



(12) **SØKNAD**

(19) NO

(21) **20101412**

(13) **A1**

NORGE

(51) Int Cl.

G06F 9/445 (2006.01)

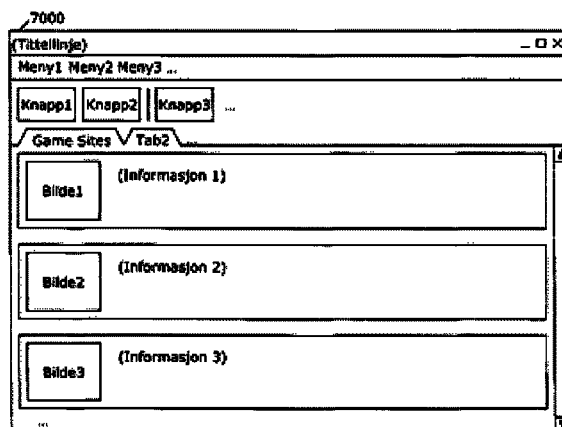
A63F 13/12 (2006.01)

Patentstyret

(21)	Søknadsnr	20101412	(86)	Int.inng.dag og søknadsnr
(22)	Inng.dag	2010.10.12	(85)	Videreføringsdag
(24)	Løpedag	2010.10.12	(30)	Prioritet
(41)	Alm.tilgj	2012.04.13		
(73)	Innehaver	Gridmedia Technologies AS, Selva, 7318 AGDENES, Norge		
(72)	Oppfinner	Ole-Ivar Holthe, Selva, 7318 AGDENES, Norge		
(74)	Fullmektig			

(54)	Benevnelse	Et system for grafikk og dataspillopplevelser på Internett
(57)	Sammendrag	

Oppfinnelsen omhandler et system for fjernspilling og lokal spilling av dataspill via Internett. Oppfinnelsen inkluderer et presentasjonssystem (vist i FIG 1) som bukers i en Web browser plug-in, et fjernspillingsvindu, og eller i dataspillene. Dataspill kan eksekveres på servere og kan lastes ned til brukerens datamaskin. Web browser plug-in'en kan brukes til fjernspille lokalt spille dataspill. En desktop applikasjon kan brukes til å laste ned, synkronisere, og administrere spillressurser lokat, (f.eks 3D modeller med videre). Et fjernspillingsvindu kan også brukes til å fjernspille og lokalt spille dataspill.



INTRODUKSJON

Oppfinnelsen er generelt sett relatert til informasjonsteknologisystemer, og mer spesifikt til et system for å tilby høykvalitets interaktive brukergrensesnitt, dataspill- og fjernspillingsopplevelser på Internett.

5

BAKGRUNN

Det eksisterer en god del kjent teknikk innen det tekniske området til oppfinnelsen. Noen eksempler på dette er:

- Systemer for høykvalitets interaktive brukergrensesnitt på Internett:
 - 10 ○ Microsoft Silverlight og Adobe Flash
 - HTML 4.01 standarden og HTML 5.0 forslaget
 - MPEG-4 standarden
 - System for providing distraction-free content in a Flash-based gaming environment (US patent application number 2010/0175058 A1)
- 15 ○ Systemer for fjernspillingopplevelser på Internett:
 - OnLive

Eksisterende teknikk er hovedsaklig fokusert på å presentere grafikk og multimedia for Web browsere (f.eks. Silverlight, Flash, HTML og MPEG4), og tilby ren fjernspilling over Internett (f.eks. OnLive). Et problem med eksisterende teknikk, i forhold til å publisere dataspillaktige nettsider (dvs. nettsider som ser ut som og fungerer som "main title" dataspill gjør idag), er at dataspillressurser (dvs. 3D modeller, teksturer, filmer, script, kode, med videre) har stor filstørrelse (ofte flere gigabyte). Det medfører videre slike problemstillinger som at det vil ta lang tid å laste ned filene (f.eks. 10 GB tar cirka en time å laste ned på et 3 Mbps nettverk), filene vil ta opp mye plass på brukerens datamaskin (f.eks. 100 nettsider med 10 GB hver, vil ta opp 1 TB diskplass), synkronisering av innholdet vil også bli et problem etterhvert som det oppdateres av innholdsleverandøren, og trygging av innholdet for barn og ungdom er et videre, og vedvarende, problem for Internett innhold (dvs. sikre at barn og ungdom ikke blir eksponert for skadelig innhold, som f.eks. pornografi, vold, "sexual predators," med videre). Fjernspillingsløsninger (som f.eks. OnLive) omgår mange av disse problemstillingene ved at dataspillene eksekverer på servere, slik at en tynnklient programvare applikasjon brukes for å spille dataspillene over Internett (dvs. en audiovisuell strøm sendes fra dataspillene, og en "user input" strøm sendes fra tynnklienten). Fjernspillingsløsninger er dog helt avhengige av Internettforbindelsen i mellom serveren og sluttbruker. Overføringshastigheten vil begrense kvaliteten på den audiovisuelle strømmen. Forsinkelsesraten ("latency") vil medføre treghet i spillopplevelsen (f.eks. at spillkarakteren hopper ett sekund etter at tastatur "h" ble trykt). Hver for seg er de eksisterende teknikkene for svake i forhold til å virkeliggjøre høykvalitets dataspillopplevelser på Internett. Vi kjenner ikke til eksisterende teknikker for kombinerer av de nevnte teknikkene for å virkeliggjøre høykvalitets dataspillopplevelser på Internett.

35

OPPSUMMERING AV OPPFINNELSEN

Det er en hensikt med den foreliggende oppfinnelsen å tilveiebringe et system for å løse de ovenfor nevnte problemene i tidligere kjent teknikk.

- 5 Trekkene definert i de selvstendige kravene vedlagt karakteriserer dette systemet.

Den foreliggende oppfinnelsen tilveiebringer et system som består av en Web browser plug-in (eller Web browser, media player, eller lignende) som inneholder et presentasjonssystem for multimedia, 2D/3D grafikk, dataspill, og fjernspilling; en desktop applikasjon; og/eller et
10 eget klient program/vindu for fjernspilling, som også kan inneholde presentasjonssystemet.

KORT BESKRIVELSE AV FIGURENE

Oppfinnelsen vil i det videre bli beskrevet i detalj med henvisninger til figurene, hvor:

FIG. 1 viser systemarkitekturen til multimedia presentasjonssystemet som brukes i vår Web
15 browser plug-in.

FIG. 2 viser API'en ("application programming interface") til vår Web browser plug-in.

FIG. 3 viser elementene i filformatet til vår Web browser plug-in.

FIG. 4 viser et brukergrensesnitt for "tryggere innhold."

FIG. 5 viser et brukergrensesnitt for en fjernspillingsklient programvare applikasjon.

20 FIG. 6 viser et brukergrensesnitt for interaktiv fjernspilling av dataspill over Internett.

FIG. 7 viser et brukergrensesnitt for en "desktop" programvare applikasjon.

FIG. 8 viser innholdet i en eksempel GXML fil ("MyScene.gxml").

FIG. 9 viser innholdet i en eksempel Java fil for GXML filen ovenfor ("MyScene.java").

FIG. 10 viser innholdet i en eksempel GXML 3D grafikk stage.

25 FIG. 11 viser innholdet i en eksempel Java fil for en GXML dataspillside ("MyGame.java").

DETALJERT BESKRIVELSE

Idéen med oppfinnelsen går hovedsaklig ut på å studere om det er mulig å lage nettsider på samme måte som dataspill lages idag. Dagens "main title" dataspill representerer det beste
30 innen grafikk og multimedia opplevelser på dagens datamaskiner. Nettsider på Internett har tradisjonelt sett benyttet Web browser plug-in'er for presentasjon av grafikk og multimedia. Adobe Flash har hatt den sterkeste markedsposisjonen siden slutten av 90-tallet. Microsoft Silverlight er en utfordrer som har kommet til de senere årene. Det eksisterer mange andre Web browser plug-in'er for ulike formål (f.eks. QuickTime, WildTangent, Viewpoint, Cult 3D, PDF Viewers, med flere). Eksisterende Web browser plug-in'er er meget begrenset i forhold
35 til å tilby brukeropplevelser av samme kvalitet som "main title" dataspill. Teknisk sett er det ingen grunn til at et dataspill ikke skal kunne eksekvere like bra i en Web browser plug-in som i et vindu (f.eks. i Windows). Dataspill eksekverer typisk som en egen prosess med et eget vindu i Windows. Dataspillet kan starte tråder, allokere minne, bruke DirectX, med

videre. En Web browser plug-in eksekverer som regel i samme prosess som Web browseren og kan starte tråder, allokere minne, og bruke DirectX på samme måte som en applikasjon. Overføring av ressursfiler (3D modeller, teksturer, med videre) er et åpenbart problem. Det er også klart at synkronisering er et problem. En av hovedidéene med oppfinnelsen er dertil

5 å løse problemene med overføring og synkronisering av ressursfiler med å tilby brukerne en "desktop applikasjon" som eksekverer på brukerens maskin som en tjeneste (dvs. i "system tray" eller service). Når brukeren er på en (ordinær HTML) nettside som tilbyr Web browser-baserte dataspill opplevelser, kan brukeren f.eks. velge å prøve dataspillet med fjernspilling, eller velge å laste ned dataspillet til egen datamaskin. Dersom brukeren velger nedlasting til

