

دریافت مقاله: ۱۴۰۴/۰۷/۰۲

پذیرش مقاله: ۱۴۰۴/۰۸/۱۲

نوع مقاله: فنی

نحوه تعامل مخاطبان بازی‌های رایانه‌ای با بررسی دستگاه‌های ورودی و خروجی در یونیتی

امیرحسین اردلان*

دپارتمان فنی و مهندسی، استاد بازیسازی، کالج بین المللی ابن سینا، ایران، تهران و تغلیس، گرجستان.
پست الکترونیکی: shsbbhorse@chmail.ir

چکیده

تعامل کاربر با محیط بازی، هسته اصلی تجربه بازیکن را تشکیل می‌دهد و این تعامل از طریق دستگاه‌های ورودی و خروجی متنوعی صورت می‌گیرد. این مقاله به بررسی جامع نحوه تعامل مخاطب در بازی‌های رایانه‌ای با تمرکز بر عملکرد دستگاه‌های ورودی و خروجی و پیاده‌سازی آنها در موتور بازی‌سازی یونیتی (Unity) می‌پردازد. ابتدا، دستگاه‌های ورودی و خروجی رایج و حیاتی در بوم‌سازگان بازی‌های رایانه‌ای (مانند صفحه کلید، موش و صفحه نمایش) معرفی شده و اصول فنی کارکرد آنها تشریح می‌گردد. سپس، با بررسی کدهای نمونه و API‌های مرتبط در یونیتی، چگونگی دریافت، پردازش، و تفسیر این ورودی‌های فیزیکی و ترجمه آنها به عملیات درون بازی مورد تحلیل قرار می‌گیرد و شیوه ارتباط آنها با سخت‌افزار توضیح داده می‌شود. هدف این پژوهش، ایجاد پلی میان مبانی سخت‌افزاری دستگاه‌های تعاملی و پیاده‌سازی نرم‌افزاری آنها در یک موتور بازی‌سازی قدرتمند مانند یونیتی است. درک این چرخه کامل، از لحظه ورود فیزیکی تا ترجمه آن به عملیات درون بازی، برای توسعه‌دهندگان بازی و پژوهشگران این حوزه ضروری بوده و می‌تواند به

طراحی رابط‌های کاربری بهینه‌تر و خلق تجربه‌های بهتری در بازی‌ها منجر شود.
کلید واژه‌ها: بازی رایانه‌ای، بازی، یونیتی، دستگاه‌های ورودی، برنامه‌نویسی

۱- مقدمه

تعامل کاربر با محیط بازی‌های رایانه‌ای، یکی از ارکان اساسی تجربه کاربری است و نقش تعیین‌کننده‌ای در جذب و نگهداشت مخاطب ایفا می‌کند. این تعامل از طریق دستگاه‌های ورودی و خروجی متنوعی صورت می‌پردازد که هر یک با سازوکارها و فناوری‌های خاص خود، امکان ارتباط بین بازیکن و دنیای مجازی را فراهم می‌آورند. دستگاه‌های ورودی مانند صفحه کلید، موش، اهرم کنترل، کنترل‌کننده‌های حرکتی و دیگر دستگاه‌ها، ورودی‌های فیزیکی کاربر را دریافت و به سیگنال‌های دیجیتال تبدیل می‌کنند. در حالی که دستگاه‌های خروجی مانند صفحه نمایش، گوشی، بلندگو و دستگاه‌های بازخورد لمسی، خروجی‌های بازی را به کاربر منتقل می‌سازند. کیفیت و کارایی این تعامل، مستقیماً بر تجربه بازی تأثیر می‌گذارد و می‌تواند تفاوت میان یک بازی موفق و یک تجربه ناکام را مشخص کند.

* نویسنده مسئول

با بهره‌گیری از این دانش، رابط‌های کاربری بهینه‌تری طراحی کنند و تجربه‌های بازی جذاب‌تر و غوطه‌ورتری خلق نمایند.

می‌توان ساختار این مقاله را به چند بخش مهم و اساسی تقسیم نمود. ساختار این مقاله به این شرح است: بخش اول مقدمه و توضیحات پایه است. در بخش دوم، دستگاه‌های ورودی و خروجی رایج در بازی‌های رایانه‌ای معرفی و اصول فنی کارکرد آنها تشریح می‌شود. بخش سوم به بررسی پیاده‌سازی این دستگاه‌ها در موتور یونیتی و تحلیل کدهای نمونه اختصاص دارد. در بخش چهارم، نتایج و تحلیل‌های حاصل از پیاده‌سازی ارائه می‌شود. در پایان، نتیجه‌گیری و پیشنهادهایی برای پژوهش‌های آتی ارائه خواهد شد.

۲- چالش‌های اصلی در تعامل با مخاطبان

ارتباط برقرار کردن با بازی‌کنندگان بسیار اهمیت دارد. اگر مخاطبان نتوانند با یک بازی ارتباط برقرار کنند؛ آن بازی کنار گذاشته خواهد شد و هرگز به فروش و سود نخواهد رسید. باید دستگاه‌های ورودی و خروجی طوری تنظیم شوند که کاملاً قابل تعامل باشند. راحتی در ارتباط برقرار کردن یکی از همین موارد است. کاربر باید به راحتی با بازی ارتباط برقرار کند. نباید این کار بسیار پیچیده باشد یا نیاز به سخت‌افزارهای کمیابی داشته باشد و گرنه تعداد افرادی که می‌توانند با بازی تعامل کنند کمتر می‌شود. شخص باید خودش را قسمتی از بازی بداند. باید بتواند آن بخش از بازی را مدیریت کند و این وظیفه بخش تعامل است که دستگاه‌های ورودی و خروجی را شامل می‌شود. گاه ممکن است یک بازی چندنفره باشد. در این صورت چالش‌های دیگری نیز پیش روی طراحان است. در هر صورت، باید با سخت‌افزار و کدنویسی آنها آشنا شد تا مشکلات این امر کمتر شود. در ادامه به معرفی ابزارهایی برای تعامل بهتر اشاره می‌شود.

