**(به نام خدا)**

**پروژه درس مبانی برنامه نویسی (استاد داوود آبادی)** **ترم: 402/1**

**آرین سعیدکندری 4023521306** **عرفان قاسمیان امیری 402521495**

# (کوییید 2030)

روند کار

چالش ها

چیزهایی که اضافه کردیم

اولین مرحله کار ساخت منوی بازی بود که یک تابع MainMenu تعریف کردیم و در آن تابع با گرفتن ورودی وارد تابع های دیگر میشدیم.

یک چالش در همان ابتدا گذاشتن موسیقی روی بازی بود که ویدیویی که شما لینک کردید قابلیت میوت کردن صدا را نداشت و سرچ زیادی کردیم تا راهش را در dev c++ پیدا کردیم + با اضافه کردن کتابخانه windows.h

یک تابع Settings تعریف کردیم که با انتخاب آن از منو اصلی وارد آن می شد. در این تابع قابلیت قطع یا وصل کردن صدای بازی را پیاده سازی کردیم.

یک تابع Credits تعریف کردیم که در آن اطلاعات سازندگان آمده در اینجا با استفاده از تابع sleep_sec که به ما دادید زمانی را قبل از بازگشت خود به خود به صفحه اصلی مشخص کردیم. و یک شمارش معکوس برای بازیکن پیاده سازی کردیم.

یک تابع Help تعریف کردیم و در آن قوانین بازی و کارکرد کلید ها و علائم در صفحه را به شکل جالبی با نشان دادن صفحه کیبورد و استفاده از رنگ های متنوع گفتیم.

برای استفاده از رنگ ها در ترمینال طبق گفته خودتون از ANSI color codes استفاده کردیم و 8 رنگ اصلی به علاوه تعدای رنگ روشن مورد نیاز را اضافه کردیم.

یک تابع Exit تعریف کردیم که یک ورودی میگیرد و در صورت تایید کاربر از برنامه خارج میشود و در غیر اینصورت به منو اصلی بر میگردد.

در همه این تابع ها با استفاده از ASCII ART سعی در زیباتر کردن منو ها و رابط کاربری کردیم، متن ها را با استفاده از این سایت به ASCII تبدیل کردیم.

https://patorjk.com/software/taag/#p=display&f=Roman&t=COVID%202030

یک تابع Load تعریف کردیم که برای ادامه بازی از آخرین سیو است و در ادامه بیشتر به آن میپردازیم.

در تمام منو ها قابلیتی در نظر گرفتیم که اگر فرد ورودی اشتباه داد به او هشدار دهد و بگوید ورودی درست را وارد کند، این کار را با استفاده از یک متغییر سراسری Input انجام دادیم.

حال تابع NewGame را تعریف کردیم که همه متغییر ها را به اندازه پیش فرض تنظیم میکند و برنامه را آماده دست جدید میکند.

پس از آن تابع NewLevel را تعریف کردیم که اول هر لول جدید اجرا می شود و جای واکسن ها و زامبی ها و تیر ها را به صورت رندوم با استفاده از 14% ()rand مشخص می کند و تعداد آنها را با توجه به لول بازی (متغییر سراسری level). همینطور برای در نظر داشتن جای این متغییر ها که در ادامه نیازمان می شود جای این متغییر ها را در آرایه های مخصوص خودشان که به صورت سراسری تعریف کردیم گذاشتیم.

سپس تابع Round را نوشتیم که در هر راند (با هر ورودی) اجرا می شود، در این تابع یک سری از اتفاقاتی که در هر راند باید بیفتد مثل پرینت تابع statsو Board و گرفتن ورودی از کاربر و حرکت زامبی ها در راند های زوج و تکرار دوباره.

تابع Stats را تعریف کردیم که اطلاعات کلی بازیکن است که همه متغییر های سراسری هستند که توسط تایع ها قابل تغییرند و در طی بازی تغییر میکنند، در این تابع ما تصمیم گرفتیم رنج شات گان (فاصله ای که تیر به هدف میخورد) و ظرفیت خشاب را هم نمایش بدیم.

(Ammo بصورت سه تایی شده: تیرهای بیرون خشاب / تیرهای درون خشاب/ ظرفیت خشاب )

تابع Board که وظیفه پرینت صفحه بازی را دارد، در این تابع برای هر علامت رنگی خاص در نظر گرفتیم.

یکی از چالش های ما در ادامه خورده شدن واکسن ها و تیر ها توسط زامبی ها بود که با چند خط کد در تابع Board این مشکل را حل کردیم تا زامبی ها از روی آنها رد شوند.

تابع بعدی یکی از مهم ترین تابع ها Movement است در این تابع یک ورودی از کاربر میگیریم که یا از نوع حرکت یا از نوع شلیک یا ریلود یا خروج یا منو آپگرید یا منو بازی یا سیو است.

