

TIEŠSAISTES SPĒLES SERVERA IZSTRĀDES PRINCIPI ONLINE GAME SERVER DEVELOPMENT PRINCIPLES

Autors: **Artūrs Līcītis** e-pasts: a.licitis@yahoo.com, +37126945256
Zinātniskā darba vadītājs: **Pēteris Grabusts, Dr.sc.ing.**, e-mail: peteris.grabusts@rta.lv,
Rēzeknes Tehnoloģiju akadēmija, Atbrīvošanas aleja 115, Rēzekne, Latvija

Abstract. *The author in this work provides insight into developing the online game using Unity game engine. Themes which are discussed are game server types and client communication with them.*

Keywords: *game development, game server types, online game server, unity game engine.*

Ievads

Videospēles izstrāde ir interesants un tajā pašā laikā sarežģīts process, kas dažkārt tiek salīdzināts ar filmas veidošanu. Mūsdienās ir pieejami dažādi rīki un spēļu dzinī, kā piemēram Unity 3D, Unreal engine, CryEngine3 u.c., ar kuru palīdzību ir iespējams realizēt visdažādākās ieceres un projektus. Ņemot vērā, ka videospēles izstrāde ir garš un sarežģīts process, konkrētajā darbā autors aplūko tiešsaistes spēles serveru veidus, konkrētāk autoritatīvo un neautoritatīvo serveru principus, un klienta komunikāciju ar spēles serveriem, izmantojot Unity 3D platformu, versiju 4.7.

Unity 3D

Unity 3D ir dažādu platformu izstrādes rīks ar kura palīdzību ir iespējams realizēt projektus uz vis dažādākajām platformām, kā Windows, MAC, Android, iOS un citām platformām, galvenā rīko priekšrocība ir iespēja realizēt projektu un ar minimālām, vai pat bez nekādām izmaiņām to īstenot uz jebkuras platformas. [1]. Unity piedāvā bezmaksas personālo izdevumu, kas ļauj izmantot visas spēles dzināja funkcijas, bet ierobežo piekļu beta testiem, projektu izstrādi mākonī un dažādiem atskaišu rīkiem[2]. Rīks ļauj izstrādāt visdažādākos projektus sākot ar 2 dimensiju un beidzot ar 3/2 dimensiju tiešsaistes spēlēm. Kopš komerciālās licences paplašināšanas, nodrošinot bezmaksas pieeju visām spēles dzināja funkcijām, Unity tiek uzskatīts par vienu no labākajiem izstrādes rīkiem priekš neatkarīgajiem spēļu izstrādātājiem[3].

Spēles serveris un klients

Unity dzinis ļauj pavisam vienkārši realizēt tīkla funkcionalitāti, taču atkarībā no iecerētā projekta, tā realizācija var krasī sarežģīties. Unity plaši izmanto tīkla skata (*NetworView*) komponentes, lai apmainītos ar datiem tīklā, tāpēc ir svarīgi saprast, kā tas darbojas. Viena no funkcijām ko nodrošina tīkla skats ir RPC, jeb attālinātās procedūras izsaukumi (*Remote procedure calls*), kas ļauj izsaukt funkciju vienai, vai vairākām tīklā esošajām ierīcēm, kas var būt spēles klienti, vai serveri[4]. Stāvokļa sinhronizācijas metode tiek izmantota, lai regulāri atjauninātu konkrētu datu kopu, piemēram, objektu pozīciju un rotāciju 3D telpā, starp divām, vai vairākām spēles instancēm, kas darbojas tīklā.

Spēles objektu īpašnieku atsekošanai tiek izmantota tīkla ielādes funkcija (*Network Instantiate*)[4]. Ar šīs funkcijas palīdzību tīklā tiek izveidots jauns spēles objekts un objekta īpašnieks ir ierīce, kas izsauc konkrēto funkciju attiecīgi ja funkciju izsauc spēles klients, tad šis konkrētais klients būs objekta īpašnieks un varēs veikt manipulācijas ar to, piemēram, pārvietot objektu 3D telpā.

