

به نام خدا

پروژه شبکه های کامپیوتری - بازی آنلاین دوز

ترم 1401 - 1402

فاز اول :

پیاده سازی پروتکل شبیه UDP :

در این فاز قصد داریم پروتکل UDP را برای بازی آنلاین شبیه سازی کنیم. به این صورت که در این مرحله ، اطلاعات دریافتی از بازی که به صورت پکت (شرح در ادامه) میباشد ، به سمت مقصد یا همان پلیر دوم ارسال میشود. به این منظور ابتدا باید به وسیله سوکت به بازی متصل شوید شماره سوکتی که بازی به آن وصل میشود در فایل config.txt قرار دارد و سپس میتوانید اطلاعات را دریافت کنید. اگر اتصال بازی به سوکت موفقیت آمیز باشد دیتایی مانند زیر برای شما ارسال میکند : H/C,IP,port

H یعنی شما نقش سرور دارید و C یعنی شما نقش میزبان. مثال : H,192.168.1.1,8000

Ip و port هم به تناسب مربوط به سرور یا کلاینت میباشد.

فاز دوم :

پیاده سازی پروتکل شبیه TCP :

در این فاز پروتکل TCP را شبیه سازی میکنیم به این صورت که شما باید از رسیدن پکت ها به مقصد به صورت سالم ، مطمئن شوید. رعایت اصولی مانند ارسال و دریافت ACK/NACK و بررسی CRC پکت ها و رعایت ویژگی های خاص و مربوط به TCP .

فاز سوم :

راه اندازی Firewall :

تابعی در فایل network.py به نام Firewall قرار دارد که پکت هایی که از سمت کلاینت یا همان شبکه می آیند ابتدا به این تابع پاس داده میشوند و سپس در صورت تایید در استک ذخیره میشود و میتوانید از آنها استفاده کنید و اگر تایید نشوند حذف میشوند و در جایی ذخیره نمیشوند.

در این فاز شما باید این Firewall را به درستی برنامه ریزی کنید تا از ورود پکت های خراب و همچنین حملات DDoS جلوگیری کند.

توضیحات توابع :

تمامی کد های مورد نیاز شما در فایل network.py و در قسمت Bridge Service Functions که به صورت کامنت گذاری جدا شده است قرار دارد و کد های فایل های دیگر را نباید تغییری بدهید به جز بخش __init__ که برای اتصال به کلاینت و یا اتصال به سرور باید تغییراتی بدهید :

: Firewall

این تابع به شما packet ورودی به شبکه را میدهد و پس از بررسی و در صورت صحیح بودن (True , packet) بر میگردانید و در غیر این صورت (False , packet) بر میگردانید.

: OnGameData

اگر پلیمر حرکتی انجام دهد و آن حرکت مجاز باشد اطلاعات پلیمر به صورت {0-9} , {0-9} , {0,1} به شما داده میشود به عنوان مثال: 1,0,1
یعنی پلیمر شماره یک (سرور) در ردیف 0 و ستون 1 حرکتی انجام داده است

: OnNetworkData

اگر از سمت شبکه (کلاینت یا حملات DDoS) پکتی دریافت شود. این تابع فرا خوانی میشود. تاخیر مسیر هم در این تابع اعمال میشود.
یعنی پکت هنگامی که دریافت میشود پس از تاخیر 0.1 تا 0.3 ثانیه ای به شما اطلاع داده میشود.

: ReadLastPacket

آخرین پکتی که در استک ذخیره شده است را برمیگرداند.

: SendDataToGame

به عنوان ورودی یک رشته (string) میگیرد و آنرا به بازی به وسیله game_connection ارسال میکند که game_connection را باید به درستی متصل کنید (در فاز 1)

: SendDataToClient

به عنوان ورودی یک رشته (string) میگیرد و آنرا به کلاینت به وسیله network_connection ارسال میکند که network_connection را باید به درستی متصل کنید (در فاز 1)

چالش های برنامه :

در فایل config.py یک سری متغیر قرار دارد که میتوانید پورت شبکه و پورتی که بازی به آن متصل میشود را تنظیم کنید. و همچنین متغیرهای دیگری برای تنظیمات شبکه و چالش های احتمالی وجود دارد که به شرح زیر است. احتمال های رخداد بین 0 تا 1 هستند :

game_port : شماره پورتی که بازی به آن متصل میشود

network_port : شماره پورتی که کلاینت به آن متصل میشود

max_stack_size : فضای ذخیره سازی پکت ها

network_delay : تاخیر شبکه که در توابع توضیح داده شد

DDoS : فعال یا غیر فعال سازی حملات DDoS

pcktCount : تعداد پکتی که در لحظه وقوع حمله DDoS ارسال میشود

Chance_DDoS : احتمال به وقوع پیوستن حمله DDoS

DestroyPacket : فعال یا غیرفعال سازی دستکاری و خراب کردن پکت هایی که در شبکه منتقل میشوند.

Chance_Destroy : احتمال به وقوع پیوستن خرابی پکت ها

PacketLoss : فعال یا غیرفعال سازی گم شدن و نرسیدن پکت ها به شبکه

Chance_Loss : احتمال گم شدن پکت ها

نکات پیاده سازی :

هر بار که از سمت بازی اطلاعاتی دریافت میکنید اگر به بازی OK بفرستید حرکت پلیر ثبت میشود و اگر FAIL بفرستید حرکت آخر پلیر لغو (Undo) میشود.

اگر بازی از طرف پلیر به هر دلیلی بسته شود بازی به شما پیام CLOSE میفرستد.

در تمامی فاز ها به جز فاز 1 باید تمامی چالش ها به درستی اجرا شوند و در روند بازی تداخلی ایجاد نشود.

اگر هر کدام از پلیر ها ارتباطش قطع شود باید در صورت اتصال دوباره بتواند به بازی ادامه دهد.

اگر دیتای اشتباه یا خراب به بازی بفرستید. با پیغامی از سمت بازی روبرو میشوید و باید دوباره از اول اجرا کنید.

پروژه میتواند به صورت تک نفری یا دو نفری انجام شود.