از آنجایی که این تعامل بر بستر موتور بازی‌سازی

موتور بازی‌سازی یونیتی، به عنوان یکی از پرکاربردترین ابزارهای توسعه بازی، امکانات گسترده‌ای برای مدیریت و پردازش ورودی‌ها و خروجی‌ها فراهم می‌آورد. این موتور با ارائه API‌های قدرتمند و انعطاف‌پذیر، امکان دریافت، تفسیر و ترجمه ورودی‌های کاربر به عملیات درون بازی را به صورت کارآمد فراهم می‌سازد. با این حال، پیاده‌سازی بهینه این تعاملات نیازمند درک دقیق از اصول فنی دستگاه‌های سخت‌افزاری و همچنین توانایی استفاده از کدها و نوشتارهای مناسب در یونیتی است. عدم آگاهی از سازوکارهای سخت‌افزاری و نرم‌افزاری این دستگاه‌ها می‌تواند منجر به طراحی رابط‌های کاربری ناکارآمد، کاهش عملکرد بازی، و در نهایت، تجربه کاربری نامطلوب شود.

پژوهش‌های پیشین در این حوزه بیشتر بر روی بررسی جداگانه دستگاه‌های ورودی و خروجی یا تحلیل عمومی موتورهای بازی‌سازی متمرکز بوده‌اند، اما کمتر به بررسی جامع و یکپارچه چرخه کامل تعامل کاربر از لحظه ورود فیزیکی تا ترجمه آن به عملیات درون بازی پرداخته شده است. این شکاف پژوهشی، توسعه‌دهندگان و پژوهشگران را با چالش‌هایی در طراحی و پیاده‌سازی رابط‌های کاربری بهینه و تجربه‌های بازی غوطه‌ور مواجه کرده است.

در این پژوهش، ابتدا به معرفی دستگاه‌های ورودی و خروجی رایج در بوم‌سازگان بازی‌های رایانه‌ای پرداخته می‌شود و اصول فنی کارکرد هر یک تشریح می‌گردد. سپس، با بررسی کدهای نمونه و API‌های مرتبط در یونیتی، چگونگی دریافت، پردازش و تفسیر این ورودی‌ها و ترجمه آنها به عملیات درون بازی تحلیل می‌شود. علاوه بر این، کاربرد هر یک از این دستگاه‌ها در بازی‌های مختلف و نحوه پیاده‌سازی آنها در پروژه‌های عملی مورد بررسی قرار می‌گیرد. هدف این مقاله، ایجاد پلی میان مبانی سخت‌افزاری دستگاه‌های تعاملی و پیاده‌سازی نرم‌افزاری آنها در یونیتی است؛ به گونه‌ای که توسعه‌دهندگان بتوانند

می‌شود. تفاوت بین دو نوع کلیدهای مکانیکی و غشایی در نوع ساخت و استفاده از آن است.

کلیدهای مکانیکی: هر کلید مکانیکی شامل یک سوئیچ فیزیکی است که با فشار دادن، یک مدار الکتریکی را کامل می‌کند. این نوع کلیدها به دلیل بازخورد لمسی و صدای کلیک، برای کاربران حرفه‌ای محبوب هستند.

کلیدهای غشایی: در این نوع صفحه‌کلید، کلیدها بر روی یک لایه غشایی قرار دارند و با فشار دادن، دو لایه غشایی به یکدیگر متصل می‌شوند و سیگنال الکتریکی تولید می‌شود. این نوع صفحه‌کلیدها معمولاً ارزان‌تر و بی‌صداتر هستند اما معمولاً دقت و طول عمر کمتری نسبت به کلیدهای مکانیکی دارند.

در بازی‌ها، صفحه‌کلید معمولاً برای کنترل حرکت شخصیت، اجرای دستورات سریع (مانند پرش، شلیک، یا استفاده از آیتم‌ها)، و ورود متن در محیط‌های چندمنوره یا گپ‌زنی استفاده می‌شود. دقت داشته باشید که مفصل‌ترین بحث را بر روی صفحه‌کلید خواهیم داشت.

۱.۱.۴. سازوکار درونی صفحه‌کلید

صفحه‌کلیدها از مدارهای الکتریکی استفاده می‌کنند که با فشار دادن هر کلید، جریان الکتریکی قطع یا وصل می‌شود و سیگنال مربوطه به سیستم ارسال می‌شود. در بازی‌ها، این سیگنال‌ها توسط موتور بازی‌سازی (مانند یونیتی) دریافت و به عملیات درون بازی ترجمه می‌شوند. کلیدهای صفحه‌کلید به صورت ماتریسی (ردیفی و ستونی) سازماندهی شده‌اند. این سازماندهی را می‌توانید در شکل‌های ۱ و ۲ مشاهده فرمایید. مدار الکتریکی این ماتریس به طور معمول با ولتاژ پایینی فعال است و دائماً توسط میکروکنترلر داخلی صفحه‌کلید پویش می‌شود. هنگامی که یک کلید فشرده می‌شود، اتصال الکتریکی در محل تقاطع ردیف و ستون مربوطه برقرار می‌گردد. این اتصال، تغییر مشخصی در سیگنال الکتریکی (ولتاژ) ایجاد می‌کند که توسط میکروکنترلر شناسایی و تشخیص

یونیتی باید پیاده‌سازی شود؛ باید میان یونیتی و دستگاه‌های ورودی و خروجی ارتباط برقرار کرد. از این جهت باید این ارتباط را بهتر درک نمود و نحوه کار با آنها را فرا گرفت. یکی از اهداف مقاله حاضر همین مورد است