Izstrādājot tiešsaistes spēli, ir jāņem vērā, ka spēles serveri var realizēt divos veidos, tas var būt autoritatīvs un ne autoritatīvs. Katrai servera pieejai ir savas priekšrocības un trūkumi, atkarībā no realizējamā projekta, viena, vai otra metode var būt piemērotāka.

Autoritatīvais serveris

Izmantojot autoritatīvā servera pieeja, spēles serveris veic visas spēles pasaules simulācijas darbības, pārvalda visas spēles darbības un apstrādā lietotāju ievadu. Katrs klients uz serveri sūta savu ievadu, taustiņ ievadu, vai pieprasījumu izsaukumu formā un tajā pašā laikā saņem nepārtrauktu informāciju par spēles pasaules pašreizējo stāvokli. Klients tiešā veidā nekad neveic nekādas izmaiņas spēles pasaulē, tā vietā klients sūta pieprasījumu spēles serverim, kas apstrādā klienta pieprasījumu un sūta atbildi vienam, vai vairākiem klientiem, lai izskaidrotu, kas, ir noticis pēc konkrētā klienta pieprasījuma[5].

Galvenā šīs servera pieejas priekšrocība ir aizsardzība pret nekorektu/neatļautu klienta ievadu, piemēram, izmantojot trešās personas programmatūru, un tā realizāciju spēles serverī. Spēles klientam nav iespējams krāpties, sūtot informāciju spēles serverim, ka tas ir veicis kādas izmaiņas ar spēles objektiem, jo klientam pašam nav tiesību veikt izmaiņas. Saņemot nekorektu/neatļautu ievadu, spēles serveri klienta pieprasījumu nerealizē, tādā veidā arī netiek ietekmēti pārējie spēlētāji[5].

Autoritatīvā servera principa galvenais trūkums ir laiks, kas tiek patērēts ziņojumu sūtīšanai starp spēles klientiem un serveri, un atpakaļ. Ja spēlētājs nospiež taustiņu, vai izsauc kādu attālināto procedūru, lai pārvietotos spēles pasaulē, ziņojums tiek sūtīts uz serveri, tas tiek apstrādāts, un tikai tad tiek sūtīta atbilde spēles klientam. Kā viena no iespējām, lai kompensētu laika zudumu, ir tā sauktā klienta puses prognoze. Šī metode ļauj klientam pašam atjaunināt savu lokālo spēles versiju, bet tai ir jābūt iespējai, nepieciešamības gadījumā, saņemt labojumus no spēles autoritatīvā servera versijas[5]. Kopumā autoritatīvā servera pieeja ir piemērotākā lieliem projektiem, kur tiek paredzēts spēlētāju skaits, kas ir mērāms simtos, vai tūkstošos, jo tas nodrošina papildus kontroli pār klientu ievadiem.

Ne autoritatīvais serveris

Izmantojot ne autoritatīvā servera pieeju spēles serveris nekontrolē, nepārvalda lietotāja ievadu un tā iznākumu. Spēles klients pat pārvalda savu ievadu un tā rezultātu sūta spēles serverim. Spēles serveris sinhronizē visas veiktās darbības un izmaiņas spēles pasaulē ar klientu. No izstrādes viedokļa šis serveru princips ir vieglāk realizējams, jo ievads tiek apstrādāts klienta pusē un spēles serverim nav jāveic nekādas papildus darbības[5].

Galvenā priekšrocība šai pieejai ir tas, ka nav nepieciešamas izmantot nekādas papildus klienta puses prognozes metodes, jo visiem klienta veidotajiem spēles objektiem īpašnieks ir klients un tas tiešā veidā veic ar tiem manipulācijas, nesūtot pieprasījumus uz spēles serveri. Spēles serverim tiek nodota atbilde par klienta veiktajām manipulācijām. Tas arī ir galvenais trūkums ne autoritatīvajā serveru pieejā, spēles serveris nepārbauda klienta sūtītos rezultātus, kas nozīmē, ka no spēles klienta puses var tikt sūtīts jebkāds pieprasījums, kas var būt modificēts ar trešās puses programmatūru un tas tiks realizēts spēles servera pusē, kas attiecīgi ietekmē arī pārējos spēlētāju.

