

**1. Jak uživatel vnímá interaktivní systém?**

Jako uživatel – tedy vnímá jen to co vidí a to na co může kliknout. Neřeší v čem je to napsané a jaké patterny jsou použity apod. Vidi jen tu slupku designu.

**2. Co uživatel očekává od UI?**

Že bude intuitivní, rychlý, splní svou funkci, bez chyb, jednoznačný, zapamatovatelný

**3. Bude si uživatel stěžovat na využití UI?**

Nebude. Uživatelé si nestěžují ani nechválí. Často si myslí, že je to jejich chyba.

**4. Rozdíl mezi SW inženýrem a HCI specialistou v oblasti vývoje UI?**

Sw inženýr přemýšlí spíš nad tím jaké patterny použít, jakou technologii. Hci chce se více soustředit na tu „lidskou stránku“ chce, aby systém byl jednoduchý, intuitivní apod. po stránce designu.

**5. Základy post-WIMP UI**

Wimp UI je jen 2d, objekty, drag and drop (windowsy).

Ale post-WIMP UI reaguje na zvuk, dotyk, pohyb. Může být 3d

**6. Jaké jsou omezení konceptu WIMP GUI?**

Nereaguje na pohyb, zvuk apod..

**7. Definuje zvukové rozhraní novou třídu UI?**

Ne, spadá pod post-WIMP UI

**8. Proč je MVC důležité pro vývoj UI?**

Protože rozděluje implementaci appku do patternu. Do 3 nezávislých modulů. A to umožňuje udržet velké aplikace přehledné a jednoduché na případné vylepšování či údržbu

**9. Popiš ovládací prvek UI Window. Jaký je rozdíl oproti Page?**

Ve window je appka zobrazena, page je ve window a ve window může být několik pagů

**10. Co je interakční designový vzor (design pattern)? Vysvětlete na příkladech.**

Je designový pattern, který poskytuje základní a opakovatelné řešení při problému při tvorbě designu rozhraní. Příklad: multi-level undo

**11. Co je deklarativní popis UI?**

To je kód v XAMLU – rozmístění a popis komponent

Procedurální – jsou procedury v kodu například: c#

**12. Popiš MVC model. Jaké má omezení?**

Model – má v sobě backend, databázi, fungování apod -> komunikuje s View

View – zobrazuje to co vidí uživatel – kliky posílá na controller

Controller – controller dostane něco že se ve view na něco klikne a zavolá konkrétní věc z modelu

Omezení – model by neměl měnit view, View a controller nejsou odděleny, uživatel pracuje s oběma. View má část logiky v modelu

**13. Jaké jsou rozdíly mezi MVC a MVP?**

Model, view, presenter. Presenter je mezi view a modelem. Řídí komunikaci mezi nimi. Uživatel teď už komunikuje pouze s View, který bere input a zároveň vykresluje output. Presenter volá potřebné věci z modelu.

**14. Popiš observer pattern. Jak lze udělat Model nezávislý na View a Controller?**

Observer – má své observery. Jakmile se v něm něco změní – informuje posluchače. Viz mapa v střílečkách. Když hráč změní svoji pozici notifikuje ostatní že změnil pozici -> updatuje observery

**15. Popiš data binding pattern. Jak ho lze použít pro implementaci MVVM modelu?**

Model, viewmodel, view. To same jako presenter, jen komponenty z view mohou být nabíndované na viewmodel. Když se změní jedna změní se druhá(two-way binding)

**16. Jak lze dosáhnout content customization? Napiš kód pro vytvoření tlačítka s obrázkem.**

```
<Button>  
    <StackPanel>  
        <Image ....\>  
        <Label ....\>  
    </StackPanel>  
</Button>
```

**17. Jak lze přizpůsobit komponenty pomocí Style?**

Stylem lze pomocí setterů měnit vlastnosti určité komponenty(button, circle apod..)

**18. Jak lze přizpůsobovat pomocí triggerů? Jaké trigger existují?**

Triggery jsou jako styly, ale spustí se až za určité podmínky.