۳- دستگاه‌های ورودی و خروجی در بازی‌های رایانه‌ای

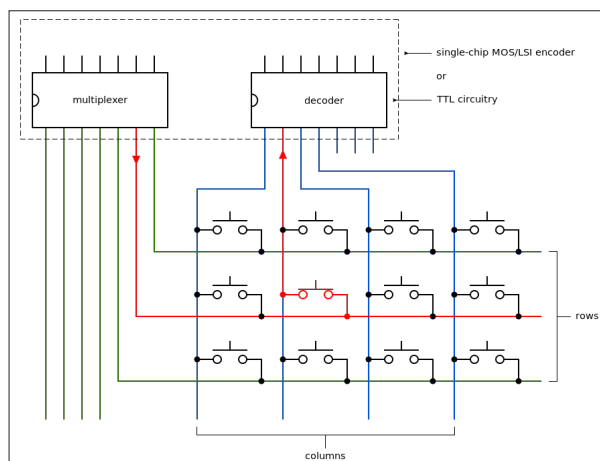
در بازی‌های رایانه‌ای، دستگاه‌های ورودی و خروجی نقش اساسی در ایجاد تعامل بین کاربر و محیط بازی ایفا می‌کنند. این دستگاه‌ها به عنوان واسطی بین دنیای فیزیکی و دنیای مجازی عمل می‌کنند و کیفیت تجربه بازی را به‌طور مستقیم تحت تاثیر قرار می‌دهند. به صورت کلی، دستگاه‌های ورودی، اطلاعاتی را وارد کامپیوتر می‌کنند مانند موش و صفحه‌کلید و دستگاه‌های خروجی، اطلاعاتی را از کامپیوتر دریافت می‌کنند. در این بخش، به معرفی جامع و فنی دستگاه‌های ورودی و خروجی رایج در بوم‌سازگان بازی‌های رایانه‌ای پرداخته می‌شود. البته با تاکید ویژه بر صفحه‌کلید، موش، و دسته بازی که این ابزارها پرکاربردترین ابزارهای تعامل در بازی‌های مدرن هستند [۲].

۴- دستگاه‌های ورودی

دستگاه‌های ورودی وظیفه دریافت دستورها و اقدامات کاربر را بر عهده دارند و این اطلاعات را به صورت سیگنال‌های دیجیتال به سیستم بازی منتقل می‌کنند. دقت، سرعت پاسخگویی و نوع ورودی ارائه‌شده توسط این دستگاه‌ها، تاثیر مستقیمی بر تجربه کاربر دارد.

۴-۱- صفحه‌کلید

صفحه‌کلید یکی از اساسی‌ترین و پرکاربردترین دستگاه‌های ورودی در بازی‌های رایانه‌ای است. این دستگاه از مجموعه‌ای از کلیدهای مکانیکی یا غشایی تشکیل شده که با فشار دادن هر کلید، یک سیگنال الکتریکی تولید



شکل (۲): تصویر نمایشی از سیم‌های اتصال داخل صفحه کلید، برای درک صحیح مبحث ماتریس و تشخیص دکمه فشرده شده.

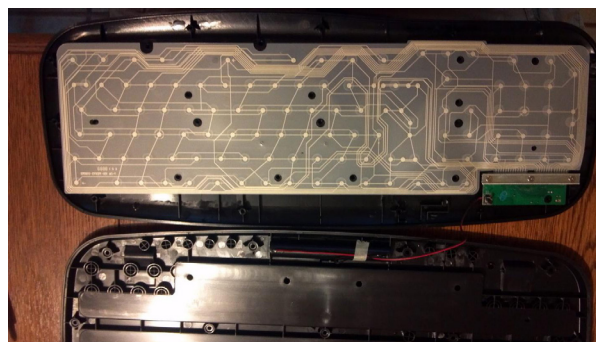
با کامپیوتر استفاده می‌شود. برای ارسال سیگنال استفاده می‌کنند. میکروکنترلر صفحه کلید، کد پویش مربوط به کلید فشرده شده را تولید و به سیستم عامل ارسال می‌کند. البته دقت داشته باشید که این نکات برای زبان‌های مختلف و در استانداردهای مختلف، ممکن است کمی متفاوت باشد.

مثال برای فشردن کلید A یا همان کلید ش:

- صفحه کلید کد پویش داخلی خود را تولید می‌کند. (مثلاً 0x04 یا هر کد دیگری که سازنده صفحه کلید تعریف کرده است. این دست کدها کاملاً متغیر و مرتبط به سازنده صفحه کلید هستند. پس در جدول ذکر نشده است.)
- صفحه کلید، کد پویش داخلی را به USB HID Usage ID استاندارد تبدیل می‌کند. برای کلید A، این عدد ۴ است.
- صفحه کلید، USB HID Usage ID (۴) را از طریق پروتکل USB به کامپیوتر ارسال می‌کند.
- درایور صفحه کلید، USB HID Usage ID (۴) را دریافت می‌کند. گرداننده، این عدد را به Virtual Key 0x41 یا VK_A تبدیل می‌کند.
- سیستم عامل، Virtual Key (0x41) را به برنامه‌ها (مثل یونیتی) می‌دهد. برنامه‌ها از این کد برای تشخیص کلید A استفاده می‌کنند.

۴-۱-۳ نقشه کلیدکدها در نرم افزار یونیتی

داخل نرم افزار یونیتی نیز کار با صفحه کلید بسیار



شکل (۱): محتویات داخل صفحه کلید. کلیدها به صورت ماتریسی متصل هستند.

داده می‌شود. سپس، اطلاعات مربوط به کلید فشرده شده برای پردازش به کامپیوتر ارسال می‌گردد. در واقع، هر صفحه کلید حاوی یک یا چند میکروکنترلر است که مسئولیت اصلی پایش، تشخیص و ارسال این سیگنال‌ها را بر عهده دارند [۳].

۴-۱-۲ نقشه کد کلیدها در صفحه کلید

صفحه کلید مجموعه‌ای از کلیدها است که هر یک برای ثبت و پردازش ورودی‌ها، یک «کد پویش» منحصر به فرد دریافت می‌کنند. این کدها، که معمولاً به صورت اعداد در مبنای شانزده نمایش داده می‌شوند؛ توسط سخت افزار صفحه کلید تولید و در لایه‌های مختلف سیستم، از جمله گرداننده صفحه کلید و حافظه (RAM)، پردازش می‌شوند. گرداننده صفحه کلید، کد پویش را به رویدادهای قابل فهم برای سیستم عامل (مثل KeyDown یا KeyUp) تبدیل می‌کند. علاوه بر این، نقشه کدکلید هر کلید را به یک عدد یا نویسه خاص نگاشت می‌کند تا ترتیب و عملکرد آن‌ها مشخص شود [۴].

به عنوان مثال، صفحه کلیدهای استاندارد معمولاً دارای ۱۰۴ یا ۱۰۵ کلید هستند و به هر کلید، اعم از حروف، اعداد و کلیدهای عملکردی، یک کد پویش اختصاص می‌یابد. نقشه کد کلیدهای اولیه صفحه کلید در جدول ۱ آورده شده است.