10 egen datamaskin, vil Web browser plug-in'en kommunisere dette til desktop applikasjonen, som starter nedlastingsprosessen (forutsatt at brukeren har installert Web browser plug-in komponenten og desktop applikasjonen). Desktop applikasjonen vil først hente en XML liste over alle filene som skal lastes ned og/eller synkroniseres, hvorpå filene blir lastet ned. Det kan konfigureres hvor mye båndbredde som skal brukes og lignende. Filene kan naturligvis

15 komprimeres (f.eks. zip/rar og lignende) og krypteres (f.eks. RSA og lignende). Med denne løsningen, kan brukeren lukke Web browseren eller navigere til andre sider mens dataspill innholdet lastes ned. Når desktop applikasjonen er ferdig med å laste ned filene, vil det gis beskjed til brukeren (f.eks. med et popup vindu nær system tray). Brukeren kan da starte dataspillet ved å gå tilbake til nettsiden til dataspillet, eller åpne desktop applikasjonen ifra

20 system tray (eller startmenyen e.l.) og starte dataspillet derfra. Desktop applikasjonen vil inneholde en liste over alle brukerregistrerte dataspill nettsteder, hvor brukeren kan starte lokale dataspill i Web browser (eller i lokal klient), starte fjernspilling av dataspillene, samt administrere dataspillene (slette, allokere, synkronisere, komprimere, backup, med videre). Desktop applikasjonen kan også inkludere en rekke ytterligere funksjonaliteter, som f.eks:

25 "instant messaging, remote desktops, FTP verktøy, cache, licenses, heads-up display tools," med videre. Fjernspilling vil typisk foregå i Web browser plug-in'en, i et eget vindu i desktop applikasjonen, eller i en egen fjernspillingsapplikasjon. Et alvorlig problem på Internett idag er at barn og ungdom er utsatt for skadelige opplevelser. Ifølge Medietilsynets rapport om barn og digitale medier 2010, har 14% av barn og ungdom (9 til 16 år) mottatt seksuelle

30 henvendelser på Internett det siste året i Norge. 22% av barna og ungdommene har vært på pornografiske nettsteder. 7% har blitt tilsendt pornografiske bilder og video'er av noen på Internett. 8% har blitt spurt om de kunne sende nakenbilder av seg selv via Internett det siste året. 23% har blitt spurt om de kunne sende personlig informasjon om seg selv på Internett, og 5% sendte denne informasjonen. Flere enn 60% av barna og ungdommene

35 har fått opplæring innen sikker bruk av Internett. Det eksisterer alvorlige problemer relatert til barn og ungdoms bruk av dataspill. 30% av barn (9 til 12 år) og 57% av ungdom (13 til 16 år) har spilt dataspill som har en aldersgrense på 18 år og eldre i Norge. Det er mange fremgangsmåter for å få tilgang til slike spill, som f.eks. piratkopiering, nedlastning, med videre. Det er underlig at eksisterende Internett produkter ikke inkluderer teknikker for å

beskytte barn og ungdom mot slikt. En av hovedidéene med våre teknikker er at vi vil kreve at alt innhold blir gitt en minimums aldersgrense ("age rating"). Innholdsleverandører som ikke "rater" innholdet eller som "rater" galt, vil bli nektet å bruke våre teknikker. Vår Web browser plug-in vil ("default") blokkere alt innhold som er ratet 12 år og eldre. Aldersbevis må fremlegges for å få tilgang til ratet innhold (f.eks. førerkort og/eller tester). Aldersbevis og identifikasjon må videre også fremlegges for å få tilgang til "instant messaging" og slike kommunikasjonsfunksjonaliteter. Denne fremgangsmåten vil sikre at alt innhold distribuert via våre teknikker er trygt for barn og ungdom. Det eksisterer en lang rekke ulike Web filter for å blokkere skadelig innhold på Internett. Problemet med disse er at de må installeres på brukerens maskin. Mange foreldre kjenner ikke til problemet eller at Web filtere finnes. Vår teknikk har den fordel at den fungerer uavhengig av foreldrenes kunnskaper og vilje til å beskytte barn og ungdom imot skadelige opplevelser på Internett. Et annet problem med de eksisterende teknikkene er at Web filter ikke fungerer med alle Web browsere. Ungdommer kan installere disse (f.eks. Google Chrome) for å omgå Web filterene. Vår teknikk har ikke denne begrensningen (ettersom Web browser plug-in'en fungerer likt i alle Web browsere). Et problem med eksisterende fjernspillingsteknikker (f.eks. OnLive) er at de er begrenset av Internettforbindelsen (dvs. overføringshastighet, "latency," quality of service, etc.). Mange brukere av dataspill bruker ofte høy skjermopløsning (ofte high definition), flere skjermer, og i senere tid også stereoskopisk visning (f.eks. med 3D briller med "active shutters" eller "polarized" teknikker). Sanntids overføring av slike audiovisuelle strømmen over Internett krever svært høy båndbredde. Dataspill har også ofte en handling som krever rask reaksjon fra brukeren. Treghet i nettverksforbindelsen ("latency") vil forsinke denne opplevelsen, slik at mange dataspill (f.eks. Trackmania) ikke blir brukbare med fjernspilling. I tillegg kommer andre problemer, som f.eks. pakketap og lignende. Våre teknikker legger opp til å dra nytte av en kombinasjon (eller en "hybrid-løsning") av det å tilby dataspill nettsted opplevelser i Web browseren med en egenutviklet Web browser plug-in, og det å tilby dataspill nettsted opplevelser i (og utenfor) Web browseren med fjernspilling i (og utenfor) Web browser plug-in'en. Vår løsning legger opp til å bruke vår Web browser plug-in, desktop applikasjon, og fjernspillingsklient på sluttbrukerens datamaskin, samt at dataspill lagres og eksekverer på servere. Hovedidéen med vår løsning er at brukere med ordinær Internett forbindelse (etter dagens standard) kan prøve dataspillet med fjernspilling (for å se om det er noe som de er interesserte i å bruke mer tid på). Dersom de er interesserte kan de også velge å laste ned dataspillet til egen datamaskin for bedre kvalitet. Løsningen kan også lett brukes for å lage "hybrid-spill" hvor deler av dataspillet er laget med fjernspilling (f.eks. "game environment, windows, portals, displays," med videre) og andre deler er laget med lokalt lagret innhold:

Hovedidéen i forhold til industriell utnyttelse i Norge er hovedsaklig i forhold til å utvikle og tilby "dataspill nettsteder," som benytter vår Web browser plug-in, desktop applikasjon, og fjernspillingsteknikker for å produsere og tilby nettsteder (på Internett), som ser ut som og

fungerer på samme måte som (sett i forhold til brukeropplevelsen) "main title" dataspill gjør idag. Per idag benyttes hovedsaklig tekst, video, og enkle multimediainnhold for utvikling av nettsteder. Vi tror at dataspill-baserte opplevelser er et naturlig neste steg, i fremtiden, for innhold på Internett/Web. Våre teknikker representerer spesifikke fremgangsmåter for å tilby dataspill nettsteder. Vi planlegger å tilby produkter, tjenester, med videre, for å skape slike dataspill nettsteder for bedrifter, organisasjoner, privatpersoner, med videre, i Norge. Web browser plug-in programvaren er illustrert i FIG. 1, som viser systemarkitekturen til multimedia presentasjonssystemet som brukes i vår Web browser plug-in. Selve plug-in'en er implementert som en ActiveX kontroll (ref. Microsoft) for Microsoft Internet Explorer 3.0 og senere versjoner, samt Netscape NSAPI (eller Gecko) for alle andre Web browsere (f.eks. Firefox, Opera, Google Chrome, Safari, med flere). Andre API'er kan også enkelt benyttes. Våre teknikker er ikke begrenset til et spesifikt operativsystem eller maskinvareenhet, men er tilgjengelig for alle typer operativsystemer, maskinvareenheter, nettverk, med videre. Vi beskriver detaljene i figurene senere i de etterfølgende avsnittene. FIG. 2 illustrerer API'en som benyttes for å lage (eller programmere) innholdet som vises i vår Web browser plug-in. På samme måte som at Microsoft Silverlight benytter Microsoft .NET API for programmering av innholdet som vises i Silverlight, har også vår Web browser plug-in en tilsvarende API. I Adobe Flash benyttes "scriptspråket" ActionScript og en Flash script API. Vår plug-in bruker programmeringsspråket Java 2.0 for programmering av innholdet (merk at selve plug-in'en ikke er laget i Java). Andre programmeringsspråk og script kan bli benyttet i fremtiden. FIG. 3 illustrerer elementene (eller bestanddelene) i filformatet til innholdet som vises i vår Web browser plug-in. Microsoft Silverlight benytter filformatet XAML, som faktisk er definert helt ut ifra API'en (deklarativt). Adobe Flash benytter filformatene SWF og MXML. Vår format er et eget format i XML- og binærform (f.eks. vårt norske patent nr. 318686). FIG. 3 illustrerer navnet på XML-elementene. Attributter og funksjoner beskrives senere i de etterfølgende avsnittene. FIG. 4 illustrerer en "dialogboks" (eller vindu) som vises når brukeren navigerer til en nettside som inneholder "ratet" innhold som har en høyere aldersgrense enn det plug-in'en er konfigurert for (f.eks. innhold er ratet til 18 års aldersgrense). Brukeren kan velge å vise siden uten det ratede innholdet, logge inn for å se ratet innhold (ved aldersbevis), eller å avbryte lastingen. FIG. 5 illustrerer brukergrensesnittet til fjernspillingsklienten. Brukeren velger et dataspill (eller server med dataspill), konfigureringer (f.eks. brukernavn, passord, bitrate for audiovisuell komprimering, skalering av bildestørrelse, rate, port, pakkestørrelse og lignende) og klikker "Koble til" for å koble til serveren ("Oppdater" oppdaterer listen over servere/dataspill). FIG. 6 illustrerer brukergrensesnittet (eller vinduet) som vises etter at fjernspillingsklienten har koblet til serveren. Dette vinduet viser videobildene fra dataspillet i klientområdet ("Spillvisning") til vinduet. Brukerinputt (tastatur, mus, "game pad, joystick," med videre) fanges opp og sendes til serveren (i sanntid). Brukergrensesnittet i FIG. 6 kan være et eget vindu, en endring av det forrige grensesnittet, fullskjerm, eget program, eller lignende. FIG. 7 illustrerer "desktop applikasjonen," som typisk er tilgjengelig i system tray.