Ne autoritatīvā pieeja ir vairāk piemērota nelielai kontrolētai videi, kur neatļautais klienta ievads ir vieglāk identificējams. Jāpiemin, ka šai servera pieejai papildus ir nepieciešama administrēšanas sistēma, lai aizsargātu pārējos spēles klienta no neatļauta ievada ietekmes.

Serveru pieeju salīdzinājums

Atkarībā no realizējamā projekta viena, vai otra servera pieeja var tikt izmantota realizācijā. Autoritatīvā servera principa galvenā priekšrocība ir pilnīgi kontrole pār spēles pasauli, kas ļauj validēt lietotāja ievadu un nekorekta, vai neatļauta ievada gadījumā, spēles pasaule netiek ietekmēta, ko nevar teikt par neautoritatīvā servera pieeju, kur visas manipulācijas veic pats spēles klients. Bet galvenais trūkums ir informācijas apmaiņas ātrums starp spēles serveri un klientiem. Pēc lietotāja ievada tiek sūtīts pieprasījums spēles serveri, apstrādājot klienta pieprasījumu, attiecīgi tiek veiktas izmaiņas autoritatīvā servera spēles pasaulē, un

informācija par veiktajām izmaiņām tiek izsūtīta spēles klientiem. Paaugstinātas noslodzes/trafika gadījumos, tas būtiski var ietekmēt spēles klientus, jo rodas aizķere starp lietotāja ievadu un tā realizāciju. Attiecīgi, lai izvairītos no šādas aizķeres, papildus ir nepieciešams izmantot klienta puses prognozes metodi. Bet ne autoritatīvā servera principam nav nepieciešamas papildus prognozes metodes, jo visas darbības realizē pats klients un serverim tiek sūtīta informācija, kādas manipulācijas ir jāveic spēles pasaule, nepārbaudot klienta iesūtīto datu korektumu. Kā viena no atšķirībām arī ir jāmin, ka autoritatīvā servera izstrāde ir daudz sarežģītāka, kā neautoritatīvajam serverim.

Ņemot vērā faktu, ka objekta veidotājs ir tā īpašnieks un autoritatīvā servera gadījumā visu objektu īpašniekam ir jābūt spēles serverim, papildus ir nepieciešams izstrādāt spēles objektu sistēmu, kas ļauj konkrētam spēles klientam pārvaldīt attiecīgi tam paredzētos spēles objektus. Ne autoritatīvā servera gadījumā, serveris realizē klienta ievadu un papildus kontroles sistēmas nav nepieciešams.

Secinājumi

Apkopojot doto informāciju, var veikt sekojošus secinājumus:

1. Autoritatīvā servera izstrāde ir krietni sarežģītāks process, bet tajā pašā laikā nodrošina spēles serverim pilnīgu kontroli pār spēles objektiem un lietotāja ievadu.
2. Autoritatīvā servera princips aizsargā spēles pasauli no nekorekta, neatļauta klienta ievada, tādā veidā netiek ietekmēti citi spēlētāji.
3. Ne autoritatīvā spēles servera pieeja ir vieglāk realizējama un mazina spēles servera noslodzi, jo lietotāja ievads tiek apstrādāts spēles klienta pusē.
4. Ne autoritatīvā servera pieejai ir lielāks risks, neatļautu darbību veikšanai izmantojot trešās personas programmatūru, jo spēles serveri neveic lietotāja ievada validāciju.

Summary

In this paper we take a look at two ways how to create online game server, their weaknesses and gains. The main comparison is between authoritative and non-authoritative game servers how they work and what must be taken to account when creating the online game server. As basis for this paper was used Unity 3D development platform.

Literatūra

1. Build once deploy anywhere: <https://unity3d.com/unity/multiplatform>
2. Get Unity: <https://unity3d.com/get-unity>
3. Top 5 3D Game Engines for Indie Game Developers: <http://www.useoftechnology.com/top-5-3d-game-engines-indie-game-developers/>
4. Legacy Network Reference Guide: <http://docs.unity3d.com/Manual/NetworkReferenceGuide.html>
5. High Level Networking Concepts (Legacy): <http://docs.unity3d.com/Manual/net-HighLevelOverview.html>