صفحه کلیدها از پروتکل‌های استاندارد مانند USB HID که در جدول ۱ آمده؛ پیروی می‌کنند. این یک استاندارد است که توسط دستگاه‌های ورودی برای ارتباط

جدول (۱): جدول اسکن کدهای ورودی صفحه‌کلید طبق استاندارد

Virtual Key (Hex)	USB HID (dec)	نام کلید	Virtual Key (Hex)	USB HID (dec)	نام کلید	Virtual Key (Hex)	USB HID (dec)	نام کلید
0x43 (VK_C)	۶	C	0x42 (VK_B)	۵	B	0x41 (VK_A)	۴	A
0x46 (VK_F)	۹	F	0x45 (VK_E)	۸	E	0x44 (VK_D)	۷	D
0x49 (VK_I)	۱۲	I	0x48 (VK_H)	۱۱	H	0x47 (VK_G)	۱۰	G
0x4C (VK_L)	۱۵	L	0x4B (VK_K)	۱۴	K	0x4A (VK_J)	۱۳	J
0x4F (VK_O)	۱۸	O	0x4E (VK_N)	۱۷	N	0x4D (VK_M)	۱۶	M
0x52 (VK_R)	۲۱	R	0x51 (VK_Q)	۲۰	Q	0x50 (VK_P)	۱۹	P
0x55 (VK_U)	۲۴	U	0x54 (VK_T)	۲۳	T	0x53 (VK_S)	۲۲	S
0x58 (VK_X)	۲۷	X	0x57 (VK_W)	۲۶	W	0x56 (VK_V)	۲۵	V
0x31 (VK_1)	۳۰	1	0x5A (VK_Z)	۲۹	Z	0x59 (VK_Y)	۲۸	Y
0x34 (VK_4)	۳۳	4	0x33 (VK_3)	۳۲	3	0x32 (VK_2)	۳۱	2
0x37 (VK_7)	۳۶	7	0x36 (VK_6)	۳۵	6	0x35 (VK_5)	۳۴	5
0x30 (VK_0)	۳۹	0	0x39 (VK_9)	۳۸	9	0x38 (VK_8)	۳۷	8
0x08 (VK_BACK)	۴۲	Back-Space	0x1B (VK_ESCAPE)	۴۱	esc	0x0D (VK_RETURN)	۴۰	Enter
0xBD (VK_OEM_MINUS)	۴۵	- Minus	0x20 (VK_SPACE)	۴۴	Space	0x09 (VK_TAB)	۴۳	Tab
0xDD (VK_OEM_6)	۴۸	Bracket [Right	0xDB (VK_OEM_4)	۴۷	Bracket] Left	0xBB (VK_OEM_PLUS)	۴۶	Equals =
0xDE (VK_OEM_7)	۵۲	' Quote	0xBA (VK_OEM_1)	۵۱	Semico-; lon	0xDC (VK_OEM_5)	۴۹	Backslash\
0xBE (VK_OEM_PERIOD)	۵۵	. Period	0xBC (VK_OEM_COMMA)	۵۴	, Comma	0xC0 (VK_OEM_3)	۵۳	` Grave
0x70 (VK_F1)	۵۸	F1	0x14 (VK_CAPITAL)	۵۷	Caps Lock	0xBF (VK_OEM_2)	۵۶	/ Slash
0x73 (VK_F4)	۶۱	F4	0x72 (VK_F3)	۶۰	F3	0x71 (VK_F2)	۵۹	F2
0x76 (VK_F7)	۶۴	F7	0x75 (VK_F6)	۶۳	F6	0x74 (VK_F5)	۶۲	F5
0x79 (VK_F10)	۶۷	F10	0x78 (VK_F9)	۶۶	F9	0x77 (VK_F8)	۶۵	F8
0x2C (VK_SNAPSHOT)	۷۰	Print Screen	0x7B (VK_F12)	۶۹	F12	0x7A (VK_F11)	۶۸	F11
0x2D (VK_INSERT)	۷۳	Insert	0x13 (VK_PAUSE)	۷۲	Pause	0x91 (VK_SCROLL)	۷۱	Scroll Lock
0x2E (VK_DELETE)	۷۶	Delete	0x21 (VK_PRIOR)	۷۵	Page Up	0x24 (VK_HOME)	۷۴	Home

0x27 (VK_RIGHT)	۷۹	Right Arrow	0x22 (VK_NEXT)	۷۸	Page Down	0x23 (VK_END)	۷۷	End
0x26 (VK_UP)	۸۲	Up Arrow	0x28 (VK_DOWN)	۸۱	Down Arrow	0x25 (VK_LEFT)	۸۰	Left Arrow
0x6A (VK_MULTIPLY)	۸۵	* Keypad	0x6F (VK_DIVIDE)	۸۴	/ Keypad	0x90 (VK_NUMLOCK)	۸۳	Num Lock
0x0D (VK_RETURN)	۸۸	Keypad Enter	0x6B (VK_ADD)	۸۷	+ Keypad	0x6D (VK_SUBTRACT)	۸۶	Keypad -
0x63 (VK_NUMPAD3)	۹۱	Keypad 3	0x62 (VK_NUMPAD2)	۹۰	Keypad 2	0x61 (VK_NUMPAD1)	۸۹	Keypad 1
0x66 (VK_NUMPAD6)	۹۴	Keypad 6	0x65 (VK_NUMPAD5)	۹۳	Keypad 5	0x64 (VK_NUMPAD4)	۹۲	Keypad 4
0x69 (VK_NUMPAD9)	۹۷	Keypad 9	0x68 (VK_NUMPAD8)	۹۶	Keypad 8	0x67 (VK_NUMPAD7)	۹۵	Keypad 7
0xA2 (VK_LCONTROL)	۲۲۴	Left Ctrl	0x6E (VK_DECIMAL)	۹۹	. Keypad	0x60 (VK_NUMPAD0)	۹۸	Keypad 0
0x5B (VK_LWIN)	۲۲۷	Left GUI ((Win	0xA4 (VK_LMENU)	۲۲۶	Left Alt	0xA0 (VK_LSHIFT)	۲۲۵	Left Shift
0xA5 (VK_RMENUE)	۲۳۰	Right Alt	0xA1 (VK_RSHIFT)	۲۲۹	Right Shift	0xA3 (VK_RCONTROL)	۲۲۸	Right Ctrl
			0x5D (VK_APPS)	۱۱۸	Menu (App Key)	0x5C (VK_RWIN)	۲۳۱	Right GUI (Win)

سی‌شارپ یونیتی نشان می‌دهد، که توسعه‌دهندگان با استفاده از آنها می‌توانند ورودی‌ها را به صورت کارآمد مدیریت کنند.