Denne applikasjonen kan også være som en tjeneste i operativsystemet (f.eks. NT service) med tilhørende grensesnitt i startmenyen, i kontrollpanelet, eller lignende. FIG. 7 inneholder menyer, verktøylinje, og "tabs," men dette kan variere mellom implementasjoner. "Game sites" (navnet kan naturligvis endres) viser en liste over dataspill som er registrert. Denne listen brukes (av brukeren) til å administrere lokalt nedlastede spillressurser, starte disse spillene i Web browser, starte fjernspilling i Web browser eller fjernspillingsklient (FIG. 5), og lignende. Brukergrensesnittene kan også inneholde annet eller benytte annen "layout." Vår industrielle utnyttelse har naturligvis en utførelsesform som består av en Web browser plug-in (eller Web browser, media player, eller lignende) som inneholder systemet i FIG. 1 til 3, for avspilling av lokale dataspill og fjernspilling, desktop applikasjon, og et eget klient program/vindu for fjernspilling (som også eventuelt kan inneholde systemet i FIG. 1 til 3). Det er mange fordeler med våre teknikker (og utførelsesformer): Nettverksproblemene med fjernspilling overkommes (på en brukervennlig måte) ved at brukerne kan "prøve" ut spillet med fjernspilling (eller bare bruke fjernspilling dersom brukeren er fornøyd med det), og så laste ned dataspillet lokalt dersom det er interessant og ønskelig. Hybrid-løsninger blir også mulige med vår løsning, hvor fjernspilling og lokale dataspillressurser kan kombineres for å lage og tilby en best mulig brukeropplevelse som er tilpasset datamaskinen til brukeren og nettverkskapasiteten (samt kostnadseffektive betraktninger ovenfor innholdsleverandører). Vår løsning gjør det også lett å kombinere dette med andre multimedia, grafikk, med videre for å lage "in-game heads-up displays," tekstlig innhold, og lignende i dataspillopplevelsene.

FIG. 1 illustrerer arkitekturen til presentasjonssystemet som benyttes av Web browser plug-in'en og eventuelt av fjernspillingsklienten. Web browser plug-in'en kan i tillegg til systemet inneholde egne funksjonaliteter som også vil variere mellom plattformer, operativsystemer, og Web browsere (f.eks. programmatisk grensesnitt til JavaScript i HTML, HTML OBJECT og EMBED tagger, oppdatering- og installeringsfunksjonalitet, med videre). Web browser plug-in, presentasjonssystemet, fjernspillingsklienten og desktop applikasjonen er implementert hovedsaklig i C++, og C hvor det er nødvendig. Presentasjonssystemet er konseptuelt laget av en programmerer som benytter et integrert utviklingsverktøy ("integrated development environment (IDE)") (1000) for å utvikle en multimedia applikasjon (Web browser plug-ins, heads-up displays, og/eller dataspill) (1001). Multimedia applikasjonen inkluderer typisk en Java 2.0 virtuell maskin (som f.eks. Sun Java 2.0 Micro Edition virtuell maskin) (1002). Det er også mulig å benytte andre virtuelle maskiner eller "script engines" (f.eks. Lua). GxmlSG (1003) inneholder C++ klassene til "scene grafen." GxmlShow (1004) inneholder klassene til hovedmotoren, spillmotoren, renderingkoden, audiovisuell funksjonalitet, og systemkode. GxmlIO (1006) inneholder klassene for lesing og skriving av GXML filer (dvs. filformatet) og tilhørende verktøykode. GxmlJNI (1005) inneholder Java og C/C++ kode for broen imellom Java og C++ koden (dvs. "Java native interface (JNI)," ref. Java 2.0). GxmlSystem (1007) inneholder systemspesifikk funksjonalitet. Presentasjonssystemet består av delene (1003)

til (1007). Delene (1000) til (1002) er tatt med for å lette forståelsen, og er ikke ment for å avgrense oppfinnelsen til bare utviklingsprosessen for å lage en slik multimedia applikasjon. Ulike GxmlJNI (1005) kodebiblioteker må utvikles spesielt for hver enkelt virtuelle maskin.

- 5 FIG. 2 illustrerer API'en ("application programming interface") til presentasjonssystemet (dvs. API'en som innholdsutviklere benytter for å programmere innholdet som skal vises). Vi har brukt Java 2.0 pakke- og klassekonvensjoner i FIG. 2, men andre programmerings- og scriptspråk kan også benyttes med tilsvarende pakker og klasser. GxmlJNI (1005) kan typisk genereres automatisk ifra denne API'en (dvs. GxmlJNI er en wrapper i mellom C++ og Java).
- 10 Vi vil heretter kalle API'en for "GXML API" for å lette diskusjonen, men navnet kan endres i fremtiden. GXML API består (på toppnivå) av tradisjonell multimedia funksjonalitet i pakkene (2000) til (2004). Tilsvarende funksjonalitet finnes i Microsoft Silverlight og Adobe Flash. Pakkene (2005) til (2008) inneholder klasser for å lage dataspillopplevelser. Microsoft XNA har tilsvarende funksjonaliteter (men ikke Silverlight og Flash). Pakken gxml.sg (2000)
- 15 inneholder klasser som inneholder og administrerer GXML scene grafen, samt Web browser (Browser). Pakken gxml.show (2001) inneholder klasser som administrerer hovedmotoren (Engine), og renderingen (de øvrige klassene). Pakken gxml.av (2002) inneholder audio-visuell funksjonalitet. Pakken gxml.sys (2003) inneholder system funksjonaliteter. Pakken gxml.graphics2d (2004) inneholder 2D grafiske funksjonaliteter. Pakken gxml.game (2005)
- 20 inneholder dataspillmotoren og relaterte funksjonaliteter. Pakken gxml.graphics3d (2006) inneholder 3D grafiske funksjonaliteter. Pakken gxml.game.input (2007) inneholder bruker input klasser (tastatur, mus, game pad, direct input, med videre). Pakken gxml.game.math (2008) inneholder matematikk funksjonaliteter. Ordinære multimediapresentasjoner arver gxml.sg.Scene klassen. Dataspill-baserte presentasjoner arver gxml.game.Game klassen.
- 25 FIG. 8 illustrerer en eksempel GXML fil for ordinært (dvs. 2D og 3D Internett multimedia innhold). Filen inneholder en "movie" med en "scene" som inneholder et bilde, en knapp, og en frihåndstegning ("path"). FIG. 9 illustrerer en eksempel Java fil for interaksjonskode for GXML-filen i FIG. 8. Koden peker Web browser'en til "http://ntnu.no" når brukeren klikker på knappen. FIG. 10 illustrerer en eksempel "stage" (i en scene) for 3D grafikk. Stagen er lagret i filen "can.x" som en DirectX X-fil (andre 3D filformater kan også benyttes).
- 30 FIG. 11 illustrerer en eksempel Java file for en dataspill nettside. Eksemplet inneholder koden for å vise en blå bakgrunn. Koden peker Web browser'en til "http://ntnu.no" når brukeren klikker "Back-knappen" på game pad'en. Både "raster" og "ray-tracing" teknikker kan benyttes.
- 35 FIG. 3 illustrerer XML-elementene i vårt XML-baserte GXML-format. Formatet kan lagres i binær form også (se f.eks. norsk patent nr. 318686). GXML-formatet (XML-basert) starter med ett gxml element (3000), som inneholder ett head element (3100), ett movie element (3200), og muligens andre elementer (3600). Gxml elementet har attributten version, som angir versjonen til GXML-formatet. Head elementet (3100) har p.t. ingen andre attributter,

og inneholder header elementer (3101) til (3112). Title elementet inneholder tittelen til GXML Innholdet. En eksempel GXML fil er illustrert i FIG. 8. Base, meta, og link elementene (3102) til (3104) er de samme som i HTML versjon 4.01. SaferContent (3105) kan beskrive innholdet med hensyn på tryggere innhold for barn og ungdom. Attributten ageRating angir minimum aldersgrense. Attributten href angir en URL til tilleggsinformasjon (f.eks. nøkler for dekryptering av kryptert innhold). SceneHeader (3106) beskriver en scene som er lagret i filen, eller en ekstern fil. Attributten name angir navnet til scenen. Attributten src angir en URL til den eksterne filen. Attributten ageRating angir minimum aldersgrense. Attributten bitrateIDs angir et passende bitrate interval for innholdet. Attributten languageIDs angir et sett med passende naturlige språk for innholdet (f.eks. norsk). Attributten screenIDs angir et sett med passende skjermer (f.eks. 1024 x 768) for innholdet. Attributten machineIDs angir et sett med passende maskinvare konfigurasjoner (f.eks. Quad Core) for innholdet. Elementene ResourceHeader (3107) til ProgramResourceHeader (3111) er henholdsvis for generelle ressurser, bilder, XML-data, tekst, og programkode som er lagret i filen, eller en ekstern fil. Disse elementene har alle attributtene name, src, og ageRating (SceneHeader). ImageResourceHeader (3109) kan i tillegg ha attributtene width, height, og bitCount, som angir bredden, høyden, og bit-dybden til bildet. Header seksjonen kan også inneholde andre elementer (3112). Ingen av header elementene eller attributtene er påkrevd ("mandatory"), de benyttes fritt. Elementer og attributter kan også endres og slettes i forhold til nye behov.

