
ROZVOJ SOFTVÉROVÝCH FUNKCIÍ VO VIRTUÁLNEJ REALITE

Bc. Matúš Granec

Ing. Juraj Vincúr, PhD.

doc. Ing. Ivan Polášek, PhD.



MOTIVÁCIA

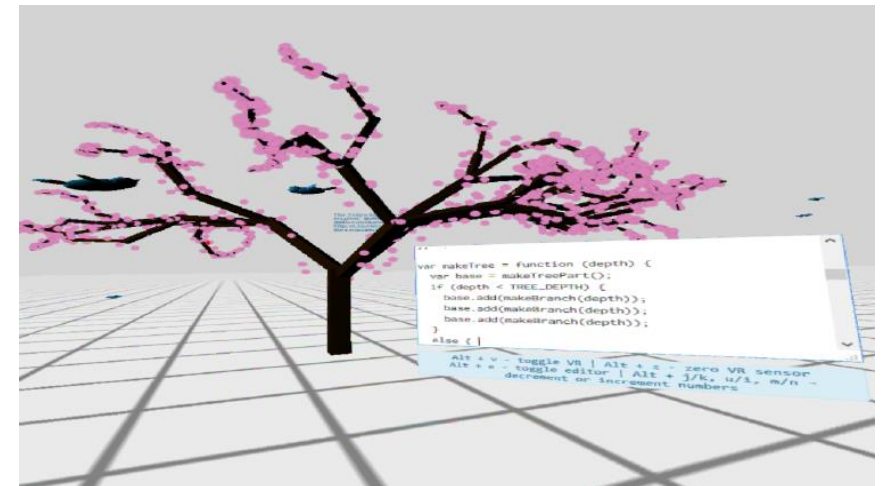
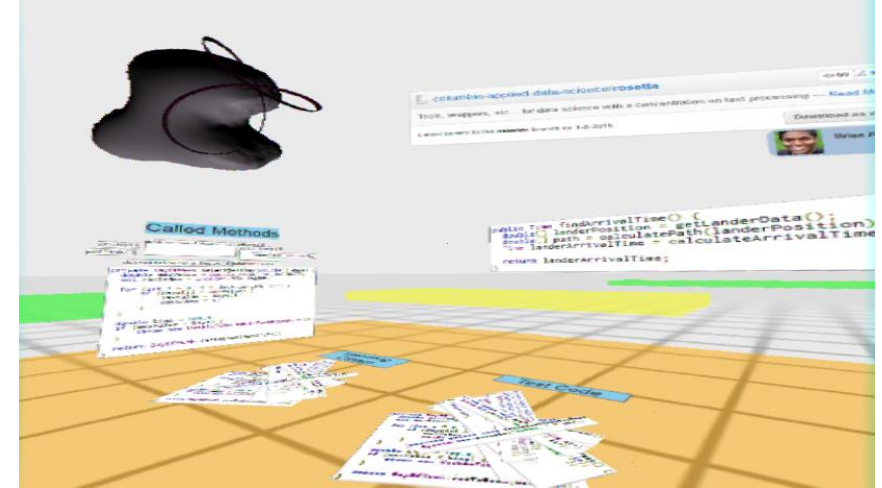
- Obmedzený priestor 2D obrazoviek
- Pomalá navigácia v kóde
- Porozumenie kódu
- Preklikávanie jednotlivých okien s kódom
- Obmedzený pohľad na program cez súbory so zdrojovým kódom

CIEĽ PRÁCE

- Analýza existujúcich podporných nástrojov a prostredí určených na vývoj softvéru
- Analýza problémov, nedostatkov a návrhov vylepšení existujúcich prístupov
- Návrh a implementácia vývojového prostredia pre virtuálnu alebo rozšírenú realitu
- Kombinácia klasického 2D vývojového prostredia s riešením v 3D priestore
- Využiť možnosti 3D priestoru na zrýchlenie navigácie v kóde a pochopenia zdrojového kódu
- Poskytnúť rôzne pohľady na zdrojový kód, triedy, vzťahy medzi objektami a pod.
- Navrhnite rozvoj tela metód na základe AI alebo VR-native
- Vyhodnotenie prínosov navrhnutého riešenia