چند نکته در خصوص جدول ۲ وجود دارد که در ادامه ذکر می‌کنیم:

- اعداد نوشته شده برای هر کلید، در یونیتی کاربرد دارد. صرفاً برای شمارش نیست.

- برخی از کلیدها چیزی را تایپ نمی‌کنند. مانند شیف‌ت که باید با کلید دیگری هم‌زمان فشرده شود تا کاری انجام دهد. برخی از ترکیب‌های کلیدها با شیف‌ت در جدول ۲ ذکر شده است.

- کلیدهای سری f^1-f^{12} معمولاً در اکثر صفحه‌کلیدها هستند اما در بازی‌ها، کمتر استفاده می‌شوند. این کلیدها هر کدام به چیزی معروف هستند. ممکن است در برخی بازی‌ها،

حیاتی است. البته کدها و عبارات، مانند جدول ۱ وارد یونیتی نمی‌شوند؛ بلکه اطلاعات ابتدا از صفحه‌کلید وارد کامپیوتر شده، توسط گرداننده صفحه‌کلید و پروتکل USB HID پردازش می‌شوند و سپس از طریق خط لوله مدیریت ورودی چهارچوب دات‌نت به یونیتی منتقل می‌گردند. علاوه‌بر گرداننده صفحه‌کلید و حافظه اصلی، دات‌نت نقش کلیدی در تفسیر و انتقال ورودی‌ها ایفا می‌کند. در زمان نگارش این مقاله، یونیتی برای تشخیص و پردازش ورودی‌ها به نسخه ۴.۸ یا بالاتر چهارچوب دات‌نت نیاز دارد. با این حال، همگام‌سازی ورودی‌های صفحه‌کلید از طریق USB HID با خط لوله دات‌نت ممکن است با چالش‌هایی مانند تاخیر، تداخل در پردازش هم‌زمان ورودی‌ها، و عدم سازگاری برخی گرداننده مواجه شود. جدول ۲، نقشه اعداد و کدهای ورودی صفحه‌کلید را در برنامه‌نویسی

جدول (۲): اعداد و کدهای ورودی صفحه‌کلید در برنامه‌نویسی سی‌شارپ یونیتی (استاندارد وقت حاضر)

محتوا	عدد	محتوا	عدد	محتوا	عدد
Enter	۲	Space	۱	تعریف نشده (None)	۰
quote : “ ” + Shift = “	۵	Backquote : “ ` ” + Shift = ~	۴	Tab	۳
Period : “ . ” + Shift = >	۸	Comma : “ , ” + Shift = <	۷	Semicolon : “ ; ” + Shift = :	۶
LeftBracket : “ [” + Shift = {	۱۱	Backslash : “ \ ” + Shift =	۱۰	Slash : “ / ” + Shift = ?	۹
Equals : “ = ” + Shift = +	۱۴	Minus : “ - ” + Shift = _	۱۳	RightBracket : “] ” + Shift = }	۱۲
C : “ c ” + Shift = C	۱۷	B : “ b ” + Shift = B	۱۶	A : “ a ” + Shift = A	۱۵
F : “ f ” + Shift = F	۲۰	E : “ e ” + Shift = E	۱۹	D : “ d ” + Shift = D	۱۸
I : “ i ” + Shift = I	۲۳	H : “ h ” + Shift = H	۲۲	G : “ g ” + Shift = G	۲۱
L : “ l ” + Shift = L	۲۶	K : “ k ” + Shift = K	۲۵	J : “ j ” + Shift = J	۲۴
O : “ o ” + Shift = O	۲۹	N : “ n ” + Shift = N	۲۸	M : “ m ” + Shift = M	۲۷
R : “ r ” + Shift = R	۳۲	Q : “ q ” + Shift = Q	۳۱	P : “ p ” + Shift = P	۳۰
U : “ u ” + Shift = U	۳۵	T : “ t ” + Shift = T	۳۴	S : “ s ” + Shift = S	۳۳
X : “ x ” + Shift = X	۳۸	W : “ w ” + Shift = W	۳۷	V : “ v ” + Shift = V	۳۶
Digit1 : “ 1 ” + Shift = !	۴۱	Z : “ z ” + Shift = Z	۴۰	Y : “ y ” + Shift = Y	۳۹
Digit4 : “ 4 ” + Shift = \$	۴۴	Digit3 : “ 3 ” + Shift = #	۴۳	Digit2 : “ 2 ” + Shift = @	۴۲
Digit7 : “ 7 ” + Shift = &	۴۷	Digit6 : “ 6 ” + Shift = ^	۴۶	Digit5 : “ 5 ” + Shift = %	۴۵
Digit0 : “ 0 ” + Shift = (	۵۰	Digit9 : “ 9 ” + Shift =)	۴۹	Digit8 : “ 8 ” + Shift = *	۴۸
LeftAlt	۵۳	RightShift	۵۲	LeftShift	۵۱
RightCtrl	۵۶	LeftCtrl	۵۵	RightAlt	۵۴
ContextMenu	۵۹	RightWindows	۵۸	LeftWindows	۵۷
RightArrow	۶۲	LeftArrow	۶۱	Escape	۶۰
Backspace	۶۵	DownArrow	۶۴	UpArrow	۶۳
Home	۶۸	PageUp	۶۷	PageDown	۶۶
Delete	۷۱	Insert	۷۰	End	۶۹
PrintScreen	۷۴	NumLock	۷۳	CapsLock	۷۲
NumpadEnter	۷۷	Pause(Break)	۷۶	ScrollLock	۷۵
NumpadPlus : “ + ”	۸۰	NumpadMultiply	۷۹	NumpadDivide : “ / ”	۷۸
NumpadEquals : “ = ”	۸۳	NumpadPeriod : “ . ”	۸۲	NumpadMinus : “ - ”	۸۱
Numpad2 : “ 2 ”	۸۶	Numpad1 : “ 1 ”	۸۵	Numpad0 : “ 0 ”	۸۴
Numpad5 : “ 5 ”	۸۹	Numpad4 : “ 4 ”	۸۸	Numpad3 : “ 3 ”	۸۷
Numpad8 : “ 8 ”	۹۲	Numpad7 : “ 7 ”	۹۱	Numpad6 : “ 6 ”	۹۰
F2	۹۵	F1	۹۴	Numpad9 : “ 9 ”	۹۳
F5	۹۸	F4	۹۷	F3	۹۶
F8	۱۰۱	F7	۱۰۰	F6	۹۹
F1	۱۰۴	F10	۱۰۳	F9	۱۰۲
OEM2	۱۰۷	OEM1	۱۰۶	F12	۱۰۵
OEM5	۱۱۰	OEM4	۱۰۹	OEM3	۱۰۸
F14	۱۱۳	F13	۱۱۲	تعریف نشده (None)	۱۱۱
F17	۱۱۶	F16	۱۱۵	F15	۱۱۴
F20	۱۱۹	F19	۱۱۸	F18	۱۱۷
F23	۱۲۲	F22	۱۲۱	F21	۱۲۰
				F24	۱۲۳