Movie elementet (3200) har p.t. ingen attributter, og inneholder scener (3300) og ressurser (3400) til (3405). Det er også mulig å lage GXML-formatet uten movie elementet, på en slik måte at scener og ressurser legges inn under gxml elementet (3000). Movie elementet er i hovedsak et grupperingselement (for lettere oversikt). Scene elementet (3300) inneholder en scene som kan presenteres til brukeren. Scene elementet (3300) inneholder attributtene name som angir et navn, autoSize som angir automatisk justering av størrelsen til scenen til kontaineren (0 er ingen, 1 er horisontal, 2 er vertikal, og 3 er begge), width er bredden til scenen, height er høyden, mousePointer angir musepeker, backColor er bakgrunnsfargen, antialias angir om "antialiasing" skal brukes, quality angir kvalitetsnivået, framesPerKSec er antall frames per 1000 sekunder, tag er ekstra data, ageRating er minimum aldersgrense, og dataSource er datakilde for elementer. Attributtene bitrateIDs, languageIDs, screenIDs, og machineIDs er de samme som i sceneHeader. Program elementet (3301) inneholder programkode for scenen. Attributten src angir en URL til binærkoden, og srcCode angir en URL til kildekoden. Aktive og plasserbare elementer (3302, 3304, 3305, 3306, 3308, 3309, 3310, 3311, 3312, 3313, 3314, 3315, 3316, 3320, 3321, 3327, og 3328) har attributtene name som angir et navn, x (eller left) som er x-posisjon, y (eller top) som er y-posisjon, width er bredden, height er høyden, enabled angir om elementet er aktivt, visible angir om elementet er synlig, anchorEdge angir hvilke kanter i kontaineren elementet er forankret til (dvs. hver kant av elementet kan automatisk følge "resizing" av kantene til kontaineren), rotation angir rotasjon (i grader "clockwise"), transparency (eller opacity, motsatt) angir

gjennomsiktighet, mousePointer angir musepeker, backColor angir bakgrunnsfarge, tag angir ekstra data, og style angir stil. De aktive og plasserbare elementene har typiske også andre attributter i tillegg. Group elementet (3302) har surface som angir om elementet har en egen rendering surface, dataSource angir en datakilde for elementene, og ageRating er minimum aldersgrense for innholdet. Layer elementet (3303) har name, enabled, visible, tag, bitrateIDs, languageIDs, screenIDs, machineIDs, style, dataSource, og ageRating (som er beskrevet tidligere). Image elementet (3304) har src som angir en URL til innholdet og ageRating. Text (3305) har src, autoSize, og dataSource. Video (3306) har src, srcImage som er en URL til et bilde som vises når videodata ikke foreligger, ageRating, og autoStart som angir om video'en skal begynne å spille automatisk. Audio (3307) er et ikke-visuelt lyd element som har name, src, ageRating, x (eller left), y (eller top), enabled, autoStart, tag, og style. Button (3308) er en interaktiv knapp. RadioButton (3309) inneholder grupperte "radio" knapper, og har dataSource. CheckBox (3310) inneholder en avkrysningsboks, og har value som angir tilstanden, samt dataSource. VSlider (3311) inneholder en vertikal glidekontroll, og har rangeMin som angir minimumsverdi, rangeMax som angir maksimumsverdi, og value som angir verdien. HSlider (3312) inneholder en horisontal glidekontroll, og har rangeMin, rangeMax, og value. VScrollBar (3313) inneholder en vertikal skrollekontroll, og har rangeMin, rangeMax, og value. HScrollBar (3314) inneholder en horisontal skrollekontroll, og har rangeMin, rangeMax, og value. Path (3315) inneholder en frihåndstegning, og har d som inneholder datapath (ref. SVG standarden), pathLength som angir lengden på path'en, fill som angir fyllet (dvs. brush), og stroke som er omrisset (dvs. pen). ComboBox (3316) er en "combo box," og har text som er teksten, appearance som angir hvordan den ser ut (0 er "dropdown combo," 1 er "dropdown list," og 2 er "listbox"), value er valget, og dataSource er datakilden. Rect (3317) er et rektangel, og har name, x (eller left), y (eller top), width, height, rx som angir x-radius for avrundet hjørne, ry som angir y-radius for avrundet hjørne, rotation, enabled, visible, transparency, mousePointer, anchorEdge, tag, style, fill, og stroke. Ellipse (3318) har samme attributter, men cx og cy er x/y-senter for ellipsen, og rx og ry er x/y-radiusene for ellipsen. Line (3319) har de samme attributtene (untatt cx, cy, rx, og ry), og har x1 og y1 som angir x/y-posisjonen til startpunktet, og x2 og y2 angir posisjonen til sluttpunktet. Polyline (3320) er en "polyline" med points som angir punktene til linjene, og stroke. Polygon (3321) er et "polygon" med points, fill, og stroke. BrushSolidColor (3322) er en enkeltfarget "brush" og har name, color, transparency, og tag. BrushImage (3323) er en "bildebrush" og har name, src, titleMode som angir "title mode" til "brush'en" (0 er ingen, 1 er title, 2 er flipX, 3 er flipY, og 4 er flipXY), size som angir størrelsen på "brush'en," transform angir en 3x3 transformasjonsmatrise, color angir en ("default") enkeltfarge, transparency, og tag. BrushLinearGradient (3324) er en lineær gradient "brush" og har name, units som angir "gradient units" (0 er userSpaceOnUse og 1 er objectBoundingBox), startPt angir gradientvektor startpunkt, endPt angir gradientvektor sluttpunkt, transparency, og tag. BrushRadialGradient (3325) er en radial gradient "brush"

og har name, units, center som er senterpunkt for gradient, rx og ry angir x og y radius, origin angir gradient origineringspunkter, transparency, og tag. GradientStop (3326) er en gradient stopper og har offset som angir posisjonen til gradient stopperen, color som angir gradient fargen, og transparency. Stage (3327) er en stage for å vise 3D innhold og har src, ageRating, InPlace som angir in-place rendering, autoStart, og loop som angir om elementet skal loope. Light (3333) brukes i en stage for 3D belysning og har type som angir lystype, diffuse som er diffus farge, specular som er spekulær farge, ambient er miljøfarge, position er 3D posisjon, og direction er 3D retning for lyset. Viewpoint (3334) brukes i en stage for å definere et kamera og har fieldOfView som er synsfelt, position, lookAt som er synsretning, og up som angir opp-retningen til kameraet. Stage (3327) kan inneholde andre elementer (3335). Glyphs (3328) inneholder et sett med "glyphs" som brukes for "fixed text" og har src, fontRenderingEmSize som er fontstørrelsen, styleSimulations som er stilflagg, bidiLevel som er "bi-directional layout level," indices er en streng med glyph spesifikasjoner, og text som er en tekststreng. Animation (3329) brukes til animasjoner og har name, autoStart, loop, frameCount som er antallet "frames," framesPerKSec er "framerate," referenceClock er referanse klokke (f.eks. et annet element), referenceClockSkew er offset på klokken, attrList er en liste med elementattributter som skal animeres (f.eks. MyImage.x), og tag. Ai (3336) (eller keyframe, frame, frm, key, k, f, kf, e.l.) er et "animation item" og har key som angir key frame posisjon, attr er attributten (indeks i attrList), val som er verdien, effect som angir om interpolasjon skal brukes, og notify som angir om en programkode hendelse skal inntreffe. Animation (3329) kan ha andre elementer (3337). Effect (3330) er en shader og/eller fx effekt for et visuelt element og har src, progress som angir progresjon, duration som angir varighet, loop, visible, clock som er klokken til effekten, og tag. Param (3338) er en parameter til effektimplementasjonen og har name som angir navnet på parameteren, og value som er verdien. Effect (3330) kan ha andre elementer (3339). Style (3331) er en stil for andre elementer og har name, src, og targetType som angir typen til de elementene som kan endres av stilen. Si (3340) (eller param, item, attr, e.l.) angir parametere til stilen og har attr (eller name) som angir navnet til attributten, og val (eller value) som er verdien. Style (3331) kan ha andre elementer (3341). Timer (3332) er en "timer" og har name, left (eller x), top (eller y), enabled, interval som er intervallet, og tag. Scene (3300) kan også inneholde andre elementer (3342). Ressursene (3400) til (3404) refererer eller inneholder henholdsvis bilder, tekst, programkode, XML-data, og generelle ressurser. Ressursene har attributtene name, src, og preload som angir om ressursene skal lastes umiddelbart. Det er også mulighet for andre ressurselementer (3405). Movie elementet (3200) kan ha andre elementer (3500). Gxml elementet (3000) kan ha andre elementer (3600). Elementer og attributter er bare enkelt forklart i denne oversikten. Mange av attributtene og elementene har similær betydning som i HTML, SVG, Adobe Flash, Microsoft Silverlight, DirectX, m.v. Farger bruker 4 bytes, alpha, red, green, og blue som "#aarrggbb" (f.eks. #ff0000ff er blå).

