

SVR 2025/26 – vypracované okruhy na skúšku stručné

Vypracované okruhy k písomnej skúške zo Systémov virtuálnej reality (SVR). Spracované stručne.

1) Virtuálna realita, pojmy, limity, avatar

Virtuálna realita (VR) je počítačom vytvorené prostredie, v ktorom má používateľ pocit, že sa nachádza v syntetickom svete a môže s ním interagovať.

Základné atribúty VR:

- **imerzia** – pocit pohltenia,
- **percepcia** – vnímanie prítomnosti v priestore,
- **interakcia** – možnosť ovládať alebo meniť virtuálne prostredie.

Medzi príbuzné pojmy patria **AR, MR a XR**. AR vkladá virtuálne prvky do reality, MR prepája realitu a virtualitu presnejšie a XR je spoločné označenie pre VR, AR a MR.

Dôležité ľudské limity sú **kybernetická nevoľnosť**, únava očí, fyzická únava a senzorické preťaženie. Problém vzniká najmä vtedy, keď zrak vníma pohyb, ale telo ho necíti.

Avatar je zástupca používateľa vo virtuálnom svete. Môže mať vizuálnu podobu, schopnosti, obmedzenia a implementáciu v systéme.

2) Kategorizácia VR systémov

VR systémy sa delia podľa úrovne technických prostriedkov:

- **Entry VR/XR** – najjednoduchšie systémy,
- **Basic VR/XR** – základná úroveň,
- **Medium VR/XR** – lepšie sledovanie, obraz a interakcia,
- **Immersive VR/XR** – najvyššia úroveň imerzie.

Druhé delenie je podľa dynamiky prostredia a pozorovateľa:

- **SESO** – statické prostredie, statický pozorovateľ,
- **DESO** – dynamické prostredie, statický pozorovateľ,
- **SEDO** – statické prostredie, pohyblivý pozorovateľ,
- **DEDO** – dynamické prostredie aj pohyblivý pozorovateľ.

Najkomplexnejší typ je **DEDO**, lebo sa hýbe prostredie aj používateľ.

3) VR systém, podsystémy a interakcia

VR systém sa skladá z viacerých podsystémov:

- **vizualizačný** – obraz, rendering, osvetlenie,
- **akustický** – zvuk a priestorové audio,
- **hmatový** – vibrácie, dotyk, spätná väzba,
- **kinematický/statokineticý** – sledovanie polohy a pohybu,
- doplnkové: čuchový, biosenzorický.

Interakcia vo VR funguje ako slučka:

vstup používateľa → spracovanie v jadre systému → výstup/spätná väzba

Základné interakčné techniky:

- **selekcia** – výber objektu,
 - **manipulácia** – presun, rotácia, zmena objektu,
 - **lokomócia** – pohyb používateľa vo virtuálnom priestore.
-

4) X-realita, Hyper-realita, adaptabilita rozhraní

XR / X-realita je spoločná platforma pre **VR, AR a MR**. Cieľom je zjednotiť tieto technológie na aplikačnej úrovni.

VR je plne virtuálne prostredie.

AR vkladá virtuálne prvky do reality.

MR prepája reálne a virtuálne prvky tak, aby spolu priestorovo súviseli.

Hyper-realita (HR) je pokročilá kombinácia VR, AR a MR, prípadne aj fyzického prostredia. Ide teda o ešte komplexnejšie prepojenie viacerých realít.

Adaptabilita rozhraní znamená, že XR systém sa prispôsobuje zariadeniu alebo používateľovi. Inak sa ovláda mobil, inak HMD headset a inak desktop. Príkladom adaptácie je ovládanie pohľadom alebo prispôsobenie výstupu podľa typu zariadenia.

5) Technológia digitálneho dvojčaťa

Digitálne dvojča je digitálny model reálneho objektu, výrobku alebo procesu. Zbiera dáta z reality, spracúva ich a umožňuje simuláciu, vizualizáciu a optimalizáciu.

Základná myšlienka:

reálny objekt → dáta zo senzorov → virtuálny model → analýza/simulácia → spätné riadenie reality

Dôležité pojmy:

- **virtualizácia** – prenos údajov z reality do virtuálneho modelu,

- **teleriadenie** – spätné ovplyvnenie reality na základe virtuálneho modelu.

Využitie: priemysel, výroba, údržba, zdravotníctvo, stavebníctvo, automobilový priemysel.

6) Zmiešaná realita, funkcia systému ZR, klasifikácie ZR

Zmiešaná realita (ZR/MR) spája reálny svet s virtuálnymi objektmi v reálnom čase.

Základné znaky MR:

- kombinuje **reálne a virtuálne prvky**,
- je **interaktívna**,
- funguje **v reálnom čase**,
- virtuálne objekty sú **registrované v 3D priestore**.

MR patrí medzi realitu a virtualitu.

AR vkladá virtuálne objekty do reality.

AV vkladá reálne objekty do virtuálneho sveta.

Klasifikácia MR:

- podľa pohľadu: **optical see-through** a **video see-through**,
 - podľa značiek: **marked** a **markerless**,
 - podľa prostredia: **interiérové, exteriérové, kombinované**.
-

7) Typy ZR a možnosti využitia

Typy ZR/MR:

- **s priamym pohľadom** – používateľ vidí realitu cez priehľadný displej,
- **so sprostredkovaným pohľadom** – kamera sníma realitu a systém ju zobrazí na displeji,
- **so značkami** – systém rozpoznáva marker/QR značku,
- **bez značiek** – systém používa obraz, GPS, WiFi alebo SLAM.