برای کدهای قدیمی است. الگوی این کدها بدین صورت بود که ابتدا کلمه ورودی Input نوشته می‌شد و بعد از آن دیگر موارد ذکر می‌شدند. در ورودی‌های جدید یونیتی، ابتدا اسم دستگاه ورودی نوشته می‌شود و سپس مابقی آن تکمیل می‌گردد. در هر دو مورد، حرف A نوشته شده است. یعنی برای تشخیص ورودی این حرف استفاده می‌شود. از زمانی که این کلید فشرده شود تا زمانی که رها شود؛ این کد فعال خواهد بود و کار خواهد کرد. البته می‌توان آنها را به صورت کد شرطی نیز نوشت. یعنی هر زمان که این کلید فشرده شده است؛ کاری را انجام دهد. می‌توان عددی نیز کار کرد. مثلاً یک متغیر int داشته باشیم. این متغیر، اعداد صحیح را در خود ذخیره می‌کند. اگر آن را در تابع بروزرسان Update برابر با کد ورودی قرار دهیم؛ عددی بین ۰ یا ۱ خواهد بود که عدد ۰ یعنی غیرفعال و عدد ۱ یعنی فعال است. و متغیر bool نیز وجود دارد که فعال/ غیرفعال بودن شرایط را می‌سنجد. در اینجا، فشرده شدن کلید A را بررسی می‌کند.

۱ $\xrightarrow{old}$ input.GetKey(KeyCode.A)
۲ $\xrightarrow{new}$ Keyboard.current.aKey.ispressed

رابطه (۱): مثال کدهای ورودی جدید و قدیمی
رابطه ۲ طبق کد ورودی جدید نوشته شده است. برای این که آن را تبدیل به کدهای ورودی قدیمی کنید؛ کافی است از رابطه ۱ استفاده کنید و به جای جملات جدید که استفاده شده‌اند؛ جمله اول در رابطه ۱ را قرار دهید. تحلیل رابطه ۲ چنین است که متغیری عددی از جنس عدد صحیح داریم. اسم آن input0 است. یک متغیر صحیح/ غلط به نام inputa هم داریم. زمانی که بازی شروع شود و مخاطب بر روی حرف A کلیک کند؛ چند اتفاق می‌افتد. ابتدا متغیر عددی ما یعنی input0 برابر با ۱ می‌شود. و متغیر صحیح/ غلط یعنی inputa برابر با مقدار صحیح یا همان true خواهد شد. سپس عبارت "a!!pressed" چاپ می‌شود. این چاپ شدن به صورت محلی و در پنجره console خواهد بود [۱]

جدول (۳): معرفی کلیدهای f¹-f¹² به همراه کاربردها

F1	باز کردن منوی راهنما
F2	باز کردن صفحه اقدامات/عناوین
F3	باز کردن صفحه اطلاعات و گزارشات
F4	تکرار مورد فعلی
F5	خواندن یا ادامه دادن به صورت پیوسته
F6	دیکته صوتی
F7	کاهش حجم صدا
F8	افزایش حجم صدا
F9	کاهش سرعت صدا
F10	افزایش سرعت صدا
F11	تغییر حالت تمام صفحه
F12	باز کردن ابزار توسعه‌دهنده مورد نیاز

به همان چیزی که در جدول ۳ معروف‌اند؛ استفاده شوند.
● سازنده تجهیزات اصلی: OEM کلیدهایی هستند که در هر کامپیوتر یا لپ‌تاپی وجود ندارند. این کلیدهای خاص و کمیابی هستند که در برخی صفحه‌کلیدها و سیستم‌ها وجود دارند. ممکن است ترکیب دو کلید مختلف باهم باشد.
● از کلیدهای f¹³-f²⁴ معمولاً استفاده‌ای نمی‌شود. زیرا در صفحه‌کلیدهای معمولی قرار ندارد. همچنین در یونیتی نیز نمی‌توان به صورت مستقیم از این کلیدها استفاده نمود.
● در برخی موارد عدد ۱۱۱ منظور همان f¹³ است.

۴-۱-۴ برنامه‌نویسی برای تشخیص ورودی‌های صفحه‌کلید

در برنامه‌نویسی یونیتی که با سی‌شارپ انجام می‌شود؛ انواع روش‌ها برای استفاده از ورودی‌ها فراهم است. شرکت یونیتی در سال ۲۰۲۵ با انتشار نسخه رسمی یونیتی ۶ اعلام کرد که کدهای جدیدی برای ورودی در نظر گرفته است. البته کدهای قدیمی [۲] نیز همچنان وجود دارند و کاملاً منسوخ نشده‌اند. البته بدون داشتن اعداد و اسم جدول‌های ۱ و ۲ نیز می‌توان ورودی‌ها را برنامه‌نویسی کرد زیرا همان‌طور که گفته شد؛ روش‌های گوناگونی برای برنامه‌نویسی آنها وجود دارد.
کدهای ورودی طبق رابطه ۱ کار می‌شوند. خط اول

عناوینی معروف شدند. در جدول ۴ برخی از این کلیدهای معروف و متداول، معرفی شده‌اند. برنامه‌نویسان و بازی‌سازان، همچنان از همین کلیدهای صفحه‌کلید استفاده می‌کنند [۱].

دیگر دستگاه‌های ورودی نیز شبیه به صفحه‌کلید هستند و کدهای مختلفی دارند اما مبحث صفحه‌کلید را بسیار کامل و جامع‌تر مورد بررسی قرار دادیم.