Presentasjonssystemet i FIG. 1 kan virke relativt omfattende og komplisert, men det er rett frem for en ekspert med gode kunnskaper og lang erfaringer innen C/C++ programmering, Microsoft COM, Java virtuelle maskiner, DirectX, OpenGL, video koding, og shader språk å implementere systemet. Det anbefales at systemet implementeres som et C/C++ bibliotek.

5 Den virtuelle maskinen bør også struktureres som et C/C++ bibliotek. Det anbefales at tråd håndteringen i den virtuelle maskinen endres slik at tråden for hovedeksekvering taes bort og erstattes med en DoIterate() funksjon som kalles periodisk av Engine klassen i (1004). Dette vil gjøre synkronisering mye enklere. Dette er dog ikke et krav, og oppfinnelsen vil også omfatte virtuelle maskiner som bruker vanlig trådbasert eksekvering. GxmlJNI (1005) kan genereres automatisk ettersom dette er "bridge/wrapper" kode imellom Java og C++.

10 Java pakkene og klassene i FIG. 2 har funksjoner som er markert som "native" (ref. Java). Disse "native" funksjonene skal implementeres i C/C++. Java virtuelle maskiner inneholder som regel dokumentasjon om hvordan slike "Java Native Interfaces" kan genereres. GxmlIO (1006) består av hovedklassene Reader, Writer, Gxml20, og Tools. Reader kan f.eks. enkelt lages med Expat til å parse en GXML fil og instansiere en scene graf GxmlSG (1003). Writer

15 lager en GXML fil ifra en scene graf GxmlSG (1003). Gxml20 inneholder versjonsspesifikk parsingfunksjonalitet for Reader for GXML 2.0. GxmlSystem (1007) inneholder systemkode (f.eks. systeminformasjon, zlib komprimering/dekomprimering, MD5 hash, JPEG, PNG, GIF, DirectX og OpenGL hjelpefunksjonalitet, med videre). GxmlSG (1003) er scene grafen. De

20 samme elementene og attributtene som i FIG. 3 blir benyttet i GxmlSG (1003). Klassene vil også implementere funksjoner (f.eks. kan Image inneholde kode for å dekomprimere JPEG, PNG, og GIF bilder). Vi vil beskrive "public" funksjoner til scene grafen i gjennomgangen av FIG. 2 i neste avsnitt. GxmlShow (1004) inneholder hovedmotoren Engine (eller Engine2D), spillmotoren Game (eller Engine3D), Graphics2D, Graphics3D, audiovisuelle klasser, system

25 klasser, og rendering klasser (Show klasser). Engine klassen er relativt enkel og består av en GXML Reader, en Java VM instans, og en Show klasse. Engine klassen har også en URL attributt, og en DoIterate() funksjon. Et dataprogram (f.eks. en Web browser plug-in) kan benytte Engine klassen som hovedgrensesnitt til presentasjonssystemet. URL'en til en GXML fil må angis. DoIterate() må kalles periodisk (f.eks. så hyppig som mulig). Engine instansen

30 vil i DoIterate() laste ned GXML filen og parse den med Reader. Java programkoden vil bli tilgjengelig, og Engine injekterer denne i Java VM'en og starter denne. Show instansen kan brukes av Engine for å rendre scene grafen. GxmlShow (1004) implementerer forskjellige Show klasser (f.eks. ShowDX8, ShowDX9, ShowDX10, ShowDX11, ShowGL, med videre). Show kan også ha en abstrakt klasse som brukes av Engine, og som Show klassene arver.

35 Show klassene kan inneholde all grafikk bibliotek spesifikk kode for å rendre scene grafen. Hvert element (f.eks. Image) gis et vertex buffer (i f.eks. DirectX). Selve bildet rendres som en tekstur på vertex'ene. Shadere kan brukes for å manipulere renderingen (dvs. effekter). Game klassen er lik Engine, men laster ikke en GXML fil. Rendering gjøres 100% i Java.

FIG. 2 illustrerer Java API'en til presentasjonssystemet. Pakken gxml.sg (2000) inneholder scene grafen. Scene grafens C/C++ representasjon er i (1003) i FIG. 1, og GXML fil format representasjon er i FIG. 3. De fleste elementene i FIG. 3 finnes i (1003) og (2000), som de er beskrevet med attributter tidligere. Java API'en benytter "get" og "set" funksjoner (som er vanlig i Java 2.0) for å lese og endre innholdet til attributter (f.eks. Image::setX(int v)).

Klassene Video, Audio, og Animation har funksjonene play (eller start), pause, stop, med videre. Browser klassen representerer Web browseren (for Web browser plug-in'en). Denne klassen kan brukes til å kontrollere Web browseren, og har funksjonen navigate, som kan brukes til å navigere til en ny nettside. Klassene i gxml.sg (2000) har flere funksjoner, men det blir for detaljert for beskrivelsen (og "scope") av oppfinnelsen. Det er også aktuelt for enkelte implementasjoner at scene grafen til formatet er en egen pakke i tillegg til gxml.sg.

Pakken gxml.show (2001) inneholder Engine klassen, som kan kontrollere hovedmotoren, og har funksjonen play for å spille en spesifikk scene, samt funksjoner (f.eks. setShow) for å endre show modulen som brukes (f.eks. til ShowDX8, ShowDX9, ShowGDI, etc.). Pakken gxml.show (2001) inneholder også klassene ShowDX, ShowGL, ShowGDI, og ShowPIX som brukes for grafikk bibliotek spesifikk rendering (navnene kan variere). Klassene EffectDX og EffectGL benyttes for å kontrollere henholdsvis DirectX og OpenGL effekter. Effekt klassene har funksjonene play (eller start), pause, og stop for kontroll av animerte effekter. Pakken gxml.av (2002) inneholder audiovisuell funksjonalitet. Klassen FilterGraph inneholder GPU/shader-baserte filter grafer. Klassen FilterSource representerer lesing og dekomprimering av audiovisuelle filer (kodek) og har funksjonene open, close, play (eller start), pause, og stop. Klassen FilterEffect representerer en audiovisuell GPU effekt og har funksjonene play (eller start), pause, og stop. Klassen FilterDRM representerer kryptering og dekryptering av en audiovisuell fil eller strøm. Klassen AudioCue representerer lister av lydeffekter og musikk. Klassen Audio3D representerer avspilling av 3D lyd. WaveBank representerer en samlig lyder. De audiovisuelle klassene vil typisk ha lignende funksjoner som i Microsoft DirectX, Windows Media Format, Windows Media Rights Manager, og .NET. Pakken gxml.sys (2003) inneholder system funksjonalitet. Klassen Convert inneholder vanlige Java konverterings-funksjoner som toLong med videre. Klassen Debug inneholder debugging funksjoner som print for å skrive til en debugging konsoll. Klassen Cache brukes til midlertidig lokal lagring. Klassen System inneholder system spesifikk informasjon og funksjonalitet som getCPUCount og lignende. Klassen Network inneholder nettverksfunksjonalitet. Klassen Storage brukes til permanent lokal lagring. Pakken gxml.graphics2d (2004) inneholder funksjonalitet for 2D grafikk. Klassene er tilsvarende som klassene som finnes i Microsoft GDI+ med funksjoner.

Pakken gxml.game (2005) inneholder klasser for å lage dataspill nettsider. Klassen Game brukes til å implementere dataspillet (en derivert klasse implementeres). Klassen GameWnd brukes til å styre vinduet som dataspillet vises i (f.eks. Web browser plug-in vindu eller eget vindu) og har funksjoner som f.eks. setFullScreen for fullskjerm visning. ContentManager klassen brukes til å laste og administrere innholdet (f.eks. 3D modeller) og har funksjoner

som f.eks. load. Pakken gxml.game.graphics3d (2006) inneholder 3D grafikk funksjonalitet. Klassene er tilsvarende som klassene som finnes i Microsoft XNA med funksjoner. Klassen Graphics3D er tilsvarende GraphicsDevice i XNA. Klasser som ikke er angitt noen funksjoner har ingen spesifikke funksjoner. Funksjoner kan bli lagt til senere (og er utenfor "scope" av oppfinnelsen). Indirekte/refererte funksjoner (f.eks. GDI+ og XNA) er også utenfor "scope" av oppfinnelsen. Funksjoner som er utenfor "scope" er bare nevnt for å gjøre det lettere for en tredjepart å implementere oppfinnelsen. Alle kombinasjoner (innenfor "scope") er derfor en del av oppfinnelsen ettersom det naturligvis utgjør ulike utføringer av oppfinnelsen.