Data trigger

Event trigger

Property trigger

**19. Jak lze udělat vlastní seznam položek? (custom list of items)**

CustomControl se dají např. do stack panelu.

(Tak prý využijeme trigger a setery pro interaktivitu a definujeme cíl content template a jednotlivé položky v item template)

**20. Jak lze využít ContentTemplate pro přizpůsobení**

Nevím..mrdat

Vytvoríme custom componentu a vložíme ji do contenttemplate, která swapne strom

**21. Porovnej UserControl a CustomControl.**

CustomControl si uděláme každý vlastní, lze stylizovat.

UserControl už existuje, nelze stylizovat.

**22. Jak se definují validační pravidla s Data source exception?**

Binding.validationRules->ExceptionValidationRule

**23. Jak se definují validační pravidla s Data error Interface?**

IDataErrorInfo -> DataErrorValidationRule

Asynchronní volání

**24. Jak se definují validační pravidla s vlastní validační třídou?**

Vytvoříme novou třídu dědící z validationRule a přepíšeme metodu Validate.

**25. Jak lze zobrazit chybové stavy při použití ControlTemplate a Style?**

Objektu dame error template, který definujeme pomocí:

-Control template

-Style, trigger

**26. Jak implementovat navigaci v Okně? Kdy se používá Page?**

Jako v browser navigation. Odkazuje cestu na jednotlivé pages

**27. Jaký je proud událostí mezi stránkami? (Pages)**

Navigation process-> loading -> init 2nd page -> unload 1st page -> navigated -> LoadCompleted -> 2nd page loaded

**28. Jak posílat a získávat data mezi stránkami?**

Forward passing - navigate parametr, událost LoadCompleted  
Returning data - používá se PageFunction místo Page

**29. Popište rozdíly mezi Webovou mobilní aplikací a Nativní mobilní aplikací.**

Webova funguje na všech platformách – je to jakoby web. Má vyšší dosah, ale menší strávený čas.  
Nativní – více využívá HW, pouze na jedné platformě. Nižší navstevnost než webova, ale vyšší strávený čas.

**30. Jak implementovat hybridní mobilní aplikaci? WebView vs Compiled approach.**

WebView – jako web a pak se zkompiluje do webu  
Compiled – jeden kus kódu a zkompiluje se do verzi pro danou platformu

**31. Jaký je rozsah UI specifikace?**

Screenflow, popisky, obrázky, uživatele, scénáře, obsah, design interakce, indikátor vyzrálosti, dekompozice, konkrétní komponenty (button apod.)

**32. Definuj formativní vyhodnocování. Jak je začleněno do vývojového procesu?**

Mělo by být v každé fázi vývoje. Viz jestli se splnil účel jestli to funguje apod.

**33. Co by nemělo chybět v UI specifikaci, aby byla implementace více efektivní?**

Komponenty a tyhle sračky. Maturity of specification, vyhodnocování, apod

**34. Co znamená indikátor vyzrálosti v UI specifikaci? (maturity indicator)**

Připraveno na přímou implementaci - většina relevantních situací, pořádné uživatelské testování (hodně účastníků a iterací), stabilní design, málo implementačních iterací  
Implementace s konzultacemi - většina relevantních situací, ale jen zběžné uživatelské testování (málo účastníků i iterací), design se může měnit  
Nutné vyhodnocení designu - nereflektuje většinu relevantních situací, zběžné testování, design se výrazně mění, implementace se často zahazuje

**35. Co je accessibility API? Popiš architekturu a mapování MVC / MVVM modelu.**

Rozhraní pro lidi s postižením, aby mohli používat appky.  
Kvůli omezením mohou být určité View a controlery nepoužitelné.

