

با تعالی

عنوان:

طراحی کنسول بازی – بازی T-REX

عنوان دوره:

آزمایشگاه ریزپردازنده

نام استاد:

وحید

اعضا:

محمدعلی مجتهد سلیمانی - ۹۹۲۰۲۳۰۳۹

حامد خسروی - ۹۹۲۰۲۳۰۱۰

محمدجواد نجفی - ۹۹۲۰۲۳۰۳۴

تاریخ:

۱۴۰۳/۰۲/۲۸

چکیده

این پروژه طراحی و پیاده‌سازی بازی **T-Rex** ، که از بازی آفلاین موجود در گوگل کروم الهام گرفته شده است، با استفاده از **microcontroller Arduino** ارائه می‌دهد. هدف اصلی بازسازی مکانیک‌های بازی است، جایی که بازیکن یک دایناسور **T-Rex** در حال دویدن را کنترل می‌کند که باید از روی موانع بپرد، با استفاده از اجزای فیزیکی و برنامه‌نویسی نرم‌افزار. اجزای سخت‌افزاری شامل یک برد **Arduino Uno** ، یک نمایشگر **LCD** برای نمایش تصاویر بازی و دکمه‌های فشاری برای ورودی کاربر می‌شود. نرم‌افزار با استفاده از **IDE Arduino** توسعه یافته و منطق بازی برای پردازش ورودی‌ها، به‌روزرسانی نمایشگر و تولید موانع برنامه‌ریزی شده است.

پروژه به دو بخش اصلی تقسیم می‌شود: توسعه نرم‌افزار و مونتاژ فیزیکی. توسعه نرم‌افزار بر نوشتن کد بهینه برای شبیه‌سازی محیط بازی و اطمینان از تعامل روان بین کاربر و بازی تمرکز دارد. مونتاژ فیزیکی تمامی اجزا را بر روی یک برد ترکیب می‌کند، که پایداری و عملکرد صحیح را تضمین می‌کند.

آزمایش‌های گسترده‌ای برای ارزیابی عملکرد سیستم انجام شد، تا اطمینان حاصل شود که بازی به‌طور روان اجرا می‌شود، به ورودی‌های کاربر به‌درستی پاسخ می‌دهد و وضعیت صحیح بازی را بر روی نمایشگر **LCD** نمایش می‌دهد. نتایج نشان می‌دهد که سیستم پیاده‌سازی شده به‌طور موفقیت‌آمیز مکانیک‌های اصلی بازی **T-Rex** را بازسازی می‌کند و یک تجربه کاربری تعاملی و جذاب فراهم می‌آورد. چالش‌هایی مانند حذف نویز دکمه‌ها و بهینه‌سازی نرخ تازه‌سازی نمایشگر برای بهبود عملکرد کلی رفع شدند.

این پروژه نه تنها پتانسیل **Arduino** در ایجاد بازی‌های تعاملی را نشان می‌دهد، بلکه به‌عنوان یک ابزار آموزشی برای درک طراحی سیستم‌های مبتنی بر **microcontroller** نیز خدمت می‌کند. بهبودهای آینده می‌تواند شامل گنجاندن گرافیک‌های پیچیده‌تر و افکت‌های صوتی برای غنی‌سازی تجربه بازی باشد.

مقدمه

❖ پیش زمینه

بازی T-Rex یک بازی offline معروف در مرورگر google chrome است که برای سرگرم کردن کاربران در زمان قطع اتصال اینترنت طراحی شده است. در این بازی بی پایان، بازیکن یک دایناسور T-Rex پیکسلی را کنترل می کند که از میان یک منظره بیابانی حرکت می کند و باید از روی موانعی مانند کاکتوس ها بپرد و از زیر پرندگان پرنده خم شود. بازی به صورت بی پایان ادامه دارد و با افزایش امتیاز بازیکن، دشواری آن افزایش می یابد. سادگی و جذابیت این بازی آن را به یک کاندید عالی برای بازسازی با استفاده از سیستم های مبتنی بر microcontroller تبدیل کرده است.