VIRTUAL REALITY IN SOFTWARE ENGINEERING: AFFORDANCES, APPLICATIONS, AND CHALLENGES¹

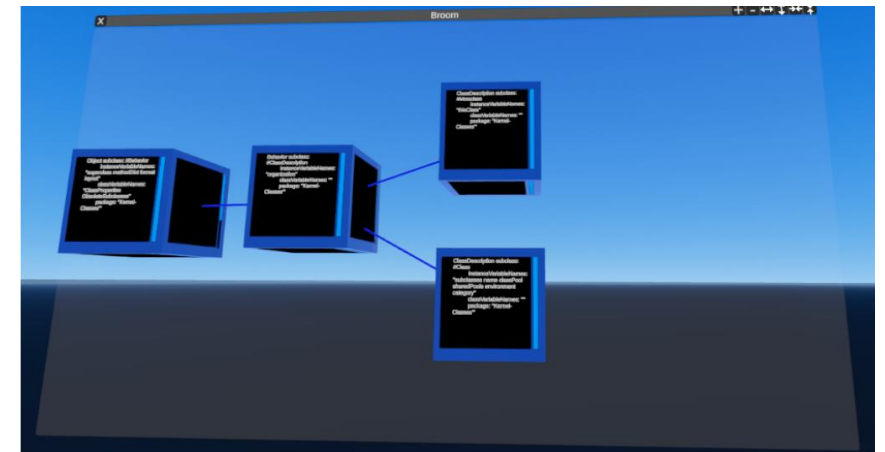
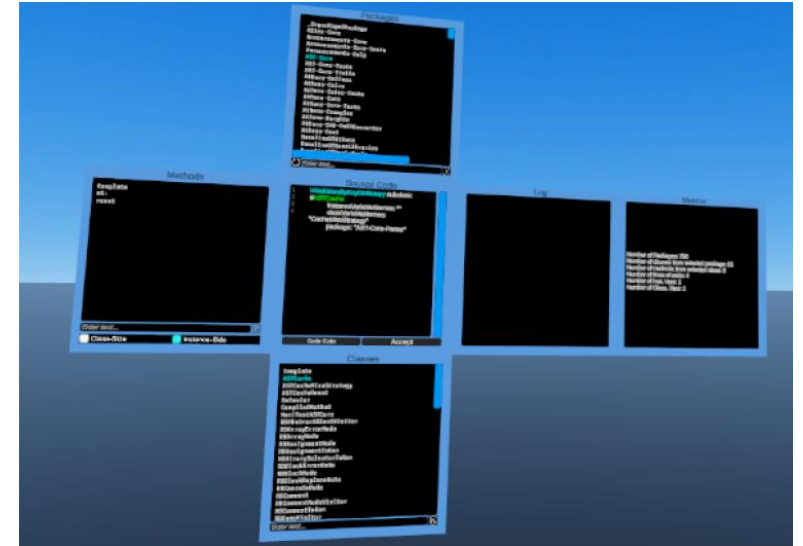
- Využíva priestorovú pamäť, manipuláciu a okamžitú spätnú väzbu
- Vyššia produktivita a spokojnosť užívateľov.
- Code Review (IMMERSSION): Vizualizácia kódu ako 3D objektov a hromád pre jednoduchšie hodnotenie
- Live Coding (RIFTSKETCH): Prostredie pre písanie kódu s okamžitou vizualizáciou zmien v 3D.
- Oddelenie od fyzického prostredia a obmedzenia technológie
- Potenciál zmeniť spôsob, akým sa softvér vyvíja a recenzuje, najmä pri riešení problémov v reálnom čase a v tímovej spolupráci.



¹) Anthony Elliott, Brian Peiris, and Chris Parnin. Virtual reality in software engineering: Affordances, applications, and challenges. In 2015 IEEE/ACM 37th International Conference on Software Engineering, volume 2, pages 547–550, 2015.

VR-BASED USER INTERACTIONS TO EXPLOIT INFINITE SPACE IN PROGRAMMING ACTIVITIES¹

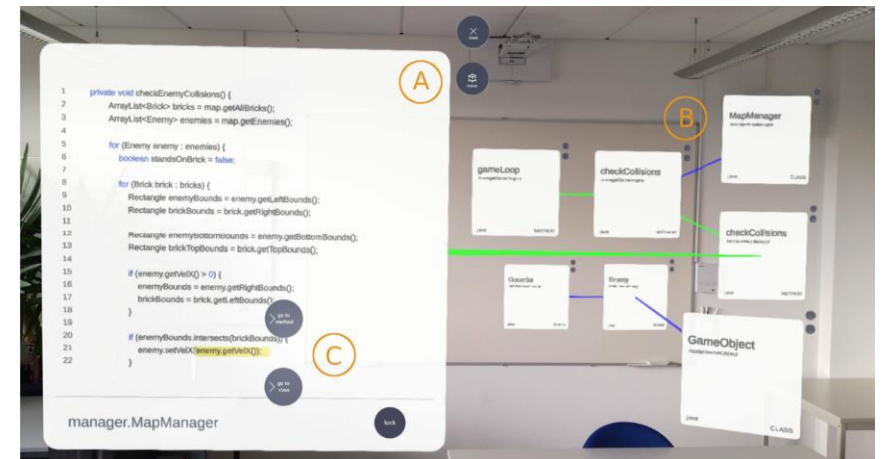
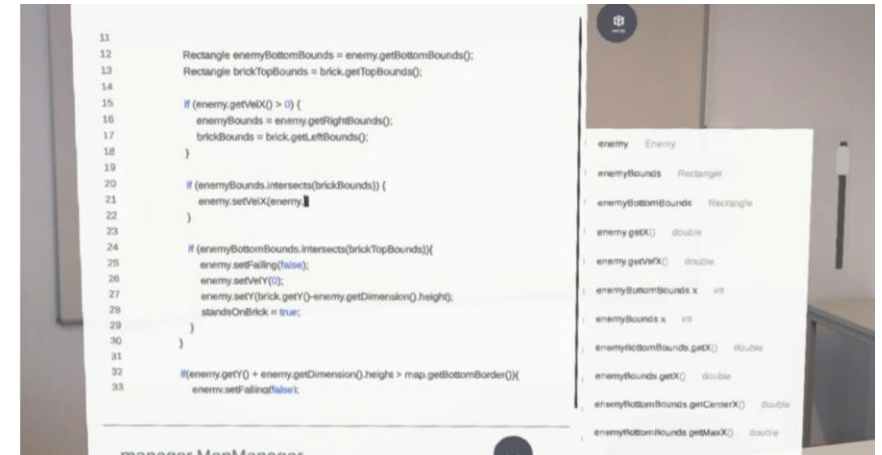
- Metafora kocky – Code cube
- Výhody:
 1. Rôzne pohľady na jeden objekt/triedu pomocou Code cube
 2. Využitie nekonečného 3D priestoru
- Nevýhody:
 1. Zadávanie dlhšieho textu
 2. Odrezanie od reálneho sveta
 3. Obmedzenia pri riešení komplexných problémov



¹) Castillo, Victor & Merino, Leonel & Hecht, Geoffrey & Bergel, Alexandre. (2021). VR-Based User Interactions to Exploit Infinite Space in Programming Activities. 1-5. 10.1109/SCCC54552.2021.9650396.