اگر ورودی از نوع حرکت بود P را حرکت دادیم و بررسی کردیم که اگر واکسن یا تیری گرفته بود از صفحه حذف و به مشخصات اضافه شود و همینطور اگر نزدیک زامبی شدیم از جان هایمان کم شود.

اگر ورودی از نوع شلیک بود چک کردیم اگر تیر در شات گان بود در همان سمت شلیک شده تا رنج شات گان گشتیم و اگر زامبی بود آنرا حذف کرده و پیام شلیک دادیم.

اگر ورودی M بود وارد منوی بازی شدیم که مانند منوی اصلی است با این فرق که پس از تمام شدن کار به صفحه بازی بر میگردیم نه منوی اصلی.

اگر ورودی U بود وارد منوی آپگرید میشویم و در این منو با توجه به لول ما میتوانیم ارتقاهایی انجام دهیم. هزینه ارتقاها با فرمول محاسبه شده و برای تراکنش موفق رنگ سبز و برای تراکنش ناموفق رنگ قرمز انتخاب شده.

اگر ورودی E بود وارد تابع Exit میشویم و در صورت تایید بازیکن به منوی اصلی برمیگردد.

اگر ورودی R بود تعداد تیرهای در دست بازیکن و تیر های درون خشاب چک میشود اگر بازیکن تیر داشت و خشاب هم خالی بود خشاب تا جایی که یا تیر تمام شود یا خشاب پر شود پر میشود، در غیر اینصورت پیام خشاب پر است یا تیر نداری به بازیکن می دهد.

یک چالش این بود که بعد از هر ورودی نخواهیم اینتر بزنیم و خودش وارد شود، برای اینکار از کتابخانه conio.h و ()getch به جای cin استفاده کردیم که ورودی را کاراکتر به کاراکتر می گیرد.

اگر ورودی C بود تابع Save را داریم که مختصات و اندازه تمام متغییر ها را با استفاده از کتابخانه fstream در یک فایل تکست ذخیره میکند که بعدا میتوان از منوی اصلی بازی را از جای ذخیره شده ادامه داد.

ذخیره کردن بازی چالش خیلی بزرگ و پر باگی بود که با تلاش و امتحان بسیار درست شد.

پس از گرفتن ورودی تابع Movement چک میکند که اگر به واکسن یا تیر رسیده بودیم آنرا حذف و امتیازش را به ما بدهد.

همچنین چک می کند که اگر به مقصد D رسیده بودیم و همه واکسن هارا گرفته بودیم به لول بعد بریم و در غیر اینصورت پیام می دهد تا واکسن ها را بگیریم.

این تابع همچنین چک میکند اگر زامبی ای در نزدیکی ما بود یکی از جان هایمان کم شود و اگر جان هایمان به صفر رسید تابع Lose فراخوانی میشود.

تابع Lose آپشن بازگشت به منوی اصلی یا شروع بازی جدید را به ما می دهد.

پس از تابع Movement به تابع حرکت زامبی ها Zombie می رسیم که پرباگ ترین و سخت ترین بخش ماجرا بود.

در این تابع با توجه به موقعیت پلیر زامبی ها را حرکت دادیم اگر پلیر و زامبی در یک ستون بودند زامبی ها به سمت پلیر (بالا یا پایین) و اگر در یک ستون نبودند زامبی ها فقط افقی به سمت بازیکن میرود.

چالش های اصلی بازی با حرکت زامبی ها  به وجود آمد، از خورده شدن واکسن ها و تیر ها توسط زامبی ها تا پرش چندتا چندتای زامبی ها و اسپاون کمپ کردن زامبی ها که به فرد راه فراری نمیداد و ...

یکی از چالش های دیگر روی هم افتادن تیر ها و زامبی  ها و واکسن ها در ابتدای لول بود که با چند شرط if حل شد.

پیام هایی در زیر صفحه بازی نسبت به تعداد کیل هایی که گرفتیم نوشته می شود.

در آخر تابع Win را زدیم که اگر بازیکن لول 10 را به موفقیت پشت سر بگذارد برنده خواهد شد.

10 لول برای این در نظر گرفته شد چون در لول های بالاتر تعداد علائم موجود در صفحه خیلی زیاد میشد و تعداد بالای زامبی ها بازی را غیرممکن میکرد.

روی رنگ های متون و علائم و پیام های ارور در صورت ورود کاراکتر نامربوط و در کل شکل بازی سعی بسیار شده.

بازی در حال حاضر مقداری سخته که اگر سه خطی که در کد مشخص شده در تابع زامبی را کامنت کنیم ساده تر می شود.

با تشکر از شما