Využitie MR:

- priemysel a architektúra,
- medicína,
- výučba a tréning,
- armáda,
- marketing,
- hry a zábava.

Najdôležitejšia myšlienka: **MR pomáha zobrazit' digitálne informácie priamo v reálnom prostredí.**

8) Proces práce aplikácie ZR, značkováná AR

Proces MR/ZR aplikácie:

1. kamera sníma reálnu scénu,
2. systém hľadá značku alebo objekt,
3. vypočíta polohu a orientáciu,
4. vloží virtuálny objekt na správne miesto,
5. používateľ vidí realitu doplnenú o 3D objekt.

Značkováná AR používa marker, napríklad QR kód alebo špeciálny obrazec.

Postup pri značkovanej AR:

- vytvorí sa 3D model,
- vytvorí sa marker,
- model sa priradí k markeru,
- kamera marker rozpozná,
- systém zobrazí 3D objekt na mieste markera.

9) Metódy detekcie v ZR systémoch

Detekcia znamená, že systém musí rozpoznať, **kde sa má virtuálny objekt zobrazit'.**

Základné metódy:

- **SIFT** – hľadá významné body v obraze, odolné voči zmene mierky a otočeniu,
- **SURF** – rýchlejšia obdoba SIFT, používa deskriptory obrazu,
- **SLAM** – naraz určuje polohu zariadenia a vytvára mapu prostredia.

Najdôležitejšia je asi **SLAM**, lebo sa používa v moderných markerless AR/MR systémoch.

10) Limity systémov ZR

Hlavné limity ZR/MR:

- **výpočtová náročnosť** – systém musí spracovať kameru, senzory a 3D grafiku v reálnom čase,
- **latencia** – oneskorenie medzi pohybom a zobrazením,
- **nepresná registrácia** – objekt môže „plávať“ alebo nesediť v priestore,

- **vplyv prostredia** – zlé svetlo, lesklé povrchy, prázdne steny,
- **biologické limity** – únava očí, nevoľnosť, preťaženie používateľa.

Najdôležitejšie na zapamätanie: **ZR musí byť presná, rýchla a stabilná. Ak má veľkú latenciu alebo zlú registráciu, virtuálne objekty nepôsobia reálne.**

11) Kolaboratívna XR/VR

Kolaboratívna XR/VR je systém, kde viacerí používatelia spolu zdieľajú jedno virtuálne alebo XR prostredie v reálnom čase.

Dôležité pojmy:

- **CVR/CXR** – kolaboratívny VR/XR systém,
- **CVE** – kolaboratívne virtuálne prostredie,
- cieľom je **spoločná komunikácia a interakcia viacerých používateľov**.

Vlastnosti:

- viac používateľov naraz,
- zdieľané prostredie,
- spoločný cieľ,
- komunikácia a interakcia v reálnom čase.

Využitie: výučba, tréning, hry, priemysel, medicína, simulácie.

12) Lokálne a globálne zdieľané systémy

Lokálne zdieľané systémy fungujú v jednom fyzickom priestore, napríklad v laboratóriu alebo CAVE. Výhodou je prirodzená komunikácia medzi používateľmi. Nevýhodou je, že používatelia musia byť na rovnakom mieste.

Globálne zdieľané systémy umožňujú spoluprácu používateľov cez internet. Používatelia môžu byť na rôznych miestach, ale potrebná je sieťová infraštruktúra a synchronizácia dát.

Základný rozdiel:

- **lokálne** = rovnaké fyzické miesto,
 - **globálne** = vzdialená spolupráca cez sieť.
-

13) Aplikačná architektúra kolaboratívnej XR/VR

Architektúra určuje, ako spolu komunikujú používatelia a kto rozhoduje o stave prostredia.

Základné typy:

- **klient-server** – server riadi dáta a synchronizáciu,
- **peer-to-peer** – klienti komunikujú medzi sebou bez centrálného servera,
- **distribučovaná serverová architektúra** – viac serverov si delí úlohy.

Najjednoduchšie na zapamätanie:

- klient-server = najväčšia kontrola a lepšia konzistencia,
- peer-to-peer = bez servera, ale viac problémov s konzistenciou,
- distribučovaná architektúra = vhodná pre väčšie a zložitejšie systémy.

Dôležitý pojem je **autorita** – určuje, kto rozhoduje o platnom stave virtuálneho sveta.

14) Zdieľané virtuálne prostredia a konzistencia

Konzistencia znamená, že všetci používatelia vidia rovnaký stav virtuálneho prostredia.

Ak jeden používateľ pohne objektom, táto zmena sa musí dostať aj k ostatným. Inak by každý videl inú scénu a systém by bol nekonzistentný.

Modely konzistencie:

- **centralizované virtuálne prostredie** – stav riadi server,
- **aktívne replikované prostredie** – každý používateľ má kópiu a zmeny sa medzi nimi posielajú,
- **distribučované virtuálne prostredie** – stav je rozdelený medzi viac uzlov/serverov.

Najdôležitejšie: **zdieľané prostredie musí zabezpečiť, aby sa zmeny objektov prenášali všetkým používateľom.**

15) Vlastníctvo údajov a replikácia

Vlastníctvo objektu znamená, že systém určí, kto má právo s daným virtuálnym objektom manipulovať. Je to dôležité, aby dvaja používatelia nemenili ten istý objekt naraz a nevznikol konflikt.