۴-۲. موش

موش مانند صفحه‌کلید، ابزاری ورودی است که نقش اساسی در تعامل با سیستم‌های رایانه‌ای ایفا می‌کند. این دستگاه با وجود تعداد کلیدهای کمتر نسبت به صفحه‌کلید، عمدتاً مسئولیت کنترل اشاره‌گر (Pointer) و انتخاب گزینه‌ها را بر عهده دارد. موش با تشخیص حرکت در دو محور افقی (X) و عمودی (Y) امکان جابه‌جایی دقیق اشاره‌گر را در محیط گرافیکی سیستم‌عامل فراهم می‌کند و کاربر از این طریق می‌تواند با عناصر بصری مختلف تعامل نماید [۵]. در ادامه این مقاله، به بررسی کاربردها و نقش کلیدهای موش در حوزه بازی‌سازی خواهیم پرداخت.

۴-۲-۱ تاریخچه و انواع موش

موش‌ها در ابتدا بسیار ساده بودند. همان‌طور که در شکل ۳ مشاهده می‌کنید؛ ابتدا موش‌ها چوبی بودند. یک دکمه هم بیشتر نداشتند که برای انتخاب کردن؛ مورد استفاده قرار می‌گرفت. در قسمت پایینی و کف این موش‌ها، غلطکی وجود داشت که حرکت در محور افقی و عمودی را تشخیص می‌داد. چندین سال بعد، حسگر حرکتی همراه با یک منبع نور کوچک در موش‌ها قرار گرفت تا بتواند میزان جابه‌جایی را تشخیص دهد و آن غلطک‌های قدیمی، برای همیشه حذف شدند. بعدها، موش‌ها خوش‌دست‌تر شدند و از پلاستیک ساخته می‌شدند. همچنین موش‌های بی‌سیم ساخته شدند تا کار با آنها راحت‌تر باشد و مشکلاتی مثل قطعی سیم وجود نداشته باشد.

```
using UnityEngine;
using UnityEngine.InputSystem;
public class NameCodeMono : MonoBehaviour
{
    public Key Key0;
    public int input0;
    public bool inputa;
    void Start()
    {
        Key0 = (Key)15;
        print(Key0);
    }
    void Update()
    {
        input0 = System.Convert.ToInt32(Keyboard.
current[Key0].isPressed);
        inputa = Keyboard.current[Key0].isPressed;
        if (Keyboard.current[Key0].isPressed)
        {
            print(Pressed a!!);
        }
    }
}
```

رابطه (۲): تشخیص دکمه فشرده شده و چاپ آن در پنجره کنسول

در رابطه ۲ به یک کد اشاره می‌شود که طبق جدول ۲ می‌توان آن را تفسیر کرد. در اواسط کد از عدد ۱۵ استفاده شده است. این عدد، دقیقاً طبق همان نقشه‌های عددی صفحه‌کلید است که در جدول ۲ ذکر کرده‌اند. یعنی منظور از این عدد، همان ورودی حرف A است. همچنین یعنی می‌توان اعداد دیگر را نیز جایگذاری کرد تا رابطه‌های عددی مختلفی به دست بیاید. به این صورت، می‌توان انواع ورودی‌ها را دید و نقشه آنها را مورد بررسی قرار داد.

۴-۱-۵ کلیدهای پر استفاده و معمول در بازی‌ها

در سال‌های گذشته یعنی حدوداً از سال ۱۳۷۰ شمسی برابر با ۱۹۹۰ میلادی، بازی‌های رایانه‌ای گسترش ویژه‌ای پیدا کردند. از همان زمان، صفحه‌کلیدهای قدیمی و جدید در انواع کامپیوترها و حتی دستگاه‌های بازی دیگر مانند سگا sega عرضه شد. همچنین برخی از کلیدهای صفحه‌کلید به

جدول (۴): کلیدهای پرستفاده و معمول در بازی‌های رایانه‌ای

نام کلید	نام لاتین (و مورد استفاده در برنامه‌نویسی)	استفاده معمول	شکل معمول کلید
جهت بالا	upArrowKey	برای حرکت به سمت بالا یا جلو، انتخاب گزینه بالاتر در منوها	
جهت پایین	downArrowKey	برای حرکت به سمت پایین یا عقب، انتخاب گزینه پایین‌تر در منوها	
جهت راست	rightArrowKey	برای حرکت به سمت راست، انتخاب گزینه راست در منوها	
جهت چپ	leftArrowKey	برای حرکت به سمت چپ، انتخاب گزینه چپ در منوها	
ص، w	W, wKey	حرکت به سمت بالا (بدل جهت بالا)	
س، s	S, sKey	حرکت به سمت پایین (بدل جهت پایین)	
ی، d	D, dKey	حرکت به سمت راست (بدل جهت راست)	
ش، a	A, aKey	حرکت به سمت چپ (بدل جهت چپ)	
فاصله	Space, spaceKey	برای پرش و پریدن، گاه‌ها افزایش انرژی یا شلیک	
تأیید	Enter, Return, enterKey	تأیید انتخاب در منوها، ارسال چت و تأیید انجام یک کار	
ظ، z	Z, zKey	حمله اصلی، ضربات متوالی (چندبار پشت‌سرهم فشرده شود).	
ط، x	X, xKey	حمله ثانویه، نوعی دیگر حمله. گاه‌ها با حمله اصلی ترکیب می‌شود.	
ق، r	R, rKey	برای چرخش قطعات، زیرا r ابتدای کلمه Rotate (چرخش) است. و برای پرکردن خراب تفنگ یا همان Reload در بازی‌های اکشن	
Esc	Escape, escapeKey	برای متوقف کردن بازی یا بازشدن منوی stop	
Tab, F1	tabKey, f1Key	برای بازکردن نقشه بازی، دسترسی سریع یا راهنمای بازی	
Ctrl	leftCtrlKey, rightCtrlKey	حمله (گاه‌ها بدل حرف Z)، فعال‌سازی قابلیت خاص مثل افزایش سرعت دوییدن یا غلط زدن یا استفاده از یک آیتم خاص	
Shift	leftShiftKey, rightShiftKey	دویدن، غلط زدن، آیتم‌ها، دقت کنید که این کلید نباید چندبار پشت سرهم فشرده شود زیرا پنجره sticky key را باز می‌کند.	