FIG. 4 illustrerer brukergrensesnittet som vises (av Web browser plug-in eller lignende) når en GXML-fil inneholder "ratet" innhold (f.eks. at et element har minimums aldersgrense på 18 år). Brukergrensesnittet kan for eksempel vises som et Windows dialog-boks vindu som illustrert i FIG. 4 (tittellinje, minimeringsknapp, lukk/slutt-knapp, og lignende). Andre typer brukergrensesnitt som er (eller blir) vanlige i operativsystemer, plattformer, Web browsere, og lignende, for brukergrensesnittet, er også omfattet av oppfinnelsen. Brukergrensesnittet inneholder informasjon til brukeren, en "combobox" (eller annen liste) for å velge hva som skal gjøres (f.eks. "vis uten 'ratet' innhold" som illustrert i FIG. 4), brukernavn, passord, en avkryssningsboks for om beskjednen alltid skal vises, en konfigureringsknapp, en "ok" knapp (eller lignende) og en avbrytningsknapp. Konfigureringsknappen viser en dialog-boks (eller vindu, Web side, eller lignende) for konfigurering av brukerkontoer, passord, og lignende. Hver bruker som skal få tilgang til "ratet" innhold må fremlegge aldersbevis (f.eks. tester og førerkort). Brukernavn og passord til en aldersverifisert brukerkonto må oppgis for tilgang til "ratet" innhold. Hovedidéen med dette brukergrensesnittet (og løsningen) er å beskytte barn og ungdom fra skadelig innhold på Internett (det forutsettes ikke noen aktive inngrep fra foreldre). Foreldre kan bes om ikke å opprette brukernavn og passord for å hjelpe barn og ungdom med å få tilgang til "ratet" innhold (dvs. hjelpe dem med å omgå systemet).

FIG. 5 og 6 illustrerer brukergrensesnittet til fjernspillingsklienten som brukes på brukerens datamaskin for å fjernspille dataspill (eller spesifikt Web-baserte dataspill) over Internett. Brukergrensesnittet kan for eksempel vises som et Windows dialog-boks vindu som illustrert i FIG. 5 og 6 (tittellinje, minimeringsknapp, lukk/slutt-knapp og lignende vinduselementer). Andre typer brukergrensesnitt som er (eller blir) vanlige i operativsystemer, plattformer, Web browsere, og lignende, for brukergrensesnittet, er også omfattet av oppfinnelsen. FIG. 5 viser brukergrensesnittet som vises ved oppstart (av programmet) og eventuelt etter at en fjernspillingssesjon er ferdig. Brukergrensesnittet inneholder informasjon, og brukeren angir en server (eller et dataspill) i "comboboksen" (eller annen liste). Brukergrensesnittet kan også inneholde konfigurasjonskontroller for å konfigurere slike ting som bitraten som blir brukt for komprimering av audiovisuelle signaler, fullskjerm, "3D vision, grab input, in-game cursor," pakkestørrelse, bildeskalering, rammerate, nettverksport, brukernavn, og

passord. Denne listen er et eksempel og kan naturligvis endres for ulike implementeringer. "Koble til" knappen (eller lignende) brukes for starte fjernspillingssesjonen (mot serveren). Oppdater-knappen (eller lignende) brukes til å oppdatere listen over servere (og lignende). Avbrytingsknappen (eller lignende) brukes til å lukke/avslutte brukergrensesnittet/klienten.

5 Fjernspillingsklienten viser brukergrensesnittet i FIG. 6 når fjernspillingssesjonen eksekverer (untatt i fullskjerm-modus hvor spillvisningen dekker skjermen). Spillvisning viser dekodet videosignal fra serveren (dvs. en komprimert audio og video strøm av skjermbildene til et dataspill som eksekverer på serveren). Videosignalet kan være komprimert med et hvilket som helst video og lyd kodek (f.eks. Microsoft WMV2 og WMA). Fjernspillingsklienten fanger opp brukerinputt (som f.eks. tastatur, mus, game pad, joystick, Direct Input, eller lignende) og sender brukerinputt data komprimert (eller ukomprimert) til serveren (dvs. dataspillet). Fjernspillingsklienten er relativt rett frem og enkel å implementere. Fjernspillingsklienten kan også inneholde presentasjonssystemet for å visualisere 2D, 3D og multimedia innhold (med brukergrensesnitt) i spillvisningen. Integreringen av presentasjonssystemet i spillvisning er relativt rett frem og enkel, og video strømmen (dvs. for fjernspillingen) kan f.eks. 15 vises med et Video element (1003) og (2000) i presentasjonssystemet. På serveren trengs det komponenter for muliggjøre fjernspilling. Det trengs programkode i dataspillet som tar imot brukerinputten (eventuelt dekomprimering av denne) og injekterer/håndterer denne i dataspillet. Dataspillet må også inneholde programkode for å administrere start og stopp av fjernspilling (med konfigurasjoner), fanging ("capture") av skjermbilder og lyd i dataspillet, 20 komprimering av audio og video, samt sende audio/video pakker over nettverket. Det er også mulig at presentasjonssystemet er inkludert i dataspillet. Implementasjon av all denne funksjonaliteten i dataspillet er relativt rett frem og enkelt. Ulike komprimeringsteknikker kan benyttes. I DirectX er de ulike Present-funksjonene en naturlig plass for "capture" av 25 skjermbildet. Oppfinnelsen omfatter også dataspill (og servere) som har slike komponenter for muliggjøring av fjernspilling. Brukergrensesnittene i FIG. 5 og 6 kan settes sammen på mange måter, f.eks. som to dialog-bokser/vinduer, som Microsoft Remote Desktop, slik at vinduet i FIG. 5 endres til vinduet i FIG. 6 når brukeren trykker "Koble til," med videre. Det er også mulig at andre layout og navn benyttes i grensesnittene.

30

FIG. 7 illustrerer "desktop applikasjonen" som typisk er tilgjengelig i "system tray." Desktop applikasjonen vil typisk starte når datamaskinen starter opp (som et ikon i system tray, slik som f.eks. Skype). Desktop applikasjonen kan også implementeres som en tjeneste (f.eks. som en NT Service i Microsoft Windows). Brukergrensesnittet kan for eksempel vises som et 35 Windows dialogvindu som illustrert i FIG. 7 (tittellinje, minimerings- og maksimeringsknapp, lukk/slutt-knapp og lignende). Andre typer brukergrensesnitt som er (eller blir) vanlige i operativsystemer, plattformer, Web browsere, og lignende, for brukergrensesnittet, er også omfattet av oppfinnelsen. Menylinjen (dvs. Meny1, Meny2, Meny3, med videre) inneholder menyfunksjoner som f.eks. exit for avslutning av applikasjonen, backup for å ta backup av

et nedlastet dataspill (eller dataspill ressurser), options for å endre innstillinger, help for å gi tilgang til dokumentasjon, samt about for informasjon om desktop applikasjonen. Nedenfor menylinjen er verktøylinjen (dvs. Knapp1, Knapp2, Knapp 3, osv.). Verktøylinjen inneholder som regel hyppig brukte funksjoner (som også finnes i menyen). Nedenfor verktøylinjen er en "tab" kontroll som viser hovedkategorier av funksjonaliteter (som f.eks. "Game Sites" og annet). Det er aktuelt at menylinjen og verktøylinjen varierer når "tab" kontrollen blir brukt (dvs. hovedkategori funksjonalitetene har ulike menylinjer og verktøylinjer). "Game Sites" er vist som valgt i FIG. 7. Desktop applikasjonen kan også inkludere en rekke hovedkategori funksjonaliteter (f.eks. "instant messaging, remote desktops, FTP verktøy, cache, licenses, heads-up display tools," med videre). "Game Sites tab" inneholder en liste over registrerte dataspill (lokalt nedlastbare dataspill og/eller dataspill for fjernspilling). Listen i figuren viser 3 eksempler på dataspill i listen. En "scrollbar" (til høyre) kan brukes til å "scrolle" i listen. Hvert dataspill i listen er vist med et lite "thumbnail" bilde (Bilde1, Bilde2, og Bilde3) som viser et lite representasjonsbilde av dataspillet. Ved siden av "thumbnail" bildet er det et informasjonsfelt (f.eks. Informasjon 1) som inneholder informasjon om dataspillet (som f.eks. beskrivelser, utgiver/publisher, publiseringsdato, URL, antallet bytes det tar opp på harddisken, antall filer, bruksstatistikk, med videre). Informasjonsfeltet kan også inneholde kontroller (som f.eks. knapper og lignende). Typiske funksjoner som kan utføres på hvert enkelt dataspill i listen er å starte lokale dataspill i Web browser (eller i lokal klient), starte fjernspilling av dataspillene, samt administrere dataspillene (slette, allokere, synkronisere, komprimere, backup, med videre). Dataspill kan typisk registreres i listen i fra Web browser plug-in'en, eller i fra desktop applikasjonen direkte (f.eks. "add game" i menyen). Dersom et dataspill registreres i fra Web browser plug-in'en, kan det f.eks. skje via en Windows message (i Microsoft Windows), registry, en fil, et "remote procedure" kall, eller lignende. Desktop applikasjonen vil hente en XML liste over alle filene som skal lastes ned og/eller synkroniseres, hvorpå filene blir lastet ned. Det kan konfigureres hvor mye båndbredde som skal brukes og lignende. Filene kan naturligvis komprimeres (f.eks. zip/rar og lignende) og krypteres (f.eks. RSA og lignende). En hvilken som helst struktur kan benyttes for denne XML-filen, og f.eks. en liste med "file" og "folder" elementer (inkludert fil/foldernavn, dato, bytes, md5/crc, URL, ageRating, med videre) innenfor et "synch" element. Når desktop applikasjonen er ferdig med å laste ned, kan den sende en melding tilbake til Web browser plug-in'en og/eller fjernspillingsklienten (meldinger kan også sendes om "progress").

Selv om oppfinnelsen har blitt beskrevet med referanse i en legemliggjørelse, vil de som er erfarne innse at variasjoner i form/detalj kan gjøres uten å avvike fra intensjonen i skopet til oppfinnelsen som definert i de vedlagte krav. De spesifikke detaljene ovenfor er ment bare for å være illustrative og skopet av oppfinnelsen er definert av de vedlagte krav. For eksempel kan oppfinnelsen praktiseres som fjernspilling i Web browser og oppfinnelsen kan integreres i operativsystem, Web browsere, maskinvare, og dataspill.