Základné udalosti vlastníctva:

- **získanie vlastníctva** – používateľ začne objekt ovládať,
- **odovzdanie vlastníctva** – používateľ dá objekt inému,
- **odstúpenie vlastníctva** – používateľ sa objektu vzdá.

Replikácia znamená šírenie zmien medzi používateľmi.

Typy replikácie:

- **synchronná** – čaká sa na potvrdenie systému, napríklad pri pripojení používateľa,
- **asynchronná** – zmeny sa posielajú priebežne, napríklad pohyb avatara alebo správy.

Najdôležitejšie: **vlastníctvo rieši, kto môže objekt meniť, a replikácia rieši, aby tú zmenu videli všetci.**

16) Latencia v zdieľanej XR/VR

Latencia je oneskorenie medzi akciou používateľa a reakciou systému.

V zdieľanej XR/VR je problém ešte väčší, lebo sa musí riešiť aj sieť a synchronizácia medzi používateľmi.

Zdroje latencie:

- snímanie vstupu,
- výpočty v jadre systému,
- sieťová komunikácia,
- renderovanie obrazu.

Dôsledky vysokej latencie:

- horšia interakcia,
- zlá synchronizácia používateľov,
- narušenie konzistencie scény,
- zníženie imerzie,
- riziko kybernetickej nevoľnosti.

17) Interakcia používateľov a viacpoužívateľský prístup

V zdieľanej XR/VR interaguje viac používateľov s jedným virtuálnym prostredím v reálnom čase.

Základné interakcie:

- **selekcia** – výber objektu,
- **manipulácia** – práca s objektom,
- **lokomócia** – pohyb v scéne.

Problém vzniká, keď viacerí používatelia chcú naraz meniť rovnaký objekt.

Riešenia:

- **blokovanie** – objekt môže ovládať iba jeden používateľ,
- **voľná interakcia** – používatelia si môžu konkurovať, systém rozhodne, kto má kontrolu.

Bezpečnosť prístupu:

- identifikácia používateľa,
 - autorizácia práv,
 - roly používateľov, napríklad editor, inštruktor, pozorovateľ,
 - ochrana dát a súkromia.
-

18) Základné implementačné stupne XR systému

Implementácia XR systému sa dá rozdeliť na tri hlavné časti:

- **technické prostriedky** – hardvér, senzory, headsety, výpočtový systém,
- **programové prostriedky** – softvér, jadro systému, transformácie, kolízie, skripty,
- **ostatné prostriedky** – fyzikálna, medicínska, psychologická a bezpečnostná podpora.

Jednoducho: XR systém nie je len aplikácia, ale kombinácia **hardvéru, softvéru a podpory používateľa**.

Dôležité je riešiť aj:

- latenciu,
 - bezpečnosť,
 - komfort používateľa,
 - limity človeka.
-

19) Meranie komfortu používateľa XR systémov

Komfort používateľa znamená, ako dobre a bezpečne sa človek cíti pri používaní XR systému.

Meria sa dvoma spôsobmi:

- **subjektívne** – dotazníky, hodnotenie používateľa, rozhovor,
- **objektívne** – meranie výkonu, času, chýb, fyziologických reakcií.

Príklady dotazníkov:

- **SUS** – použiteľnosť systému,
- **NASA-TLX** – mentálna záťaž,
- **VRSQ** – nevoľnosť vo VR,
- **IPQ** – pocit prítomnosti vo virtuálnom prostredí.

Najdôležitejšie: vyššia technická úroveň systému automaticky neznamená vyšší komfort.

20) Koncepcia distribuovaného systému XR

Distribuovaný XR systém rozdeľuje výpočty a dáta medzi viac častí systému, napríklad viac serverov alebo modulov.

Základná myšlienka:

- **primárny server** rieši používateľov a spoločný stav prostredia,
- **sekundárny server** rieši zložitejšie podsystémy alebo výpočty,
- klienti dostávajú aktuálny stav scény.

Dôležité moduly:

- modul používateľa,
- úložisko virtuálnej scény,
- synchronizačný modul,
- fyzikálny/pohybový modul,
- komunikačný modul.

Výhoda distribuovaného systému: lepšia škálovateľnosť, rozdelenie záťaže a stabilnejšia práca pri väčšom počte používateľov.

21) Základný návrh XR systému na interakčnej báze

XR systém je navrhnutý okolo **interakcie používateľa so systémom**.

Základná slučka:

používateľ → vstupné rozhranie → jadro systému → výstupné rozhranie → spätná väzba

Jadro systému spracúva:

- logiku,
- transformácie,
- kolízie,
- stav virtuálneho sveta.

Rozhranie spája používateľa a digitálny systém. Môže byť mechanické, znakové, grafické, prirodzené alebo bioadaptované.

22) Proces interakcie v XR systémoch

Interakcia v XR je proces výmeny informácií medzi používateľom, zariadením a systémom.

Základná úroveň:

- používateľ vykoná akciu,

- vstupný podsystém ju zachytí,
- jadro systému ju spracuje,
- výstupný podsystém vytvorí reakciu.

Dôležité je, že rozhranie musí preložiť ľudský vstup do podoby, ktorej rozumie stroj. Napríklad gesto ruky sa zmení na príkaz v systéme.