IDEVELOPAR: A PROGRAMMING INTERFACE TO ENHANCE CODE UNDERSTANDING IN AUGMENTED REALITY¹

- IDEVELOPAR
- Výhody:
 1. Lepšie porozumenie zložitým štruktúram v kóde
 2. Potenciálne zvýšenie produktivity vďaka intuitívnej vizualizácii kódu
- Nevýhody:
 1. Hardvérové obmedzenia
 2. Project Northstar gestá môžu byť neintuitívne pre používateľov



1) L. Kreber, S. Diehl and P. Weil, "IDEvelopAR: A Programming Interface to enhance Code Understanding in Augmented Reality," 2022 Working Conference on Software Visualization (VISOFT), Limassol, Cyprus, 2022, pp. 87-95, doi: 10.1109/VISOFT55257.2022.00017. keywords: {Visualization;Software maintenance;Codes;Navigation;Statistical analysis;User interfaces;Usability;augmented reality;programming tool;program comprehension;code navigation},

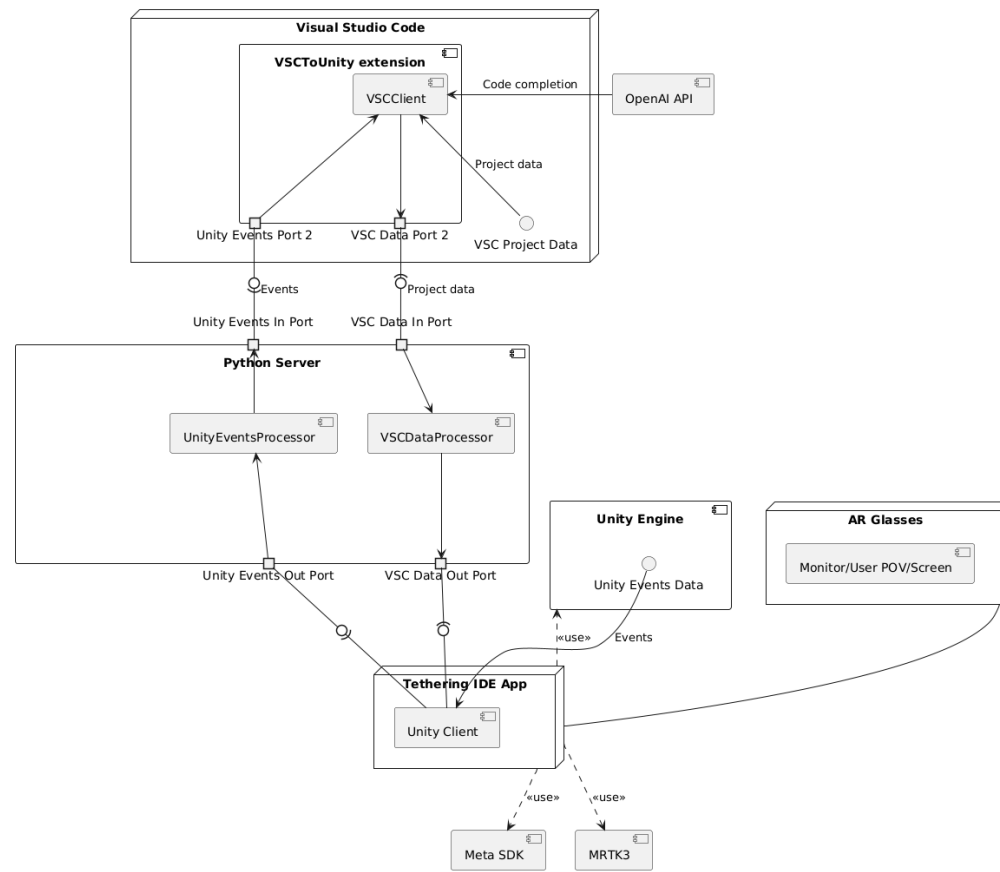
MOMENTÁLNE PROBLÉMY PRÍSTUPOV

- Vkládanie/písanie dlhších textov
- Únava rúk pri dlhšom programovaní
- Pomalšie riešenie programátorských problémov ako pri použití PC
- Neexistuje natívne vývojárske prostredie pre rozšírenú alebo virtuálnu realitu
- Prepínanie kontextu (VR prostredie -> PC a naopak)
- Odrezanie používateľa od reálneho sveta

