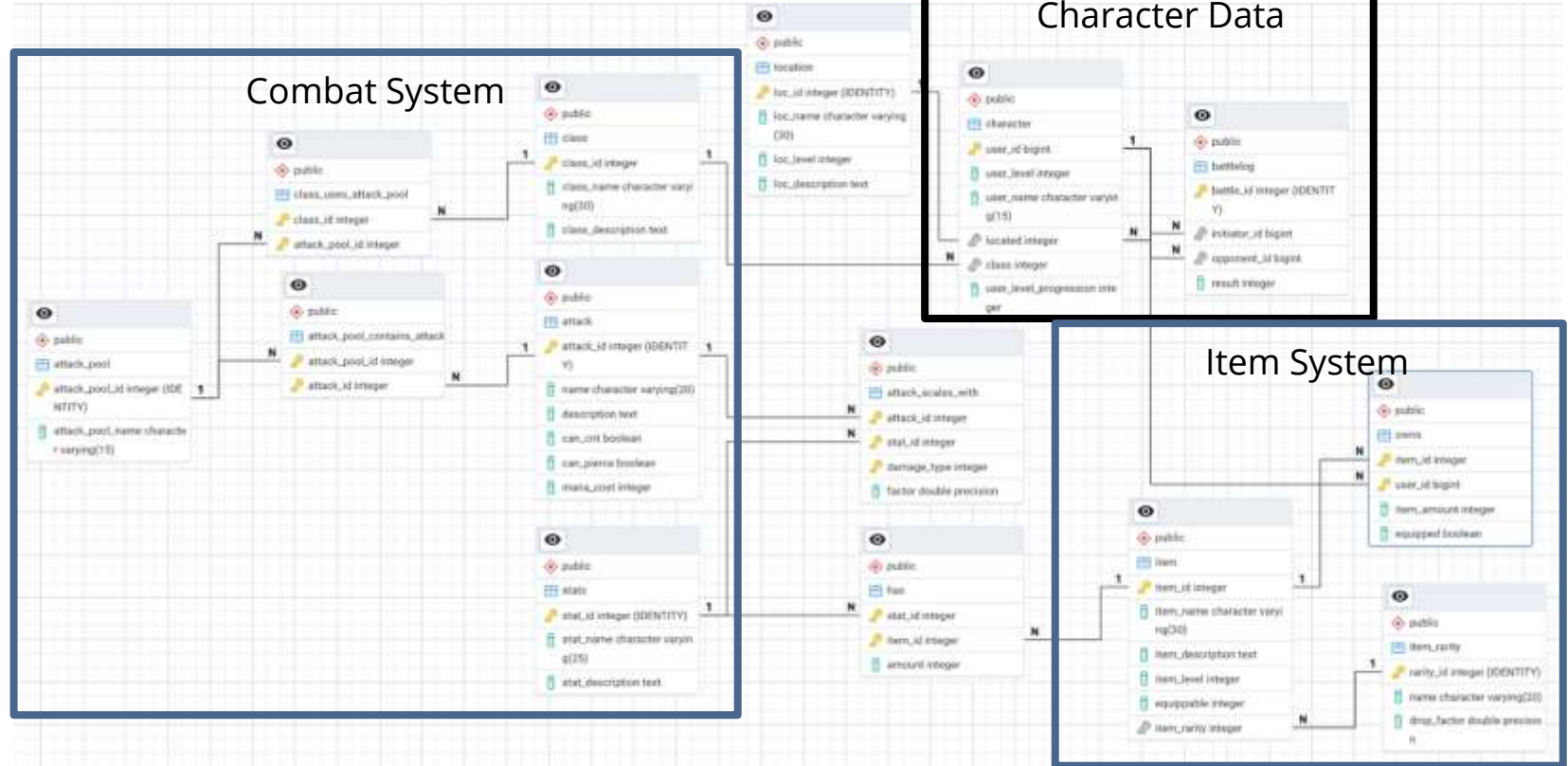
At the start of the project we googled how to connect yourself to the database with python - psycopg2 was recommended the most, so we used it. We know there is the more modern psycopg3 but it was too bothersome to change it :)