Najdôležitejšie: **rozhranie funguje ako transformátor medzi človekom a strojom.**

23) Konštrukcia voľne viazaných, tesne viazaných a kombinovaných XR systémov

XR systémy sa podľa konštrukcie delia na:

- **tesne viazané systémy,**
- **voľne viazané systémy,**
- **kombinované systémy.**

Tesne viazané systémy sú priamo na tele používateľa, typicky **HMD headset**. Môžu byť káblové alebo bezkáblové.

Voľne viazané systémy nie sú pripevnené na používateľovi. Používateľ je v priestore, ktorý ho obklopuje, napríklad **virtuálna jaskyňa CAVE**.

Kombinované systémy spájajú oba prístupy. Napríklad používateľ má MR okuliare a zároveň je vo virtuálnej jaskyni. To sa označuje ako **hyper-realita**.

24) Rozhrania XR systémov

Rozhranie je časť systému, ktorá sprostredkuje komunikáciu medzi používateľom a XR systémom.

Typy rozhraní:

- **mechanické** – fyzické ovládače, tlačidlá,
- **znakové** – textové príkazy,
- **grafické** – GUI, okná, ikony,
- **prirodzené/naturálne** – gestá, hlas, pohyb,
- **bioadaptované** – EEG, EMG, EOG, tep, svalová aktivita.

Rozhranie ako transformácia znamená, že používateľský vstup sa prevedie na strojový príkaz. Napríklad gesto ruky sa zmení na príkaz „chyť objekt“.

Najdôležitejšie: **rozhranie prepája človeka a stroj.**

25) Adaptívne rozhrania XR systémov

Adaptívne rozhranie sa vie prispôbiť zariadeniu, používateľovi alebo situácii.

Príklad: inak sa ovláda XR aplikácia na mobile, inak v headsete a inak na počítači.

Proces adaptácie:

1. používateľ vykoná vstup,
2. systém rozpozná zariadenie a kontext,
3. vstup sa priradí k správnej funkcii,
4. systém vytvorí vhodný výstup alebo spätnú väzbu.

Adaptívne rozhrania sú dôležité hlavne v XR, kde môžu rôzni používatelia používať rôzny hardvér.

26) Adaptívne rozhrania XR podľa účelu nasadenia

Adaptívne rozhranie sa prispôsobuje používateľovi, zariadeniu alebo situácii.

Podľa účelu sú dva hlavné typy:

- **štandardné adaptívne rozhrania** – prepájajú jadro XR systému s podsystémami, napríklad fyzikou, senzormi alebo simuláciou,
- **viacpoužívateľské adaptívne rozhrania** – umožňujú spoluprácu používateľov s rôznym hardvérom, napríklad jeden je vo VR a druhý cez mobilnú AR.

Najdôležitejšie: **adaptívne rozhranie zabezpečí, aby rôzne vstupy fungovali v jednom XR systéme jednotne.**

27) XR a bio-senzorické rozhrania

Bio-senzorické rozhrania snímajú biologické signály používateľa a používajú ich ako vstup do XR systému.

Príklady signálov:

- **EEG** – mozgová aktivita,
- **EMG** – svalové napätie,
- **EOG / eye-tracking** – pohyb očí,
- tep, teplota, stres alebo iné fyziologické údaje.

Cieľom je vytvoriť užšie prepojenie medzi človekom a systémom. Systém sa potom môže prispôbiť stavu používateľa alebo ho môže používateľ ovládať biologickými signálmi.

28) Programová implementácia XR systému

Programová implementácia XR systému má tri hlavné úrovne:

- **logické jadro** – pravidlá, správanie, vzťahy a logika systému,
- **modelovanie XR** – popis scény, objektov, transformácií a skriptov,
- **vizualizácia/simulácia** – používateľ vidí scénu a interaguje s ňou.

XR systém musí pracovať v **reálnom čase**. Dôležitá je stabilná frekvencia, približne **60–120 Hz**. Ak systém nestíha, vzniká sekanie, zlá imerzia a kybernetická nevoľnosť.

Proces implementácie:

požiadavky → formalizácia → tvorba objektov → simulácia → vizualizácia a spätná väzba

29) Postup spracovania požiadaviek na XR systém

Pri návrhu XR systému treba najprv určiť, čo má systém robiť a pre koho je určený.

Základný postup:

1. určiť **oblasť záujmu** systému,
2. určiť **parametre** systému,
3. určiť **typy parametrov**,
4. definovať **vzťahy medzi parametrami**,
5. určiť typy vzťahov, napríklad priame, nepriame alebo podmienené.

Najdôležitejšie: **požiadavky sa musia previesť na parametre a vzťahy, aby sa dali implementovať.**

30) Virtuálny priestor a jeho parametre

Virtuálny priestor je základné prostredie, v ktorom sa odohráva virtuálny dej.

Charakterizuje ho:

- **dimenzia priestoru**,
- **súradnicová sústava**,
- **objekty a ich formálny opis**,
- **dynamika** – čas, pohyb, transformácie a zmeny.

Jednoducho: virtuálny priestor určuje, **kde sú objekty, ako sú popísané a ako sa môžu meniť v čase.**

31) Reprezentácia a popis virtuálneho sveta a objektov

Virtuálny svet je v XR systéme reprezentovaný ako súbor objektov, máp, terénov, senzorov, kamier, kolíznych objektov a ďalších entít.

Každý virtuálny objekt má:

- **globálne vlastnosti** – poloha, rozmery, tvar, materiál,
- **lokálne vlastnosti** – farba, zvuk, pohyb, deformácia, haptická spätná väzba,
- **správanie** – ako reaguje na udalosti a interakciu.