**36. Definuj Accessibility RIA.**

Rich internet application  
Webova aplikace, která požaduje plugíny Adobe apod. (výhoda pro hendikepované)  
ARIA to samé ale jen nějaký vývojový standard  
Hendikepovaných je ve společnosti cca 1%

**37. Popiš kroky k vytvoření ARIA aplikace.**

Nevím, vole mrdat.  
Přístupnost RIA komponent – vytvoření offline prototypu, zjednodušení architektury přidání ARIA API apod.  
Přístupnost RIA aplikace – ARIA heuristika  
Testování aplikační přístupnosti – musí testovat handicapovaní uživatelé

**38. Co by mělo být testováno při testování UI?**

Chybovost, zapamatovatelnost, jednoduchost, intuitivita, čitelnost, přehlednost, použitelnost, odolnost proti chybám

### 39. Jaké jsou metody tvorby testovacích skriptů na UI?

- Programem
- nahrávání – testovací kroky se nahrají automatizačním kódem a pak se provedou na appce
- GUI ripping – zvolí se začátek a konec a vygenerují se všechny možné scénáře mezitím

### 40. Jak lze simulovat lidský faktor při testování UI?

- MONKEY testing – random kliky – není zde lidský faktor
- Lepší opička – zařazení časové limity, aby to působilo víc lidsky

### 41. Co je Usability testing?

Test použitelnosti. V podstatě to co se má testovat v UI. Takže testujeme zda uživatel dělá hodně chyb, jak je s ním uživatel schopen pracovat po seznámení/bez seznámení. Za jak dlouho je schopen znovu používat když ji dlouhou dobu nepoužíval a chce začít znovu. Uživatelská spokojenost – zda ho to uspokojí, když splnil úkol. (dle průzkumů ženám vyhovuje lehké zavibrování)

### 42. Jaké jsou evaluační metody s / bez uživatele?

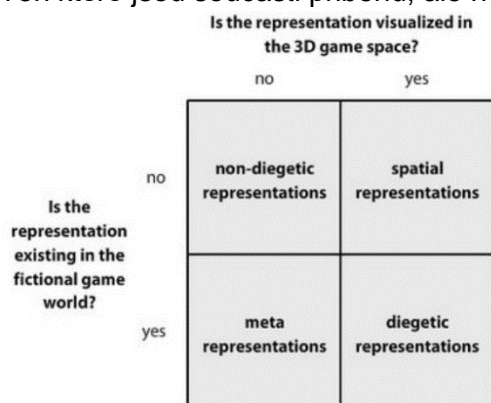
- Bez uživatele – kognitivní průchod, heuristické vyhodnocování (rizeno mnoha experty)
- S uživatelem – v laborce/v jeho prostřetí

### 43. Co je Data mocking? Co je Wizard-of-Oz?

- Wizard of oz – systém zatím neexistuje – nahrazen člověkem co ho simuluje
- Data mocking – data která neexistují, ale potřebujeme je – dame přímo do kódu

### 44. Klasifikace herního UI + popis každého typu

- Ty vole kamo mě picne
- Takže tu máme
- Ten o kterém herní postava neví : viz ping v csu, killy
- Ten o kterém herní postava ví- sama ho používá – mapa v far cry
- Ten o kterém herní postava neví a jsou aby hráčovi pomohli: pr přihrávka ve fotbale
- Ten které jsou součástí příběhu, ale ne herního světa. Pr krvavá obrazovka apod



### 45. Proč stále existuje více druhů herních enginů a ne jen jeden?

Protože je strašně moc typů her a jeden by nevyhovoval všem

### 46. Co je rozhraní HID a jaké jsou druhy?

- Human interface devices
- Něboli zařízení, kterým může uživatel ovládat hru
- Př: ps ovladač, klavesnice

### 47. Jaké jsou druhy věrnosti interakce?

Realistická – existuje v realnem svete

Nerealistická – není jako v realnem svete, je lehce lepší

Magická – totalni chaos, letani teleportace

#### **48. Jaké jsou referenční rámce ve VR?**

Ještě jednou? Nerozumím otázce...a taky ji nevím xD

Jo aha už vim. Je to proto aby systém VR mohl sledovat a menit prostředí podle toho co dělá a jak se hýbe uživatel.

world reference frame, real-world reference frame, referenční rámec těla, ruky, hlavy, očí