❖ هدف

هدف اصلی این پروژه طراحی و پیاده سازی یک نسخه کاربردی از بازی T-Rex با استفاده از microcontroller Arduino است. این شامل ایجاد اجزای نرم افزاری لازم برای شبیه سازی مکانیک های بازی است که به کاربران امکان می دهد از طریق ورودی های فیزیکی با بازی تعامل کنند. این پروژه هدف دارد تا کاربرد عملی برنامه نویسی microcontroller در ایجاد سیستم های تعاملی را نشان دهد.

❖ دامنه

این پروژه شامل بخش های کلیدی زیر است:

مونتاژ سخت افزار: این شامل انتخاب و مونتاژ اجزای الکترونیکی لازم، از جمله برد Arduino ، یک نمایشگر LCD برای خروجی بصری و دکمه های فشاری برای ورودی کاربر است .

توسعه نرم افزار: نوشتن منطق بازی در Arduino IDE ، پردازش ورودی ها، به روزرسانی نمایشگر و تولید موانع .

یکپارچه سازی و آزمایش: ترکیب اجزای سخت افزار و نرم افزار بر روی یک برد، اطمینان از کارکرد هماهنگ آنها و آزمایش سیستم برای تأیید عملکرد و کارایی.

❖ کارهای انجام شده:

طراحی مدار: ایجاد یک نمودار دقیق از مدار که اتصالات بین **Arduino** ، نمایشگر **LCD** و **push button** را نشان می دهد .

شبیه سازی: ما پیاده سازی اجزای فیزیکی خودمان را ابتدا به صورت شبیه سازی در برنامه **proteus** اجرا و بررسی کردیم و بعد اقدام به پیاده سازی عملی کردیم.

مونتاژ سخت افزار: تنظیم اجزای فیزیکی بر روی یک برد برد، اطمینان از اینکه تمامی اتصالات محکم و عملکردی هستند.

توسعه نرم افزار: نوشتن کد در **Arduino IDE** برای پیاده سازی منطق بازی، از جمله حرکت شخصیت، تولید موانع و پیگیری امتیاز .

آزمایش و رفع اشکال: انجام آزمایش های دقیق برای شناسایی و حل هرگونه مشکل، اطمینان از اجرای روان بازی و پاسخگویی دقیق به ورودی های کاربر .

مستندسازی فرآیند: تهیه یک گزارش جامع که طراحی، پیاده سازی، آزمایش و نتایج پروژه را به همراه چالش های مواجه شده و راه حل های اتخاذ شده توضیح می دهد . تا پایان این پروژه، ما هدف داریم یک بازی **T-Rex** کاملاً کاربردی ارائه دهیم که یکپارچه سازی سخت افزار و نرم افزار در یک سیستم جاسازی شده را به نمایش می گذارد و یک تجربه آموزشی ارزشمند در کاربردهای **microcontroller** فراهم می کند.

ابزار و روش‌ها

❖ لیست قطعات

قطعات زیر برای پیاده‌سازی بازی T-Rex با استفاده از Arduino مورد نیاز است:

Arduino UNO board

breadboard: برای مونتاژ مدار

نمایشگر LCD: نمایشگر ۱۶*۲ کاراکتری

نمایشگر E LCD: به Arduino متصل کردیم.

push button: یک قطعه push button کوچک

سیم‌های اتصال: سیم‌های jumper برای اتصالات breadboard

مقاومت‌ها: مقاومت‌های مختلف (انواع مختلف)

بسته به جزئیات طراحی ما، ممکن است به قطعات بیشتر یا کمتری نیاز داشته باشیم.

❖ طراحی مدار

طراحی مدار شامل اتصال Arduino به نمایشگر LCD و push button است. نمایشگر LCD برای نمایش گرافیک‌های بازی استفاده خواهد شد، در حالی که push button برای ورودی بازیکن (مثلاً برای پرش دایناسور T-Rex) استفاده می‌شود.