Vzťahy medzi objektmi určujú, ako spolu komunikujú alebo sa ovplyvňujú.

Na opis virtuálneho sveta sa používajú:

- **popisné formáty** – napríklad VRML, XML, glTF, USD,
- **skriptovanie** – napríklad C#, Python, C++, Lua.

Najdôležitejšie: **virtuálny svet je formálne popísaná scéna z objektov, ich vlastností a vzťahov.**

32) Súradnicové sústavy a transformácie

Súradnicová sústava určuje polohu a orientáciu objektov vo virtuálnom priestore.

Základné typy:

- **translačná** – popisuje posun,
- **rotačná** – popisuje otočenie,
- **kombinovaná** – spája posun aj rotáciu.

Dôležitý pojem je **DOF – stupeň voľnosti**, teda počet nezávislých možností pohybu objektu.

Transformačný reťazec prevádza objekt z lokálneho priestoru až na obrazovku:

objekt → svetová sústava → kamera → projekcia → zobrazovač

Transformácie môžu byť:

- **lineárne** – posun, rotácia, škálovanie,
- **nelineárne** – deformácie alebo perspektívne skreslenie.

Najdôležitejšie: **transformácie určujú, kde objekt je, ako je otočený a ako sa zobrazí používateľovi.**

33) Quaternióny, Eulerove uhly, základné pojmy

Rotáciu v 3D priestore možno popísať:

- **Eulerovými uhlami**,
- **quaterniónmi**.

Eulerove uhly používajú rotácie okolo osí X, Y, Z. Sú jednoduché, ale môžu spôsobiť problém **gimbal lock**.

Eulerov teorém hovorí, že každú rotáciu v 3D priestore možno vyjadriť ako otočenie o určitý uhol okolo určitej osi.

Quaternión je matematická štruktúra so štyrmi zložkami:

$$\mathbf{q} = w + xi + yj + zk$$

Kde:

- w je reálna zložka,
- x, y, z sú imaginárne zložky.

Výhody quaterniónov:

- sú vhodné na rotácie v 3D,
- sú plynulé,
- šetria pamäť,
- zabráňujú problému gimbal lock.

34) Quaternióny a rotácia, gimbal lock

Quaternióny sa používajú na stabilné a plynulé otáčanie objektov v 3D priestore.

Rotácia bodu pomocou quaterniónu sa zapisuje:

$$\mathbf{B}' = \mathbf{q} \cdot \mathbf{B} \cdot \mathbf{q}^{-1}$$

Kde:

- $\mathbf{B}$ je pôvodný bod,
- $\mathbf{q}$ je rotačný quaternión,
- $\mathbf{q}^{-1}$ je inverzný quaternión,
- $\mathbf{B}'$ je otočený bod.

Gimbal lock je problém Eulerových uhlov. Nastane vtedy, keď sa dve rotačné osi prekryjú alebo stanú rovnobežnými. Systém potom stratí jeden stupeň voľnosti a objekt sa nedá správne otáčať.

Najdôležitejšie: **quaternióny riešia problém gimbal lock a preto sa používajú v XR systémoch na rotácie.**

35) Virtualizačný reťazec a jeho realizácia

Virtualizačný reťazec opisuje postup, ako sa z reálneho sveta alebo návrhu vytvorí virtuálny svet a prípadne späť reálny objekt.

Základné kroky:

1. **získanie vstupných údajov** – meranie, skenovanie, požiadavky,
2. **modelovanie a editácia** – tvorba 3D modelov a scény,
3. **overovanie** – kontrola správnosti modelu,
4. **vizualizácia a práca s virtuálnym svetom** – používateľ vidí scénu a interaguje,
5. **3D tlač / vytvorenie reálneho objektu** – návrat z virtuálneho sveta do reality.

Dôležitý je **vizualizačný podsystem**, ktorý zabezpečuje obraz, transformácie, viditeľnosť, osvetlenie a rendering.

Dôležité je aj **sledovanie polohy**, teda tracking používateľa, headsetu alebo ovládačov.

Najdôležitejšie: **virtualizačný reťazec prepája realitu, virtuálny model, vizualizáciu a prípadne spätné vytvorenie fyzického objektu.**

36) Vizualizácia, rendering, grafické architektúry

Vizualizácia je proces, pri ktorom sa model virtuálneho sveta prevedie na obraz pre používateľa.

Rendering / vykresľovanie je samotná tvorba obrazu pomocou grafických algoritmov.

Základné kroky vizualizácie:

- spracovanie modelu,
- geometrické transformácie,
- riešenie viditeľnosti,
- osvetlenie a tieňovanie,
- rasterizácia,
- finálne zobrazenie obrazu.

V XR je problém, že všetko musí bežať **v reálnom čase**, s vysokým FPS a nízkou latenciou.

Najdôležitejšie limity:

- výkon GPU,
 - spotreba energie,
 - prehrievanie headsetu,
 - latencia,
 - zložitosť scény.
-

37) Špecifické požiadavky a pokročilé rendering techniky XR

XR vizualizácia musí byť plynulá, lebo oneskorenie spôsobuje nevoľnosť a stratu imerzie.

Dôležité požiadavky:

- **stereoskopické zobrazenie** – iný obraz pre ľavé a pravé oko,
- **vysoká obnovovacia frekvencia** – približne 90–120 Hz,
- **nízka latencia** – obraz musí rýchlo reagovať na pohyb hlavy,
- **stabilné FPS** – bez trhania.