# API Integrations



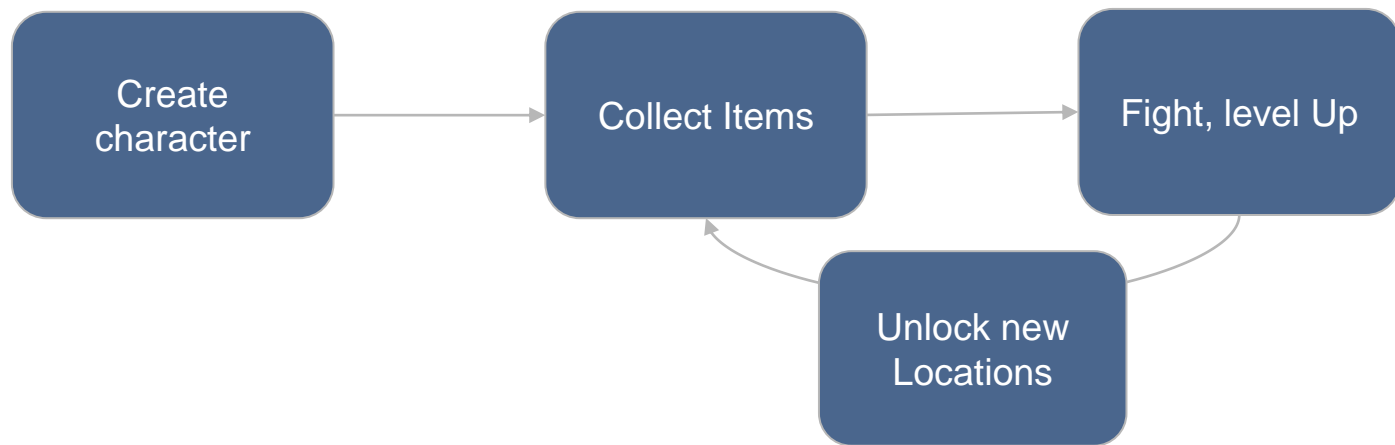


# Database Schema



# Gameplay

## Basic Gameplay loop



# Gameplay - Character Creation

`/create_character` Enter your name, press enter and choose a class afterwards. Start playing now!

`/create_character` desired\_character\_name

#

Warrior

A strong and brave fighter skilled with melee weapons, excell...

#

Mage

A master of magical arts, able to cast powerful spells to dam...

Wel


Rogue

A stealthy and agile character, proficient in sneaking, lock-pic...

This is th

Cleric

A devout healer who uses divine magic to aid allies and smite ...




Ranger

A skilled hunter and tracker, adept with bows and survival ski...

Only you can see this • Dismiss message

Message #rpg-game-channel

 RPG APP 20:03

RPG Nice, you will be known as Nami, after you choose your class: Behold the all mighty Nami! Wish them luck in these harsh lands

```
classes=[]
sheet=class_choice_view()
sheet.character_class.plugin=self
sheet.character_class.options=[]
#char_class = ui.Select(placeholder="Your Class",options=[])
#Getting Character Classes

if self.db_cur:
    self.db_cur.execute("SELECT c.class_id,c.class_name,c.class_description FROM class c")
    classes=self.db_cur.fetchall()
for i in classes:
    sheet.character_class.append_option(discord.SelectOption(label=i[1],description=shorten_description(i[2]),value=i[0]))


sheet.add_item(sheet.character_class)


# Save name for later use
data = load_json()
if not "open_creation_prompts" in data:
    data["open_creation_prompts"]={}
data["open_creation_prompts"][str(ctx.user.id)]=desired_character_name
dump_json(data)


await ctx.response.send_message(f"Nice, you will be known as {desired_character_name}, after you choose your class.",view=sheet,ephemeral=True)

data = load_json()
if "open_creation_prompts" in data:
    if str(interaction.user.id) in data["open_creation_prompts"]:
        name = data["open_creation_prompts"][str(interaction.user.id)]
        del data["open_creation_prompts"][str(interaction.user.id)]
        dump_json(data)
        self.plugin.db_cur.execute(f"INSERT INTO character VALUES (%s,1,%s,0,%s)",(str(interaction.user.id),name,int(interaction.data['values'][0])))
        self.plugin.db.commit()
        weird_announcements=["They are still young, but will conquer the world sooner or later.", "Keep an eye on them or they will throw you from your"]
        await interaction.response.send_message(f"Behold the all mighty {name}! "+weird_announcements[random.randint(0,len(weird_announcements)-1)])
    else:
        await interaction.response.send_message("Something went wrong. Maybe refer to an admin if it happens again.")
```