❖ اتصالات نمایشگر LCD:

RS (انتخاب رجیستر) به pin دیجیتال ۱۲

E (فعال سازی) به pin دیجیتال ۱۱

D4 به pin دیجیتال ۵

D5 به pin دیجیتال ۴

D6 به pin دیجیتال ۳

D7 به pin دیجیتال ۲

VSS به GND

VDD به V₅₊

اتصال push button:

یک pin push button به pin دیجیتال ۸

pin دیگر push button به GND

یک مقاومت pull-down (۱۰ کیلو اهم) بین pin دیجیتال ۸ و GND

اتصالات تغذیه (V₅ و GND) برای Arduino و اجزای breadboard باید به درستی مدیریت شوند تا از پایداری مدار اطمینان حاصل شود.

❖ پیاده سازی نرم افزار

نرم افزار با استفاده از Arduino IDE توسعه یافته است. اجزای کلیدی نرم افزار شامل موارد زیر است:

تنظیمات ابتدایی: تنظیم نمایشگر LCD و pin های ورودی

منطق بازی: مدیریت حرکت شخصیت، تولید موانع و تشخیص برخورد

به روز رسانی نمایشگر: به روز رسانی مداوم نمایشگر برای نمایش وضعیت فعلی بازی


```
#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

const int buttonPin = 8; // Push button pin
bool isJumping = false; // State of the T-Rex

void setup() {
  lcd.begin(16, 2);
  pinMode(buttonPin, INPUT);
  lcd.print("T-Rex Game");
}

void loop() {
  int buttonState = digitalRead(buttonPin);
  if (buttonState == HIGH) {
    isJumping = true;
    lcd.clear();
    lcd.print("T-Rex Jumps!");
  } else {
    isJumping = false;
    lcd.clear();
    lcd.print("T-Rex Running");
  }
  delay(200); // Delay to simulate game speed
}
```

❖ آزمایش و رفع اشکال

پس از شبیه سازی سخت افزار در **proteus** و بارگذاری نرم افزار، سیستم باید به طور کامل آزمایش شود. مراحل زیر باید در طول آزمایش انجام شوند:

تأیید اتصالات: اطمینان حاصل کنید که تمامی اجزا به درستی مطابق با نمودار مدار متصل شده اند.

آزمایش عملکرد: بررسی کنید که آیا نمایشگر **LCD** گرافیک های بازی را نمایش می دهد و آیا **push button** به درستی دایناسور **T-Rex** را کنترل می کند.

آزمایش عملکرد: اطمینان حاصل کنید که بازی به طور روان اجرا می شود و هیچ تأخیر یا وقفه ای ندارد.

❖ مشاهدات

هرگونه مشکلی که در طول آزمایش که با آن مواجه می شویم و نحوه حل آن ها را مستند می کنیم. عملکرد سیستم را یادداشت کرده و هر ناحیه ای که می توان در آن بهبودهایی انجام داد را ذکر می کنیم.

با دنبال کردن این مراحل و استفاده از قطعات لیست شده، می توانید بازی **T-Rex** را با استفاده از **Arduino microcontroller** با موفقیت پیاده سازی کنید.

نحوه تولید صدا

شاید یکی از ساده ترین روش هایی که استفاده می شود، اضافه کردن **loud speaker** به مدار باشد. زیرا هزینه اضافه کردن مدار های جانبی را ندارد، همچنین **loud speaker** حجم زیادی را هم اشغال نمی کند.