Pokročilé techniky:

- **foveated rendering** – vysoká kvalita len tam, kam sa používateľ pozerá,
- **time-warp** – dopočítanie obrazu podľa pohybu hlavy,
- **space-warp** – dopočítanie medzisnímok pri slabšom výkone,
- **asynchrónne vykresľovanie** – oddelenie renderovania od zobrazovania.

Najdôležitejšie: **cieľom je plynulý obraz s nízkou latenciou.**

38) Paralelné vykresľovanie

Paralelné vykresľovanie znamená, že výpočet obrazu nerobí jedna jednotka, ale viac procesorov alebo jadier naraz.

Používa sa preto, že XR scény sú náročné:

- veľa objektov,
- veľa polygónov,
- vysoké rozlíšenie,
- potreba real-time odozvy.

Základné typy paralelizmu:

- **funkcionálny paralelizmus** – každá časť pipeline robí inú úlohu,
- **údajový paralelizmus** – viac jednotiek spracúva rôzne dáta naraz,
- **časový paralelizmus** – rôzne procesory počítajú rôzne snímky.

Najdôležitejšie: **paralelizmus zrýchľuje rendering a pomáha udržať plynulosť XR systému.**

39) Paralelizmus v grafických architektúrach

Grafické architektúry využívajú paralelizmus, aby zvládli rýchlo spracovať veľké množstvo dát.

Tri hlavné formy:

- **funkcionálny paralelizmus** – rozdelenie výpočtu na kroky pipeline,
- **údajový paralelizmus** – súbežné spracovanie objektov alebo pixelov,
- **časový paralelizmus** – rozdelenie práce podľa snímok.

Údajový paralelizmus sa delí na:

- **objektový** – práca s objektmi a transformáciami,
- **obrazový** – práca s pixelmi, osvetlením a viditeľnosťou.

V praxi sa najčastejšie používa **hybridný prístup**, teda kombinácia viacerých typov paralelizmu.

40) Klasifikácia grafických architektúr

Grafické architektúry sa delia podľa toho, **kde v renderovacej pipeline sa triedia dáta**.

Základné typy:

- **Sort-First** – triedenie na začiatku, podľa oblasti obrazu,
- **Sort-Middle** – triedenie v strede, po transformácii objektov,
- **Sort-Last** – triedenie na konci, keď každý procesor vyrenderuje časť obrazu,
- **kompozícia obrazu** – spájanie výsledných pixelov,
- **viacnásobné triedenie** – komunikácia na viacerých úrovniach pipeline.

Najjednoduchšie porovnanie:

- **Sort-First** je dobrý pri rozdelení obrazu na oblasti,
 - **Sort-Middle** delí prácu medzi transformáciu a rasterizáciu,
 - **Sort-Last** dobre zvláda veľa objektov, ale je náročný na prenos pixelov.
-

41) Technológia GPGPU

GPGPU znamená využitie grafickej karty na všeobecné výpočty, nielen na grafiku.

GPU je vhodné na výpočty, lebo má veľa jadier a vie spracovať veľa dát paralelne.

Použitie GPGPU:

- fyzikálne simulácie,
- umelá inteligencia,
- spracovanie obrazu,
- rendering,
- výpočty vo VR/XR systémoch.

Základné prostriedky:

- **CUDA** – platforma od NVIDIA,
- **OpenCL** – otvorený štandard,
- **DirectX Compute** – riešenie od Microsoftu.

Najdôležitejšie: **GPGPU zrýchľuje náročné výpočty pomocou paralelného výkonu grafickej karty.**

42) Priestorové zobrazovanie, oko, VAC, Panumova oblasť

Priestorové videnie vzniká tak, že každé oko vidí mierne odlišný obraz a mozog z toho vytvorí hĺbkový vnem.

Dôležité pojmy:

- **akomodácia** – zaostrenie oka,
- **konvergencia** – natočenie očí na objekt,
- **zorné pole** – priestor, ktorý oko vidí,
- **binokulárna disparita** – rozdiel obrazu medzi ľavým a pravým okom.

VAC konflikt vzniká v XR headsetoch preto, že oči konvergujú na virtuálny objekt, ale akomodujú na pevnú vzdialenosť displeja. To spôsobuje únavu očí, bolesť hlavy alebo nevoľnosť.

Panumova fúzna oblasť je rozsah, v ktorom mozog ešte dokáže spojiť obrazy z oboch očí do jedného 3D vnemu.

Najdôležitejšie: **XR musí rešpektovať limity oka, inak vzniká únava a nevoľnosť.**

43) Technológie stereoskopického a priestorového zobrazovania

Priestorové zobrazovanie sa snaží vytvoriť dojem hĺbky.

Základné technológie:

- **stereoskopia** – iný obraz pre ľavé a pravé oko,
- **autostereoskopia** – 3D obraz bez okuliarov,
- **volumetrické zobrazovanie** – obraz ako body v priestore,
- **holografia** – tvorba priestorového svetelného poľa,
- **light-field displeje** – zobrazujú smer svetelných lúčov,
- **varifokálne/multifokálne displeje** – riešia problém zaostrovaní.

Stereoskopické systémy môžu používať:

- anaglyfické okuliare,
- polarizačné okuliare,

- aktívne shutter okuliare,
- XR headsety.