# Gameplay - Item Gathering


 **/scavage**  
Search for items at your current location. This has a 5min cooldown.


 **/scavage**


 ShyguyBerlin used **/scavage**  
RPG APP 30.39  
Casparius has found an item:  

**Trolling Trombone**  
A trombone that plays prank sounds instead of music notes. Musical mischief.

 Only you can see this • [Dismiss message](#)

 ShyguyBerlin used **/scavage**  
RPG APP 30.39  
You cannot loot the location now. You have 294s left on the cooldown.  

 Only you can see this • [Dismiss message](#)



 | Message #rpg-game-channel


```
if self.db_cur:
    self.db_cur.execute(f"SELECT * FROM valid_item_pools WHERE user_id=%s", (ctx.user.id,))
    db_item_pool=self.db_cur.fetchall()
    for i in range(len(db_item_pool)):
        level_weights=[0.4,0.7,1,0.2]
        print(db_item_pool[i])
        item_pool.append((i,level_weights(min(3,max(0,db_item_pool[i][6]*2)))*db_item_pool[i][5]))
    summed_weight=sum([x[i] for x in item_pool])
    rand_choice=random.random()*summed_weight
    chosen=None
    for i in range(len(item_pool)):
        if rand_choice<item_pool[i][1]:
            chosen=db_item_pool[item_pool[i][0]]
            break
        rand_choice-=item_pool[i][1]
    if chosen:
        print(chosen)
        #The selected item is already present in the players inventory
        if chosen[7]!=None:
            self.db_cur.execute(f"UPDATE owns SET item_amount=%s WHERE item_id=%s and user_id=%s", (chosen[7]+1,chosen[2],chosen[0]))
            self.db.commit()
            #The selected item is new to this player
        else:
            self.db_cur.execute(f"INSERT INTO owns VALUES (%s,%s,%s,%s)", (chosen[2],chosen[0],1,False))
            self.db.commit()


        item_embed=discord.Embed(title=chosen[3],description=chosen[8])


        await ctx.response.send_message(f"(chosen[1]) has found an item!",embed=item_embed,ephemeral=True)
    else:
        await ctx.response.send_message(f"I, the bot, am not able to randomly select an item, sorry",ephemeral=True)
    else:
        await ctx.response.send_message(f"I, the bot, am not able to connect to the database, sorry",ephemeral=True)
```


# Gameplay - Combat System

  **/scout\_battle** Start looking for some combat opportunities at your location.


 **/challenge\_player** Challenge a fellow player to a one vs one.


 **/challenge\_player** target\_player **Xeni**


 ShyguyBerlin used **/scout\_battle**  
RPG APP 18:11  
Xeni-Bot was already scouting at Blossom Fields. Casparius is challenging them!

 RPG APP 18:11  
Battle emerges between Xeni-Bot and @ShyguyBerlin! Pick your favorite and cheer them on!