NÁŠ PRÍSTUP

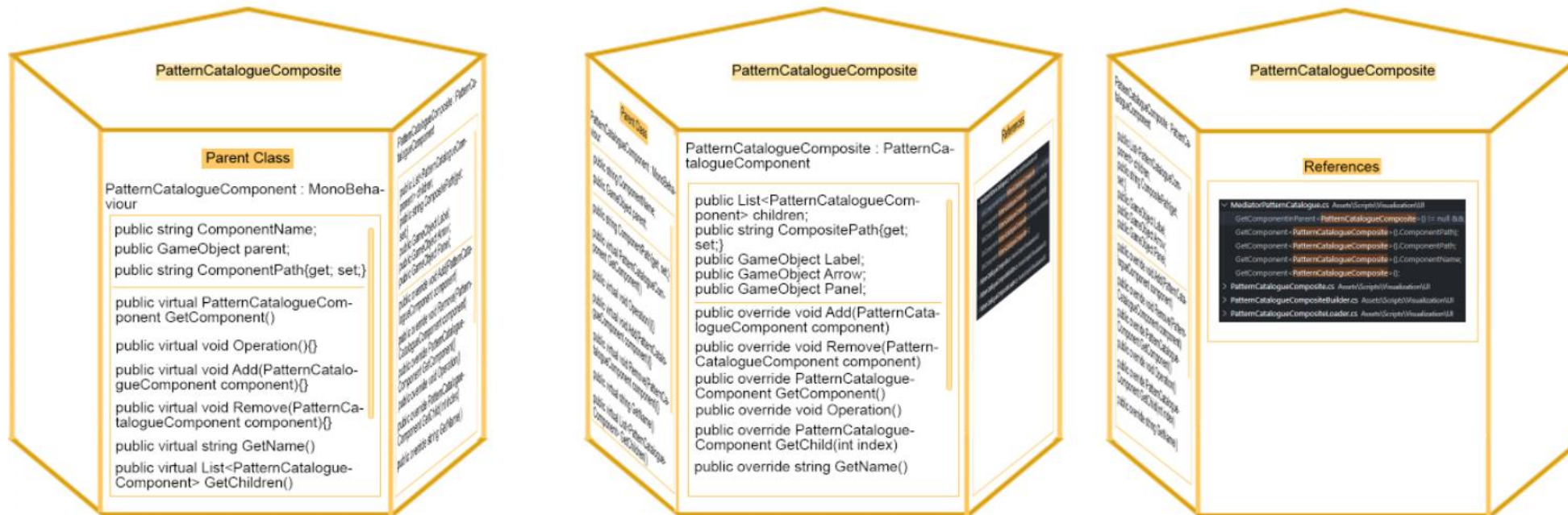
- AR Unity aplikácia v kombinácii s klasickým programovacím prostredím
- Vytvoríme vizuálnu metaforu pre triedy - Code Box
- HoloLens na zobrazenie virtuálneho prostredia
- Písanie kódu vo Visual Studio Code (monitor, klávesnica a myš)
- AI asistent na písanie kódu