Najdôležitejšie: **cieľom je vytvoriť hĺbkový vnem čo najprirodzenejšie a znížiť únavu očí.**

44) Stereogramy

Stereogram je obraz, ktorý využíva rozdiel medzi obrazmi pre ľavé a pravé oko. Mozog z týchto rozdielov vytvorí ilúziu 3D povrchu.

Typy:

- **stereopár** – dva obrazy pre ľavé a pravé oko,
- **random dot stereogram** – 3D vnem z náhodných bodiek,
- **autostereogram** – jeden obrázok, v ktorom je skrytý 3D objekt.

Na vytvorenie stereogramu sa často používa **hĺbková mapa**, ktorá určuje, ktoré časti obrazu majú byť bližšie alebo ďalej.

Spôsoby pozerania:

- **divergentne** – pozeranie akoby za obrázok,
- **konvergentne** – krížový pohľad pred obrázok.

Najdôležitejšie: **stereogram vytvára 3D ilúziu pomocou rozdielov, ktoré mozog spojí do hĺbkového vnemu.**

45) Stereoskopia – anaglyfické zobrazenie

Anaglyf je najjednoduchšia stereoskopická technika. Používa farebné filtre, najčastejšie **červený a azúrový**.

Princíp:

- jeden obraz je určený pre ľavé oko,
- druhý obraz je určený pre pravé oko,
- obrazy sa farebne odlišia,
- okuliare prepustia každému oku iba jeho obraz,
- mozog vytvorí 3D vnem.

Typy farebných filtrov:

- | | |
|---------------------|-----------------------|
| • Red-Cyan, | • Red-Blue, |
| • Red-Green, | • Yellow-Blue. |

Výhody:

- lacné,
- jednoduché,
- funguje na bežnom monitore aj papieri.

Nevýhody:

- zlé farby,
- únava očí,
- bolesť hlavy,
- nižší komfort.

Najdôležitejšie: **anaglyf je lacná 3D metóda, ale má slabú kvalitu farieb a nízky komfort.**

46) Polarizácia, Infitec, autostereoskopia

Polarizácia je pasívna stereoskopická technika. Obraz pre ľavé a pravé oko sa oddelí pomocou polarizačných filtrov.

Typy polarizácie:

- **lineárna,**
- **kruhová,**
- **eliptická.**

Používa sa napríklad v kinách alebo systémoch typu **CAVE**.

Infitec je tiež pasívna technika. Delí svetlo podľa vlnových dĺžok, takže každé oko dostane iný obraz, ale farby sú kvalitnejšie než pri anaglyfe.

Autostereoskopia umožňuje 3D obraz **bez okuliarov**. Používa napríklad paralaxnú bariéru alebo lentikulárne šošovky.

Najdôležitejšie: **polarizácia a Infitec používajú okuliare, autostereoskopia nie.**

47) Hĺbkový vnem, problémy HMD, virtuálne kamery

Hĺbkový vnem je schopnosť vnímať vzdialenosť objektov. Vzniká hlavne vďaka rozdielu medzi obrazom ľavého a pravého oka.

Faktory hĺbky:

- binokulárna disparita,
- veľkosť objektov,
- prekrytie objektov,
- tieňovanie,
- pohybová paralaxa.

Hlavný problém HMD headsetov je **VAC konflikt**:

- oči sa natáčajú na virtuálny objekt,
- ale zaostrujú na pevnú vzdialenosť displeja.

Dôsledky:

- únava očí,
- bolesť hlavy,
- nevoľnosť.

Virtuálne kamery v XR simulujú ľavé a pravé oko. Musia byť správne nastavené, aby vznikol prirodzený stereoskopický obraz.

Najdôležitejšie: **HMD musí správne zobrazovať dva pohľady, inak vzniká únava a zlý 3D vnem.**

48) Stereoskopické zobrazovacie systémy

Stereoskopické systémy zobrazujú iný obraz pre ľavé a pravé oko. Mozog ich spojí do 3D vnemu.

Základné typy:

- **pasívne systémy** – anaglyf, polarizácia, Infitec,
- **aktívne systémy** – shutter okuliare, ktoré sa striedavo zatmievajú,
- **autostereoskopické systémy** – 3D bez okuliarov.

Podľa konštrukcie:

- **tesne viazané** – headsety, okuliare, HMD,
- **voľne viazané** – CAVE, projekčné systémy.

Pri projekcii môže byť:

- **predná projekcia** – projektor je pred plátnom,
- **zadná projekcia** – projektor je za plátnom.

Najdôležitejšie: **stereoskopické systémy vytvárajú 3D obraz oddelením pohľadu pre každé oko.**

49) Kolízie vo virtuálnych scénach

Kolízia nastane, keď sa dva objekty vo virtuálnej scéne dotknú, prekryjú alebo narazia.

Na detekciu sa používajú **kolajdre** – zjednodušené neviditeľné tvary objektov.

Typy kolajdrov:

- **Sphere Collider** – guľa, rýchly výpočet,
- **Capsule Collider** – vhodný pre postavy,
- **Box Collider** – kváder,
- **Mesh Collider** – presný, ale náročný.

Typy senzorov:

- **vertikálny senzor** – aktivácia ukázaním alebo klikom,
- **horizontálny senzor** – aktivácia vstupom do oblasti,
- **pohľadový senzor** – aktivácia, keď je objekt v zornom poli.