**/use\_move** Enter the number of the attack you've found in /my\_combat\_situation.


 **/use\_move** move **1**

 ShyguyBerlin used **/use\_move**  
RPG APP 14:52  
Crit! Casparius used 'Arcane Trap' and dealt 265.1 Physical Damage + 530.63 Magical Damage damage to and thus killed Xeni-Bot. Casparius wins the combat!

 ShyguyBerlin used **my\_combat\_situation**  
RPG APP 18:11  
You are in a fight with Xeni.

**Your possible Actions:**

- 1: Arcane Trap - Let the energy of the Arcana flow through you and burst into your enemy.
  - Deals magic damage
    - Intelligence: 60%, Cosmic Blessing: 60%, Critical Hit Chance: 30%, Armor: 20%
  - Can critical strike: 50.86% (depends on your crit chance) and pierce through armor/ willpower by roughly 2.0 points of resistance
- 2: Curse - Dominate your foe with your mental supremacy
  - Deals magic damage
    - Intelligence: 95%, Willpower: 30%, Mana: 25%, Luck: 10%
- 3: Heavy Hit - Stab your weapon into the enemy using your own kinetic energy
  - Deals physical damage
    - Health: 100%, Strength: 15%, Armor: 45%
  - Can pierce through armor/ willpower by roughly 2.0 points of resistance

 Only you can see this • [Dismiss message](#)

# Problems and Observations

## pgAdmin 4

- beginner friendly
- easy way to create tables
- useful to get an overview of our database schema
- shows all options available to you

## Console commands

- more advanced commands (like getting the database dump)

# Problems and Observations

- Many available data types

Table "public.character"				
Column	Type	Collation	Nullable	Default
user_id	bigint		not null	
user_level	integer		not null	1
user_name	character varying(15)		not null	
located	integer		not null	0
class	integer		not null	0
user_level_progression	integer		not null	0

Indexes:  
"User\_pkey" PRIMARY KEY, btree (user\_id)  
"fki\_user\_loc\_id\_fkey" btree (located)

Foreign-key constraints:  
"user\_class\_id\_fke" FOREIGN KEY (class) REFERENCES class(class\_id)  
"user\_loc\_id\_fkey" FOREIGN KEY (located) REFERENCES location(loc\_id)

- Nullable fields or default values

Table "public.attack"				
Column	Type	Collation	Nullable	Default
attack_id	integer		not null	generated by default as identity
name	character varying(20)			
description	text			
can_crit	boolean		not null	false
can_pierce	boolean		not null	false
mana_cost	integer			

Indexes:  
"attack\_pkey" PRIMARY KEY, btree (attack\_id)



# Problems and Observations

- Read the docs!
- Secure queries



```
self.db_cur.execute(f"SELECT loc_id, loc_name FROM location WHERE loc_level<={player.level} and loc_name='{str(target_location)}'")
if self.db_cur.rowcount==0:
    await no_loc_found()
    return
loc=self.db_cur.fetchone()
```



```
self.db_cur.execute(f"SELECT loc_id, loc_name FROM location WHERE loc_level<=%s and loc_name=%s",(player.level,str(target_location)))
if self.db_cur.rowcount==0:
    await no_loc_found()
    return
loc=self.db_cur.fetchone()
```



# Problems and Observations

- Privilege Management

List of roles			
Role name	Attributes	Member of	Description
postgres	Superuser, Create role, Create DB, Replication, Bypass RLS	{}	
rpg_bot		{}	
rpg_bot_testing		{}	

```
--  
-- Name: TABLE "character"; Type: ACL; Schema: public; Owner: postgres  
--  
  
GRANT SELECT,INSERT,DELETE,UPDATE ON TABLE public."character" TO rpg_bot;
```

# Conclusion

Overall: fun project to learn how to write a small application with databases

If you want to play around with the bot on discord: [QR/Link]



discord

you can also host it yourself, just follow the readme in the repository

- ([https://github.com/xen1i/RPG\\_Project](https://github.com/xen1i/RPG_Project))



github



# Demonstration + Questions