KOMPONENT DIAGRAM APLIKÁCIE

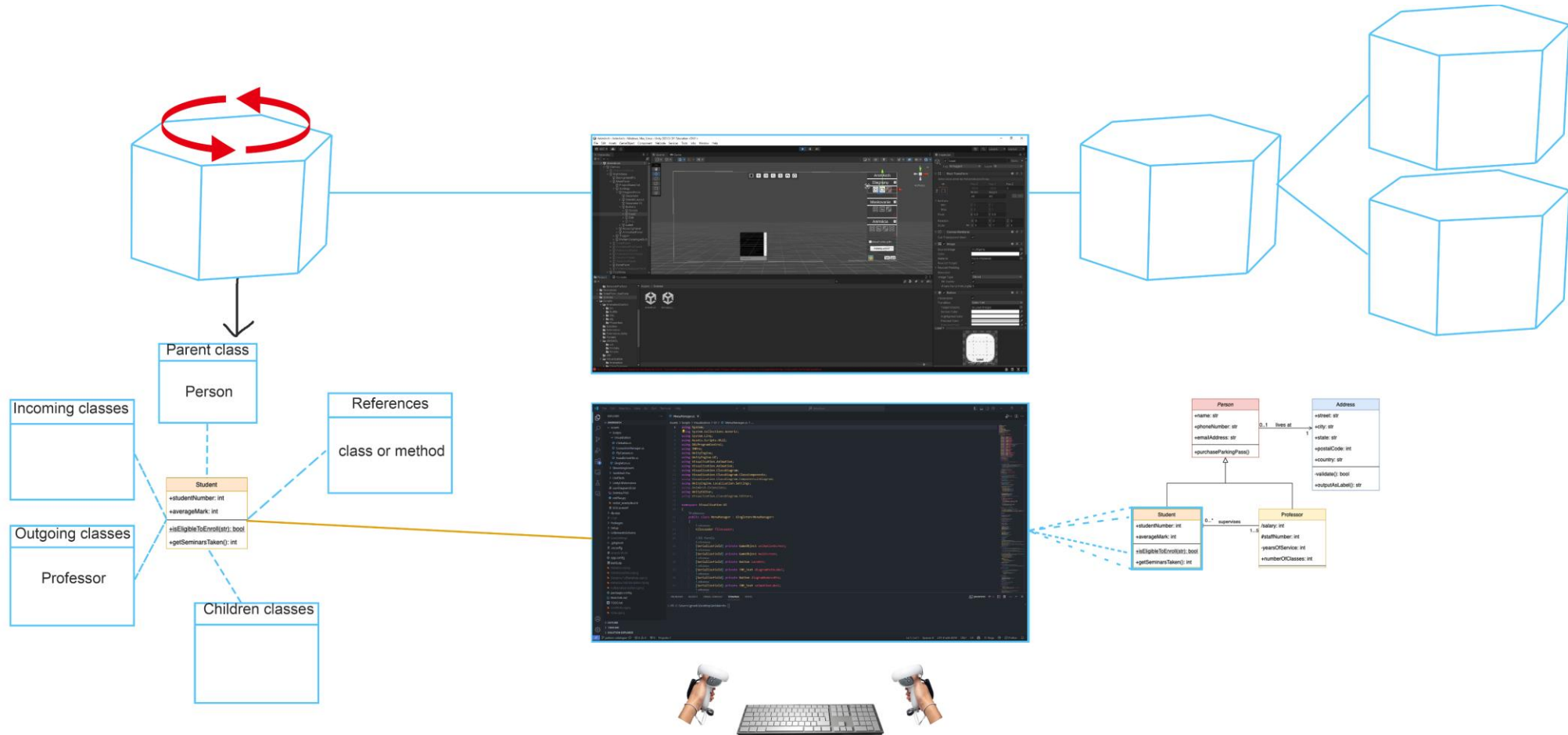


- Zariadenie – Project Northstar
- Aplikácia beží na PC
- Project Northstar slúži na zobrazovanie a ovládanie 3D priestoru
- Komunikácia medzi Unity a VSC zastrešená Python serverom
- VSC extension využíva OpenAI API na generovanie kódu

PŮVODNÝ NÁVRH PRE CODE BOX



PÔVODNÝ NÁVRH AR PROSTREDIA

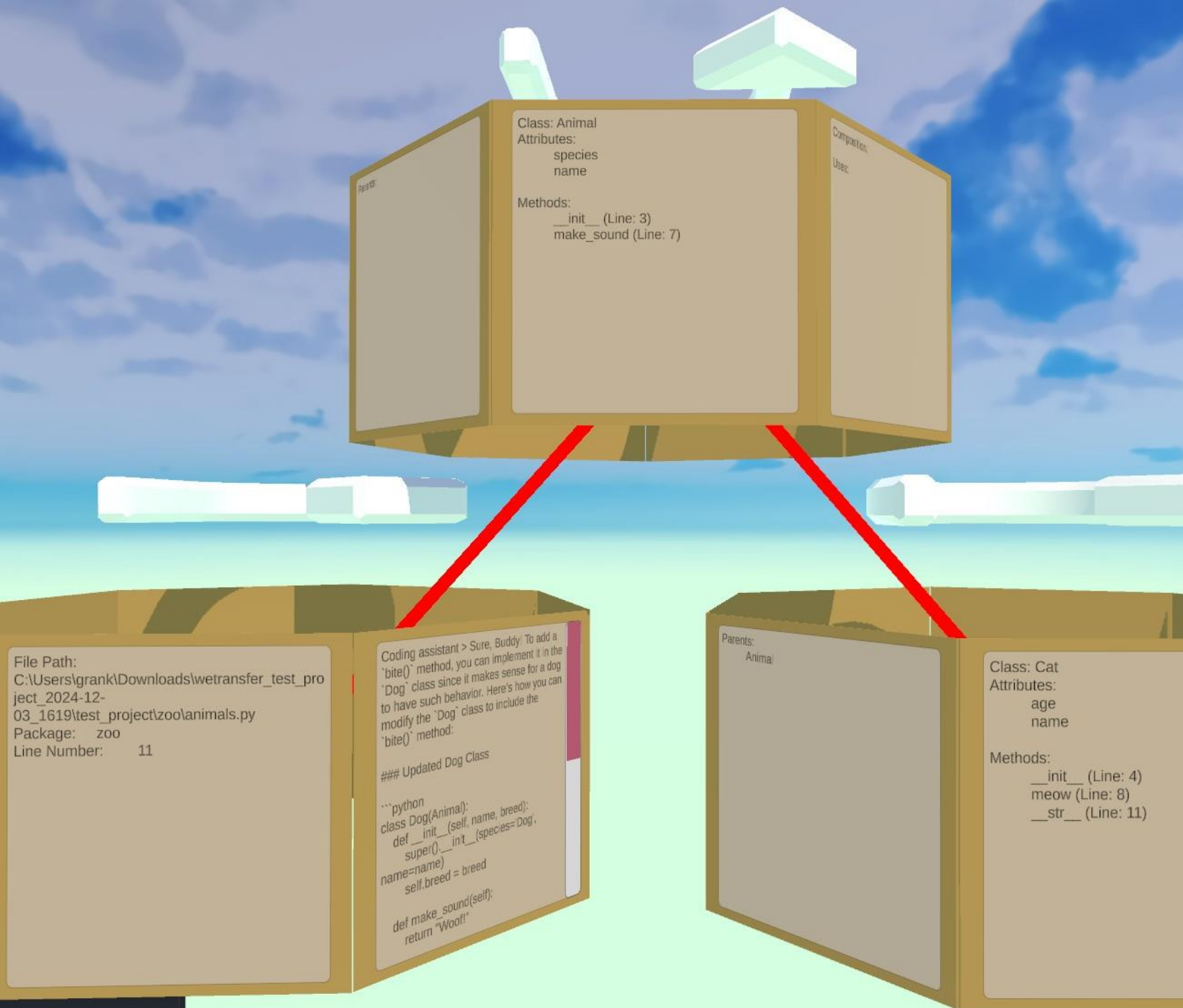


ŠTRUKTÚRA DÁT

- Dátová reprezentácia triedy vo formáte JSON
- Výstup automatickej analýzy pomocou Python AST
- Atribúty, metódy, vzťahy
- Cesta k zdrojovému kódu
- Číslo riadkov ku každej metóde – interakcie

```
{  
  "Movie": {  
    "file_path": "c:\\path\\to\\cinema.py",  
    "package": "cinema",  
    "line_number": 10,  
    "methods": [  
      {  
        "name": "__init__",  
        "line_number": 11  
      }  
    ],  
    "attributes": [  
      "duration",  
      "title"  
    ],  
    "base_classes": [],  
    "composition": [],  
    "uses": []  
  }  
}
```