Dôvody riešenia kolízií:

- objekty nemajú prechádzať cez seba,
 - kamera nemá prechádzať cez steny,
 - spúšťanie udalostí,
 - realistická fyzika.
-

50) Detekcia kolízie a algoritmy

Detekcia kolízie zisťuje, či sa objekty dotkli alebo prekryli.

Výsledkom je:

- či kolízia nastala,
- ktoré objekty kolidovali,
- kde alebo kedy nastala kolízia.

Základný postup:

1. vyberú sa objekty na testovanie,
2. vylúčia sa objekty, ktoré určite nekolidujú,
3. presne sa otestujú zvyšné objekty,
4. výsledok sa pošle jadru systému.

Fázy detekcie:

- **Broad phase** – rýchla hrubá fáza, vylúči väčšinu objektov,
- **Narrow phase** – presná fáza, rieši konkrétne kolízie.

Používané metódy:

- | | |
|--|--|
| • AABB/OBB – ohraničujúce boxy, | • BSP/OCTREE – delenie priestoru, |
| • BSH – hierarchia gúľ, | • CCD – kontinuálna detekcia, rieši rýchle objekty. |

Najdôležitejšie: **najprv sa rýchlo vylúčia nemožné kolízie a až potom sa presne testujú podozrivé objekty.**

51) Algoritmy riešenia kolízií vo virtuálnych scénach

Riešenie kolízie nastáva po tom, čo systém zistí, že kolízia vznikla.

Cieľom je určiť, čo sa má stať po náraze.

Možné reakcie:

- **zastavenie pohybu** – objekt narazí a zastaví sa,
- **zmena trajektórie** – objekt sa odrazí alebo zmení smer,
- **deformácia objektu** – objekt sa ohne, poškodí alebo zmení tvar,
- **animačný efekt** – napríklad rozbitie, zvuk alebo vizuálny efekt,
- **aktivácia udalosti** – napríklad otvorenie dverí alebo zobrazenie informácie.

Najdôležitejšie: **detekcia kolízie zistí, že kolízia nastala; riešenie kolízie určí reakciu systému.**

52) Modely organizmov, skinning, emocionálny model

Organizmus vo virtuálnom svete sa neskladá iba z jedného modelu, ale z viacerých vrstiev.

Základné modely:

- **skeletálny model** – kostra,
- **muskulárny model** – svaly,
- **povrchový model** – koža / mesh,
- **fyzikálny model** – reakcie na sily,
- **riadiaci model** – správanie alebo AI,
- **kinematický model** – pohyb,
- **emocionálny model** – výrazy a emócie.

Skinning je prepojenie kostry s povrchovým modelom. Keď sa pohne kosť, deformuje sa aj mesh.

Používajú sa váhy vrcholov, ktoré určujú, ako veľmi daná kosť ovplyvňuje časť modelu.

Emocionálny model sa často rieši cez **blend shapes / morph targets**, teda prelínanie tvarov tváre.

Najdôležitejšie: **kostra riadi pohyb a skinning prenáša tento pohyb na povrch modelu.**

53) Pohyby modelov, typy kĺbov, kinematický reťazec

Pohyb modelu je zmena polohy alebo orientácie objektu v priestore.

Pri organizmoch sa používa **kĺbová štruktúra**.

Základné pojmy:

- **báza** – pevný začiatkový bod reťazca,
- **koncový efektor** – koncový bod, ktorého polohu sledujeme,
- **DOF** – stupeň voľnosti, teda možnosti pohybu kĺbu.

Typy kĺbov:

- **translačný** – umožňuje posun,
- **rotačný** – umožňuje otáčanie.

Kinematický reťazec je postupnosť kĺbov a kostí od bázy po koncový efektor.

Najdôležitejšie: **pohyb postavy sa opisuje pomocou kĺbov, kostí a kinematického reťazca.**

54) Priama kinematika

Priama kinematika / Forward Kinematics znamená, že poznáme uhly kĺbov a z nich vypočítame polohu koncového efektora.

Jednoducho:

uhly kĺbov → poloha ruky/nohy/koncového bodu

Príklad: keď nastavíme rameno, lakeť a zápästie, systém vypočíta, kde sa nachádza ruka.

Výpočet prebieha postupne cez transformačné matice alebo quaternióny v kinematickom reťazci.

Výhoda:

- jednoduchá a stabilná.

Nevýhoda:

- používateľ musí nastavovať jednotlivé kĺby, nie cieľovú polohu.

Najdôležitejšie: **priama kinematika počíta výslednú polohu z nastavených uhlov kĺbov.**

55) Inverzná kinematika

Inverzná kinematika / Inverse Kinematics rieši opačný problém než priama kinematika.

Jednoducho:

cieľová poloha ruky/nohy → potrebné uhly kĺbov

Príklad: chceme, aby sa ruka dotkla objektu. Systém vypočíta, ako sa musia otočiť rameno, lakeť a zápästie.

Problémy IK:

- nemusí existovať riešenie,
- môže existovať viac riešení,
- kĺby majú fyzické limity,
- výpočet môže byť náročný.

Metódy:

- **analytické** – presné a rýchle, ale málo univerzálne,
- **numerické/iteratívne** – univerzálnejšie, ale pomalšie.

Príklad algoritmu:

- **CCD** – postupne upravuje kĺby od koncového efektora smerom k báze, aby sa efektor priblížil k cieľu.

Najdôležitejšie: **inverzná kinematika počíta uhly kĺbov tak, aby sa koncový bod dostal do cieľa.**