PATENTKRAV

1. System for fjernspilling og/eller lokal spilling av dataspill via Internett, karakterisert ved

5 et presentasjonssystem bestående av en API klasse for å presentere 2D innhold eller 3D stages, en API klasse for å presentere og spille lokale dataspill og/eller med en eller flere fjernspillingssesjoner i et slikt dataspill; og

en desktop applikasjon (7000) som inneholder en liste over "game sites" for dataspill ressurser som kan brukes av Web browser plug-in'en for å presentere/spille lokale dataspill med lokalspillingssesjoner og/eller med fjernspillingssesjoner.

2. System i henhold til krav 1,

karakterisert ved at presentasjonssystemet kan benytte Microsoft DirectX 9, 10 og 11 for rendering/visualisering av 2D og 3D grafikk. GXML-fil eller kode velger versjonen.

3. System i henhold til krav 1,

karakterisert ved at presentasjonssystemet viser et brukergrensesnitt når dette har lest GXML-formatert innhold som inneholder "ratet" innhold med minimum aldersgrense på 12 år og eldre. FIG. 4 viser dette brukergrensesnittet, men design/innhold kan variere.

4. System i henhold til krav 1,

karakterisert ved at serversystemet hvor dataspillet eksekverer også inneholder presentasjonssystemet for "in-game heads-up display" grafiske grensesnitt (1003).

5. System i henhold til krav 1,

karakterisert ved at presentasjonssystemet på sluttbrukers maskin inneholder hele API'en (2000) til (2006).

6. System i henhold til krav 1,

karakterisert ved at presentasjonssystemet på sluttbrukers maskin brukes i en Web browser plug-in i en Web browser.

7. System i henhold til krav 1,

karakterisert ved at presentasjonssystemet på sluttbrukers maskin brukes i et fjernspillingsvindu eller i fullskjerm (6000). Fjernspillingsvinduet kan åpnes som en del av et dataprogram vindu (5000), fra en desktop applikasjon (7000), Web browser, Web browser plug-in, fra startmenyen, "system tray" eller desktop'en til operativsystemet.

8. System for publisering av dataspill via Internett,

karakterisert ved

dataspill eksekverer på servere, dataspillene inneholder presentasjonssystemet som brukes til å presentere 2D og 3D grafikk i dataspillet, eventuelt i tillegg til annen 2D og 3D grafikk, og benytter spesifikt bare enkelte deler av GxmlSG (1003) og GxmlShow (1004);

et fjernspillingsvindu, eventuelt i fullskjerm, (6000) vises på sluttbrukers maskin og bruker presentasjonssystemet til å presentere 2D og 3D grafikk i dataspillet, en audiovisuell strøm fra serverne vises i 2D og 3D grafikken; og

dataspill ressursfiler, som inneholder minst en 3D modell, er lagret på sluttbrukerens maskin, og brukes av presentasjonssystemet for å visualisere grafikk.

9. System i henhold til krav 8,

karakterisert ved at fjernspillingsvinduet åpnes som en del av en programvare applikasjon for fjernspilling av dataspill (5000). Fjernspillingsvinduet kan være en utgave av brukergrensesnittet i (5000), eller vises i fullskjerm-modus.

10. System i henhold til krav 8,

karakterisert ved at Microsoft DirectX 11 benyttes for visualisering med raster-baserte teknikker.

11. System i henhold til krav 8,

karakterisert ved at "ray-tracing" teknikker benyttes for visualisering.

12. System i henhold til krav 8,

karakterisert ved at fjernspillingsvinduet viser et brukergrensesnitt når dataspill innholdet er "ratet" til en minimum aldersgrense på 12 år og eldre. FIG. 4 viser innholdet til en utførelse av dette brukergrensesnittet, men design/innhold kan variere.

13. System i henhold til krav 8,

karakterisert ved at presentasjonssystemet på serverne inneholder hele API'en (2000) til (2006).

14. System i henhold til krav 8,

karakterisert ved at en desktop applikasjon (7000) benyttes for å inneholde en liste over "game sites" for dataspill ressurser som kan brukes av fjernspillingsvinduet for å presentere/spille lokale dataspill med lokalspillingsssesjoner og/eller fjernspillingsssesjoner.

15. Metode for fjernspilling via Internett av dataspill med 3D modeller laget av brukerne, karakterisert ved at metoden omfatter:

en desktop applikasjon som eksekverer på en bruker maskin laster opp en 3D modell til en File Transfer Protocol (FTP) server, og 3D modellen skal brukes til å lastes og vises i et dataspill eller en annen applikasjon med 3D grafikk støtte, kalt "dataspillet";

på et senere tidspunkt initialiserer en bruker opptak av spilldata i fra dataspillet, som eksekverer på en server, der spilldataopptaket inneholder minimum et bilde, en videoframe, et lydsample, eller et brukerinputt sample, og der dataspillet viser 3D modellen;

opptak av spilldata fra spillet, og aggregering av opptaksdata for å frembringe en datarepresentasjon som er passende for overføring over et Internettprotokoll nettverk;

overføring av datarepresentasjonen over et Internettprotokoll nettverk til brukeren;

mottak av datarepresentasjonen i en web browser, web browser plug-in, eller annen programvare eller maskinvare applikasjon med støtte for web browsing, kalt "web klient";

prosessering av datarepresentasjonen i web klienten;

visning eller avspilning av minimum et bilde, en videoframe, et lydsample, og/eller et brukerinputt sample fra datarepresentasjonen i web klienten;

web klienten fanger samtidig opp brukerinputt som overføres aggregert til serveren over et Internettprotokoll nettverk; og

brukerinputten blir tatt imot på serveren, prosessert og injektert i dataspillet.

16. Metode i henhold til krav 15,

karakterisert ved at web klienten er implementert med HTML og JavaScript.

17. Metode i henhold til krav 16,

karakterisert ved at spillvisningen benytter JPEG, PNG eller HTML-canvas bilder.

18. Metode i henhold til krav 15,

karakterisert ved at web klientens konfigureringskontroller kan endres vilkårlig.

19. Metode i henhold til krav 15,

karakterisert ved at spillvisningen benytter presentasjonssystemet.

20. Metode i henhold til krav 15,

karakterisert ved at desktop applikasjonen bruker brukergrensesnittet i FIG. 7.

21. Metode i henhold til krav 15,

karakterisert ved at spillvisningen er implementert med Adobe Flash, Microsoft Windows Media Player, Silverlight, eller Apple QuickTime for bilder, video og/eller lyd.