Figure 4.3: JSON Representation of the Movie Class



CODE BOX

- 3D vizualizácia dát automatickej analýzy
- Vizualizácia vzťahov
- Zobrazenie histórie otázok a odpovedí AI programovacieho asistenta

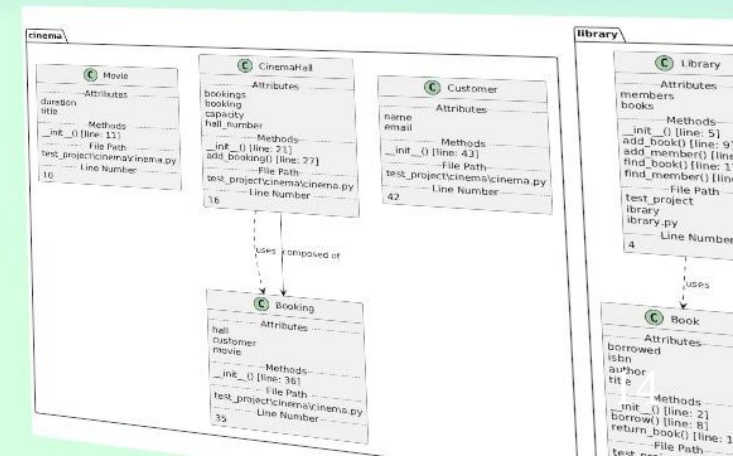
PROTOTYP APLIKÁCIE V UNITY

The screenshot shows a PyCharm IDE with a Python project named 'test_project'. The file explorer on the left lists files such as 'PROJECT', 'ma', 'init_.py', 'ema.py', 'ry', 'nit_.py', 'k.py', 'ary.py', 'mber.py', 'als', 'py', 'als.py', 'm.json', 'm.puml', 'y', and '.py'. The main editor displays the 'library.py' file, which contains the following code:

```

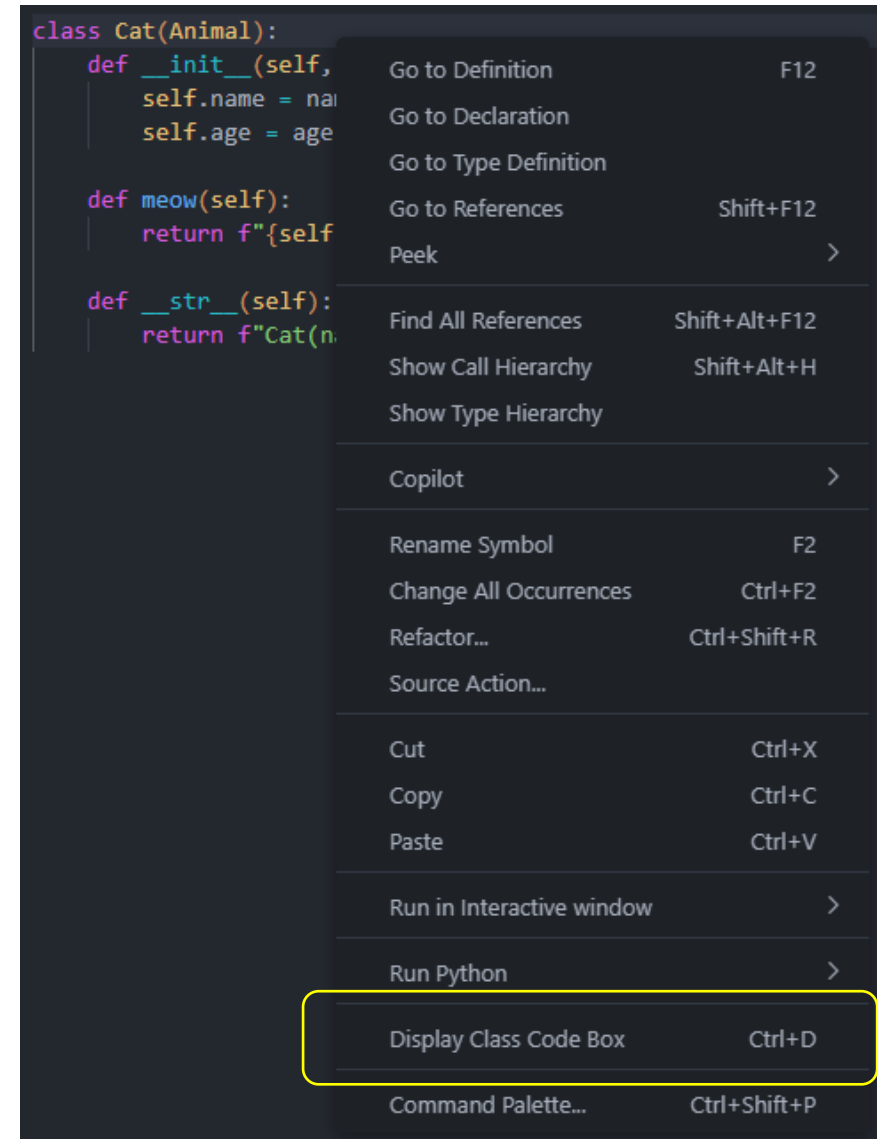
1  from .book import Book
2  from .member import Member
3
4  class Library:
5      def __init__(self):
6          self.books = []
7          self.members = []
8
9      def add_book(self, title, author, isbn):
10         book = Book(title, author, isbn)
11         self.books.append(book)
12
13     def add_member(self, name, member_id):
14         member = Member(name, member_id)
15         self.members.append(member)
16
17     def find_book(self, isbn):
18         for book in self.books:
19             if book.isbn == isbn:
20                 return book
21         return None
22
23     def find_member(self, member_id):
24         for member in self.members:
25             if member.member_id == member_id:
26                 return member
27         return None