بلندگوها برای بازتولید طیف گسترده ای از فرکانس های صوتی طراحی شده اند، که آن ها را برای موسیقی، سخنرانی و افکت های صوتی پیچیده مناسب می سازد. آن ها صدای با کیفیت بالا در یک محدوده فرکانسی وسیع تولید می کنند، که پخش صوتی شفاف و دقیق ارائه می دهد. بلندگوها از یک **diaphragm** که توسط یک الکترومغناطیس یا عنصر پیزوالکتریک حرکت می کند، برای تولید صدا استفاده می کنند. حرکت **diaphragm** امواج صوتی را در طیف گسترده ای تولید می کند. استفاده در سیستم های صوتی، سیستم های اعلام عمومی، دستگاه های چندرسانه ای و هر کاربردی که نیاز به بازتولید صدای با کیفیت بالا دارد. مزایای بلندگو نسبت به **buzzer**: محدوده فرکانسی وسیع، کیفیت صدا، کاربردپذیری و تجربه خوب کاربری. در حالی که **buzzer** ها برای عملکردهای هشدار و سیگنال دهی ساده مؤثر هستند، بلندگوها تجربه صوتی برتری ارائه می دهند با توانایی بازتولید طیف گسترده ای از فرکانس ها و تحویل صدای با کیفیت بالا. انتخاب بین **buzzer** و بلندگو بستگی به نیازهای خاص کاربرد و اهمیت کیفیت صدا در استفاده مورد نظر دارد.

در این بازی فرکانس **note** ها درون فایل **pitches.h** قرار گرفته اند. **melody** نیز درون فایل **melody.h** ذخیره شده اند.

نحوه تعریف کاراکتر دلخواه برای LCD و setup بازی

LCD های کاراکتری، حافظه‌ای به نام **CGRAM** دارند که می‌توانیم در آن تنها ۸ کاراکتر دلخواه تعریف نماییم. برای این کار ابتدا می‌بایست به‌وسیله نرم افزار **LCD custom character** کاراکتر دلخواه خود را طراحی کنیم، سپس کد آن را خروجی گرفته و به‌وسیله آردوینو به ال سی دی معرفی کنیم.

در **setup** اولیه آردوینو این کاراکترها به ال سی دی معرفی می‌شوند. همان‌طور که پیش‌تر هم اشاره کردیم، تنها می‌توانیم ۸ کاراکتر را به ال سی دی خود معرفی کنیم. این ۸ خانه، از خانه صفر تا هفتم توسط تابع **lcd.createChar(num, data)** در آردوینو قابل دسترسی هستند.

```
lcd.createChar(0, dino);  
lcd.createChar(1, cacti);  
lcd.createChar(2, bird);  
lcd.createChar(3, block);
```

در ابتدای شروع بازی، تابع **showCrashScreen()** صدا زده می‌شود تا آهنگ و متن‌های اولیه را مشاهده می‌کنیم سپس در این تابع، برنامه منتظر زدن دکمه شروع بازی توسط کاربر می‌مانیم. بعد از زدن کلید شروع توسط کاربر، وارد حلقه‌ی اصلی برنامه یعنی **loop** می‌شویم. سپس الگوریتم برنامه با استفاده از توابع مختلفی که تعریف شده است، بازی را هدایت می‌کند. برای مثال:

drawRandChar()

یک کاراکتر را به‌صورت تصادفی روی صفحه نمایش ظاهر می‌کند.

printScore()

امتیاز کسب شده تا این لحظه را در وسط خط اول نمایشگر چاپ می‌کند.

showCrashScreen()

هنگامی که کاربر بازی را می‌بازد، این تابع روی صفحه پیام "Game Over!" را نشان داده و ملودی سوختن بازی را پخش می‌نماید. سپس امتیاز کسب شده و بهترین امتیاز را روی صفحه نشان می‌دهد.

روند اجرای بازی به این صورت است که با ایجاد اشکال تصادفی روی صفحه و سپس شیفت دادن آن‌ها به سمت چپ، بازی به جریان افتاده و از طریق محاسبه‌ی مدت زمان پرش و موقعیت فعلی تی-رکس، برخورد با اجسام تشخیص داده می‌شود.