1/5

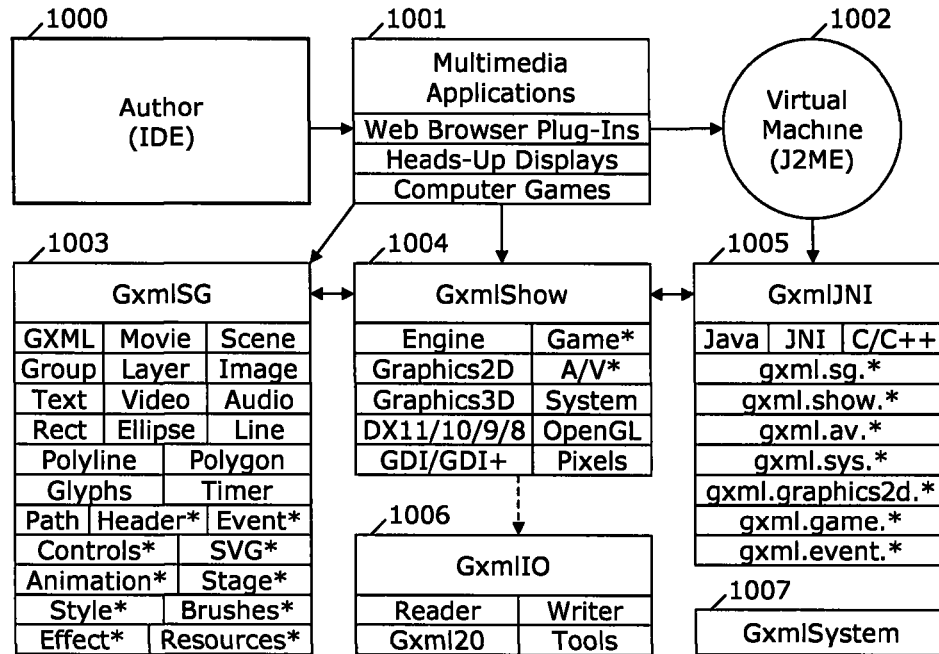


Fig. 1

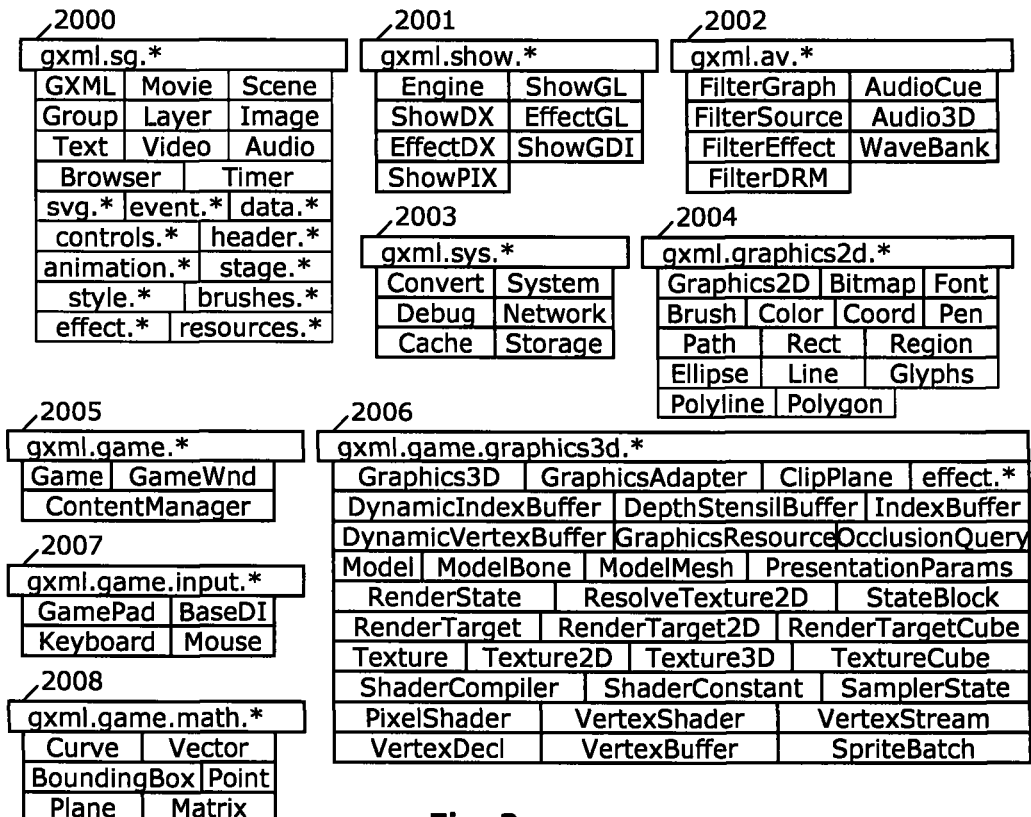


Fig. 2

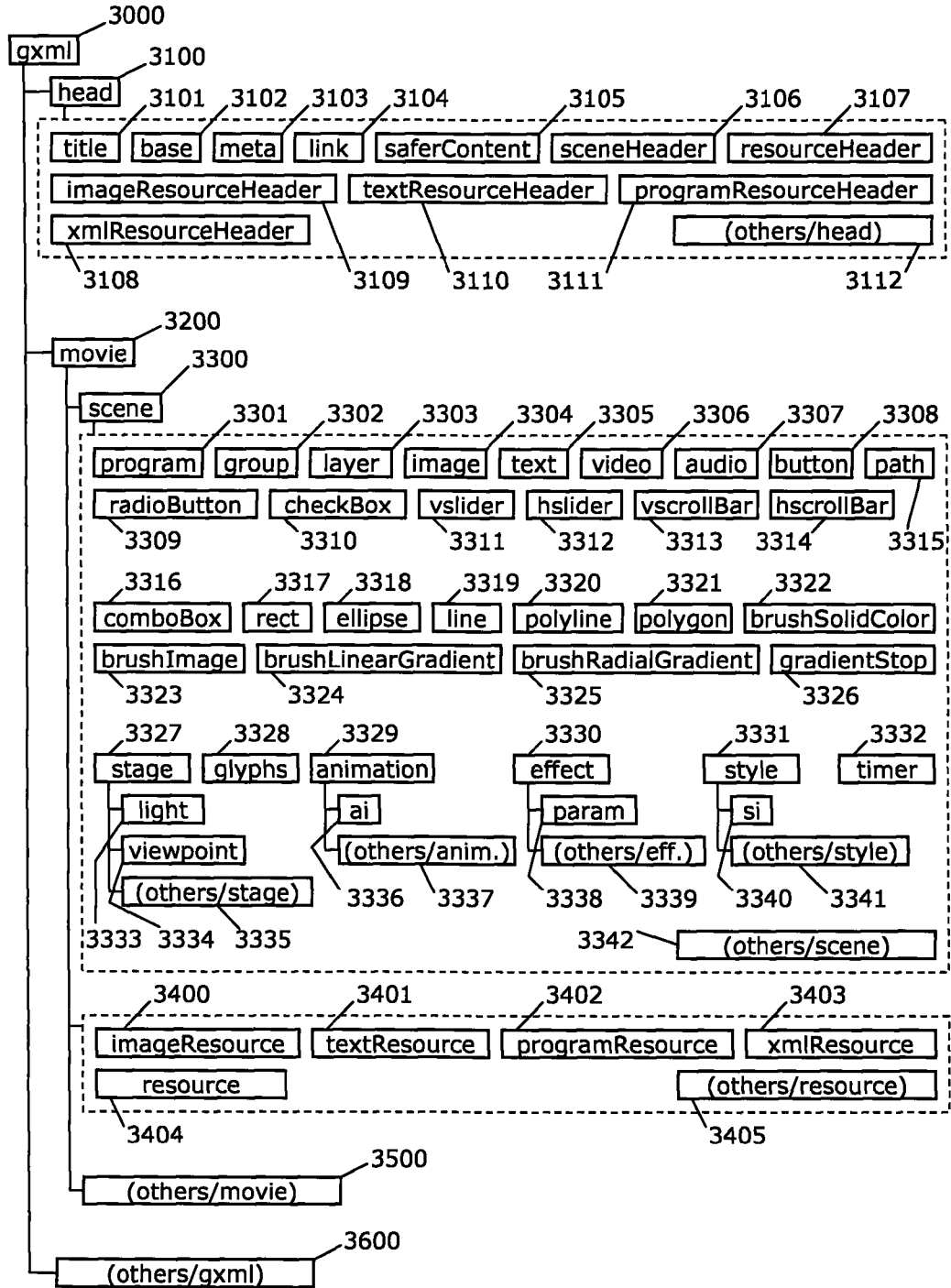


Fig. 3

4000

(Tittellinje) _ X

(Informasjon)

Vis uten "ratet" innhold ▾

Brukernavn ▾

Passord

☒ Alltid vis denne meldingen

Konfigurer.. OK Avbryt

Fig. 4

5000

(Tittellinje) _ X

(Informasjon)

Server ▾

(Konfigureringskontroller)

Koble til Oppdater Avbryt

Fig. 5

6000

(Tittellinje) _ □ X

(Spillvisning)

Fig. 6

7000

(Tittellinje) _ □ X

Meny1 Meny2 Meny3 ...

Knapp1 Knapp2 Knapp3 ...

Game Sites Tab2 ...

Bilde1 (Informasjon 1)

Bilde2 (Informasjon 2)

Bilde3 (Informasjon 3)

...

Fig. 7

```

<?xml version="1.0" encoding="UTF-8"?>
<gxml version="2.5.0.0">
<head>
  <title>Example GXML file</title>
  <saferContent ageRating="16" href="saferKeys.xml"/></head>
<movie>
  <scene name="MyScene" width="600" height="600">
    <image name="MyImage" src="MyImage.png" x="100" y="100"/>
    <button name="MyButton" y="200">
      <rect width="200" height="80" stroke="blue"/>
      <text width="200" height="80">Hello World!</text>
      <image src="btbk"/></button>
      <path name="P1" top="400"
        style="stroke:black;stroke-width:2;fill:white;" d="M18,131C18,131
          10,23 10,23C10,23 41,19 41, 19C41,19 43,37 43,37"/>
    </scene>
    <imageResource name="btbk" src="BtnBack.png" preload="true"/>
  </movie>
</gxml>

```

Fig. 8

```

import gxml.sg.*;
import gxml.sg.controls.*;

public class MyScene extends Scene {
  Image MyImage;
  Button MyButton;

  public MyScene() {
    MyImage = (Image)get("MyImage");
    MyButton = (Button)get("MyButton");
  }

  public void MyButton_OnClick(int button, int shift, int x, int y) {
    Browser browser = getParent().getBrowser();
    browser.navigate("http://ntnu.no", "", "");
  }
}

```

Fig. 9

```

<stage name="MyStage" src="can.x" width="512" height="512">
  <light type="directional" diffuse="#ffffff"
    specular="#99999999" ambient="#ff4c4c4c"
    position="3.0 2.0 -3.0" direction="0.5 -1.0 1.0"/>
  <viewpoint fieldOfView="0.785398"
    position="0.0 1.0 -2.0" lookAt="0.0 0.5 0.0" up="0.0 1.0 0.0"/>
</stage>

```

Fig. 10

```

import gxml.game.*;
import gxml.game.input.*;
import gxml.game.math.*;
import gxml.game.graphics3d.*;

public class MyGame extends Game {
    // Camera
    Vector3 CameraPos;

    public MyGame() {
        // TODO: Add constructor code here
    }

    public void Initialize() {
        // TODO: Add initialization logic here
        CameraPos = Vector3(1, 1, 1);
    }

    public void LoadContent() {
        // TODO: Add content loading here
    }

    public void Update(GameTime gameTime) {
        // Allows the game to exit
        if (GamePad.GetState(PlayerIndex.One).Buttons.Back ==
            ButtonState.Pressed)
        {
            Browser browser = getParent().getBrowser();
            browser.navigate("http://ntnu.no", "", "");
        }

        // TODO: Add update logic here
    }

    public void Draw(GameTime gameTime) {
        graphics.GraphicsDevice.Clear(0xff0000ff);
        // TODO: Add drawing code here
    }

    public void MyGame_OnClick(int button, int shift, int x, int y) {
        // TODO: Add mouse handler here
    }
}

```

Fig. 11