```



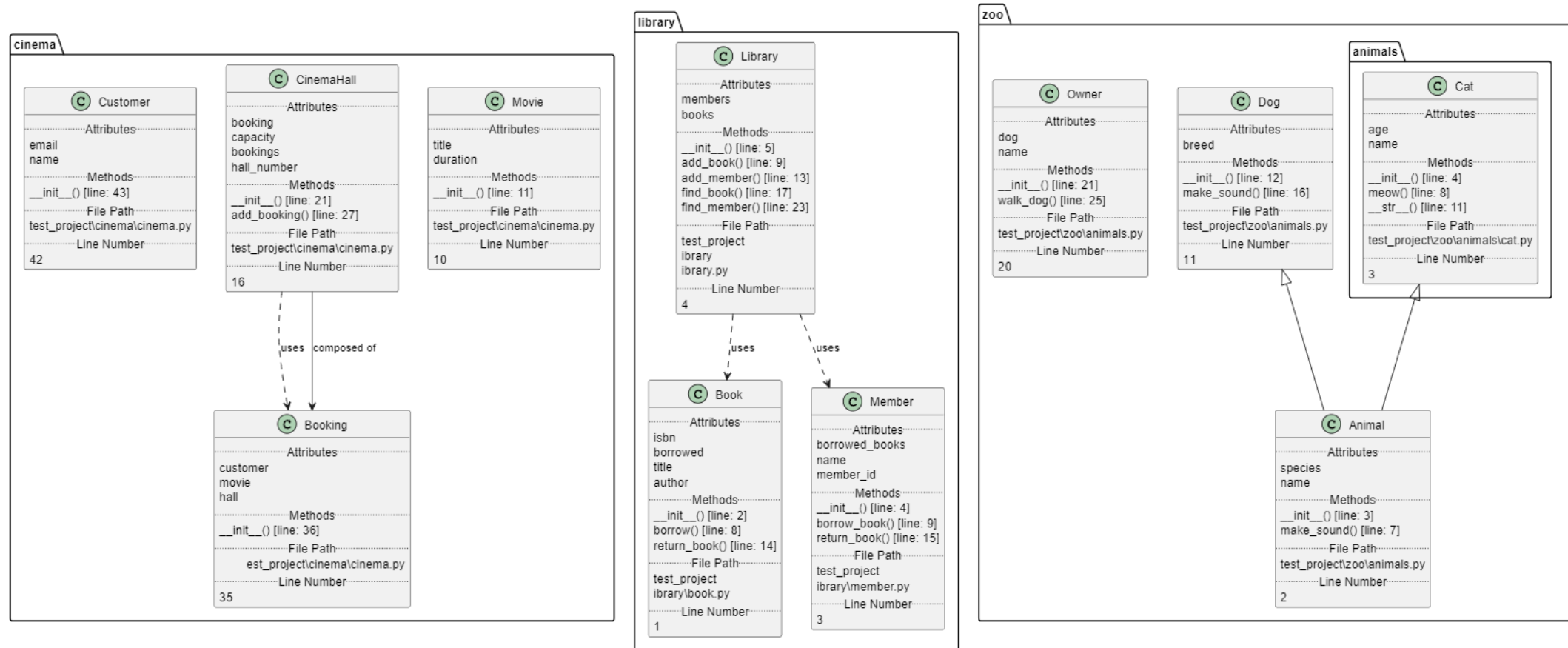
VISUAL STUDIO CODE EXTENSION

- Analýza kódu pomocou Python AST
- Výstup v JSON a PlantUML formáte
- Real-Time synchronizácia kódu s vizualizáciou
- Vysvietenie zvolených/upravovaných tried vo vizualizácii
- Integrácia OpenAI asistenta na generovanie nových metód s poskytnutým kontextom
- Posielanie/príjmanie správ z Python servera na komunikáciu s Unity aplikáciou
- Kontextové menu, klávesové skratky