نحوه نصب و اجرای بازی

نتیجه گیری

در طول توسعه بازی T-Rex با استفاده از **Arduino microcontroller**، چندین چالش مواجه شدیم که حل شدند، که هم پیچیدگی و هم ارزش آموزشی این پروژه را به نمایش گذاشتیم.

چالش‌های پیش رو و راه‌حل‌های پیاده‌سازی شده

لغزش دکمه:

چالش: دکمه‌های مکانیکی اغلب برای یک فشار چندین ورودی ثبت می‌کنند به دلیل لغزش تماس.

راه‌حل: یک روتین لغزش در کد **Arduino** پیاده‌سازی کردیم تا این سیگنال‌های نادرست فیلتر شوند و تشخیص ورودی قابل اعتماد فراهم شود.

محدودیت pin‌های I/O:

چالش: **Arduino UNO** و **NANO** تعداد محدودی pin‌های I/O دارند که اتصال چندین جزء را دشوار می‌سازد.

راه‌حل: مدیریت کارآمد pin‌ها با استفاده از ترکیبی از pin‌های دیجیتال و آنالوگ و تکنیک‌های چندپلاک‌سازی برای اتصال نمایشگر **LCD**، **push button** و سایر لوازم جانبی پیاده‌سازی کردیم.

کنتراست نمایشگر LCD:

چالش: اطمینان از خوانایی نمایشگر **LCD** در شرایط نوری مختلف نیاز به تنظیم دقیق کنتراست داشت.

محدودیت‌های حافظه:

چالش: بردهای **Arduino** حافظه محدودی دارند که نیاز به مدیریت دقیق داشت تا منطق بازی و عملکردهای نمایشگر جا شوند.

راه حل: تکنیک‌های بهینه‌سازی کد، مانند استفاده از متغیرهای محلی و ساختارهای داده کارآمد، برای به حداقل رساندن استفاده از حافظه پیاده‌سازی شدند.

اجرای روان بازی:

چالش: اطمینان از اینکه بازی بدون لگ اجرا می‌شود نیاز به بهینه‌سازی کد برای عملکرد داشت.

راه حل: حلقه اصلی بازی بهینه‌سازی شد و به‌روزرسانی‌های نمایشگر به طور کارآمدی مدیریت شدند تا تجربه بازی پاسخگو حفظ شود.

❖ پیاده‌سازی در **Proteus** و **Arduino**

Arduino: IDE Arduino برای توسعه منطق بازی استفاده شد. این شامل تنظیم نمایشگر **LCD**، مدیریت ورودی‌های دکمه و پیاده‌سازی حلقه اصلی بازی بود. توابع کلیدی برای عملکرد و کارایی حافظه بهینه‌سازی شدند تا اطمینان حاصل شود که بازی به سرعت به ورودی‌های کاربر پاسخ می‌دهد و به‌طور روان روی سخت‌افزار اجرا می‌شود.

Proteus: نرم‌افزار **Proteus** برای شبیه‌سازی مدار قبل از پیاده‌سازی سخت‌افزار واقعی استفاده شد. این امکان آزمایش و رفع اشکال طراحی مدار را به صورت مجازی فراهم کرد، شناسایی و حل مشکلات بدون خطر آسیب به اجزای فیزیکی. شبیه‌سازی شامل تأیید اتصالات، اطمینان از عملکرد صحیح نمایشگر **LCD** و آزمایش روتین لغزش برای **push button** بود.

با رفع این چالش‌ها از طریق طراحی و پیاده‌سازی دقیق، پروژه با موفقیت بازی **T-Rex** را با استفاده از **Arduino** شبیه‌سازی کرد و یک نمایش تعاملی و جذاب از قابلیت‌های **microcontroller** ارائه داد. این پروژه همچنین اهمیت ابزارهای شبیه‌سازی مانند **Proteus** را در فرآیند توسعه نشان داد که امکان آزمایش و رفع اشکال مؤثر را فراهم می‌کند.