PLANTUML VÝSTUP

- Vizualizácia projektu v PlantUML
- Vygenerované z JSON súboru



KOMUNIKÁCIA MEDZI VISUAL STUDIO CODE A UNITY APLIKÁCIOU

- Websocket pripojenie cez python server
- Obojstranná komunikácia s Unity aplikáciou
- Odosielanie dát o projekte/triedach
- Prijímanie udalostí z Unity aplikácie (kliknutie na metódu, ...)

```
api.py > ...
1 import asyncio
2 import websockets
3 from command import *
4 connected_clients = set()
5
6 async def register(websocket):
7     connected_clients.add(websocket)
8     await handle_client(websocket, connected_clients)
9
10 async def start_server(host='localhost', port=8765):
11     server = await websockets.serve(register, host, port)
12     print(f"Server started at ws://{host}:{port}")
13     await server.wait_closed()
14
```

```
export function activate(context: vscode.ExtensionContext) {
    const ws = new WebSocket('ws://localhost:8765');

    ws.on('open', () => {
        vscode.window.showInformationMessage('Connected to Python service!');
        ws.send('get_line_number');
    });

    ws.on('message', (data: string) => {
        console.log(`Received message: ${data}`);
        vscode.window.showInformationMessage(`Python service says: ${data}`);
    });

    vscode.window.onDidChangeActiveTextEditor((event) => {
        const document = event?.document;
        if (document == null || document.isUntitled) {
            return;
        }
        const filePath = document.uri.fsPath;
        vscode.window.showInformationMessage(`File changed: ${filePath}`);
        ws.send(`get_document_path:${filePath}`);
    });

    vscode.window.onDidChangeTextEditorSelection((event) => {
        const lineNumber = event.selections[0].active.line;
        ws.send(`get_line_number:${lineNumber + 1}`);
    });
}
```

```
0 references
public class NetworkListener : MonoBehaviour
{
    6 references
    private Socket server;
    4 references
    private IPAddress localAddr;
    3 references
    private int port;

    0 references
    private void Start()
    {
        localAddr = IPAddress.Parse("127.0.0.1");
        port = 7777;
        IPEndPoint localEndPoint = new IPEndPoint(localAddr, port);

        server = new Socket(localAddr.AddressFamily, SocketType.Stream, ProtocolType.Tcp);

        try
        {
            server.Bind(localEndPoint);
            server.Listen(10);

            Debug.Log($"Server started on {localAddr}:{port}");

            server.BeginAccept(new AsyncCallback(AcceptCallback), server);
        }
        catch (Exception e)
        {
            Debug.Log(e.ToString());
        }
    }
}
```

HYPOTÉZY

- **Zlepšenie pochopenia kódu:**
Vývojári získajú hlbšie pochopenie štruktúry a vzťahov v kóde vďaka priestorovej a vizuálnej reprezentácii tried a ich interakcií.
- **Zvýšenie efektivity navigácie:**
Možnosť priamo interagovať s kódovými blokmi v AR a preskakovať na relevantné súbory alebo riadky kódu zníži čas potrebný na lokalizáciu konkrétnych prvkov kódu.
- **Zvýšenie interaktivity:**
Pohlcajúce prostredie AR povzbudí vývojárov k aktívnemu skúmaniu a interakcii s kódom, čo povedie k lepšiemu riešeniu problémov a rýchlejšiemu učeniu.
- **Skrátenie času potrebného na onboardovanie nových vývojárov:**
Intuitívna vizualizácia a interakčné mechanizmy pomôžu novým vývojárom rýchlejšie sa oboznámiť s neznámymi alebo rozsiahlymi projektmi.
- **Zlepšenie možností ladenia kódu:**
Vizualizácia chýb, vzťahov a závislostí v AR urobí úlohy ladenia kódu intuitívnejšími a efektívnejšími.
- **Integrácia s existujúcimi nástrojmi:**
Tým, že aplikácia funguje spolu s VS Code, využíva známe pracovné postupy, minimalizuje narušenie a zároveň rozširuje funkčnosť.

VÝSKUMNÉ OTÁZKY 1

Použitelnosť a užívateľská skúsenosť

1. Aká je použiteľnosť aplikácie hodnotená pomocou dotazníka System Usability Scale (SUS)?
2. Ako užívateľská skúsenosť aplikácie, hodnotená dotazníkom User Experience Questionnaire (UEQ), porovnáva s tradičnými programovacími nástrojmi?
3. Existujú konkrétne problémy s použiteľnosťou aplikácie pri riešení programovacích úloh, identifikované prostredníctvom spätnej väzby od používateľov?

Efektivita a účinnosť

1. Zlepšuje aplikácia presnosť a rýchlosť pri programovacích úlohách (napr. navigácia v kóde, ladenie) v porovnaní s tradičnými pracovnými postupmi?
2. Ako efektívne pomáha aplikácia používateľom pochopiť vzťahy medzi triedami a vizualizovať interakcie v rámci kódu?
3. Znižuje aplikácia čas potrebný na nájdenie konkrétnych častí kódu alebo identifikáciu chýb v porovnaní s tradičnými metódami?

VÝSKUMNÉ OTÁZKY 2

Dopad na programovacie pracovné postupy

1. Zlepšujú funkcie rozšírenej reality a Project Northstar programovacie pracovné postupy a efektivitu úloh?
2. Ako ovplyvňuje integrácia OpenAI Coding Assistant schopnosť účastníkov generovať a chápať kód?
3. Ako dobre aplikácia podporuje učenie sa alebo zlepšenie programovacích zručností u pokročilých študentov a profesionálnych programátorov?

Porovnávacia analýza

1. Ako sa výkon používateľov v experimentálnej skupine (používajúcej aplikáciu) porovnáva so skupinou používajúcou tradičný prístup, čo sa týka úspešnosti úloh, času a miery chýb?

Adopcia a škálovateľnosť

1. Aké sú vnímané bariéry pre adopciu aplikácie pri reálnych programovacích úlohách?
2. Je aplikácia vhodná pre veľké projekty a aké sú jej limity v takýchto scenároch?

VYHODNOTENIE PRÍSTUPU

- System Usability Scale (SUS)
- User Experience Questionnaire (UEQ)
- Vstupný a výstupný dotazník
- 2 skupiny programátorov (najlepšie profesionáli z praxe/študenti neskorších ročníkov)
- Zadanie bude pozostávať z podúloh:
 1. identifikácia funkčnosti kódu,
 2. mapovanie vzťahov medzi triedami,
 3. identifikácia a oprava viacerých chýb v kóde,
 4. generovanie metód pomocou OpenAI asistenta
- 1 skupina na počítači, 1 skupina našim prístupom
- Meranie objektívnych meradiel - čas dokončenia, chybovosť riešenia, miera dokončenia úloh a pod.