شکل (۴): یک نمونه موش ارگونومی



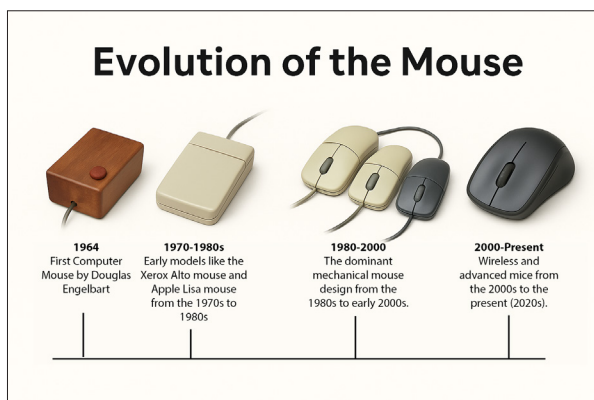
شکل (۵): یک تصویر از محیط داخلی بازی the colonist که با یونیتی ساخته شده و موش در آن، مانند استفاده معمول، برای انتخاب کردن استفاده می‌شود.



شکل (۶): یک تصویر از محیط داخلی بازی Delta Force ساخته کشور چین.

کاربرد اول، مرتبط به خود موش است. یعنی جابه‌جایی اشاره‌گر و انتخاب کردن که از ویژگی‌های اصلی و پایه موش است. به عنوان مثال در شکل ۵ انتخاب و جابه‌جایی توسط موش را شاهد هستیم. کاربرد دوم، اشاره به موارد معمول در بازی‌ها دارد. به عنوان مثال در بازی‌های جنگی، حرکت دو محور افقی و عمودی، باعث حرکت و چرخش تفنگ می‌شود. با چپ‌کلیک شلیک می‌کند و تیر می‌زند. با راست‌کلیک، خشاب را پر می‌کند. در شکل ۶ به یکی از این بازی‌ها اشاره شده است.

در برخی بازی‌ها، شکل اشاره‌گر مکان‌نما تغییر پیدا



شکل (۳): تصویر انواع موش‌ها در سال‌های مختلف ساخت

رفته‌رفته، موش‌های مختلفی ساخته شد تا کار کردن با آنها آسان‌تر باشد و برای افرادی که مدت زیادی مشغول فعالیت با موش هستند؛ مشکلات جسمانی ایجاد نکند یا این مشکلات به حداقل برسد. این موش‌ها، امروزه به «موش‌های ارگونومی» یا «موش‌های طبی» معروف هستند. یک نمونه موش طبی در شکل ۴ آورده شده است. این موش‌ها، معمولاً حدود ۱۰ تا ۳۰ درجه چرخش در محور اصلی دارند. اگر برای دست راست باشند؛ به سمت راست و اگر برای دست چپ باشند؛ به سمت چپ چرخش دارند. همچنین انگشت شست و دیگر قسمت‌های دست دیگر روی زمین نیست و با حرکت دادن موش، باعث خراشیدگی نمی‌شود. این امور باعث می‌شود که فشار کمتری به انگشتان و مچ وارد شود و بدین‌صورت از آسیب‌های احتمالی جلوگیری به عمل می‌آورد. موش‌های سنگین و قدیمی، باعث ایجاد درد در مچ و انگشتان و کبودی بخش زیر انگشت شست می‌شدند. امروزه با تجویز موش‌های طبی، این مشکلات رفع و به دشت کاهش یافته است. در بازی‌ها نیز همین امر وجود دارد. مخصوصاً بازی‌های پرتعداد که مخاطبان زمان زیادی را در بازی سپری می‌کنند. طراحی یک موش خوب باعث می‌شود تا از آسیب‌های احتمالی جلوگیری به عمل بیاید و حس خوب بازی حفظ شود.

۴-۲-۱ کلیدهای معمول و پر استفاده موش در بازی

موش در بازی‌های رایانه‌ای دو کاربرد مهم دارد.

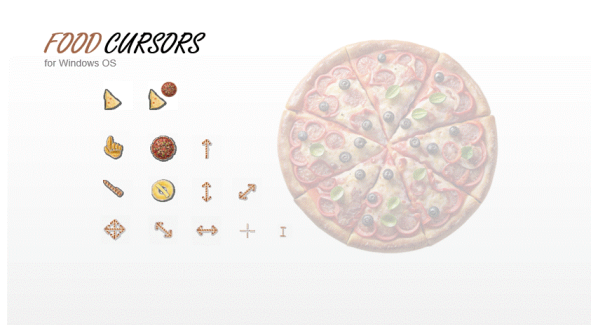


شکل (۸): تصویری از آقای تیوگوتو، طراح دسته بازی برای پیشانه پلی استیشن ۱

طراحی‌های ساده‌تری ساخته می‌شدند. با این حال، یکی از تأثیرگذارترین و موفق‌ترین دسته‌هایی که به شهرت جهانی دست یافت و با استقبال بی‌نظیری روبه‌رو شد؛ دسته پلی‌استیشن ۱ بود. این دسته‌ها حدوداً در سال ۱۹۹۴ میلادی توسط آقای تیوگوتو (Teiyu-Goto) طراحی شدند. شرکت سونی در ژاپن، با طراحی و عرضه این کنسول و دسته‌های آن، و به موازات آن، توسعه و انتشار بازی‌های متعدد و بهره‌گیری از تبلیغات گسترده، محصول خود را در بازارهای جهانی معرفی کرد. این راهبرد موفق، به تثبیت جایگاه پلی‌استیشن به عنوان یکی از پیشروترین و پرفروش‌ترین پیشانه‌های بازی در طول دهه‌های متمادی انجامید و سودآوری قابل‌توجهی را برای شرکت سونی به ارمغان آورد. در شکل ۸ تصویر دسته پلی‌استیشن ۱ همراه با آقای تیوگوتو را مشاهده می‌کنید. البته سود این‌کار فقط مادی نیست بلکه از لحاظ سیاسی و فرهنگی و تاریخی نیز تأثیرات زیادی داشته است.

۳-۴-۱-۳-۴ کلیدهای پر استفاده در دسته‌های بازی

دسته‌های بازی، طبق شکل‌های ۸ و ۹ از دو بخش اصلی سمت‌چپ و سمت‌راست تشکیل شده است. در سمت راست، ۴ کلید اصلی وجود دارد که معمولاً با اشکال ضربدر، دایره، مثلث و مربع شناخته می‌شوند. ضربدر برای تأیید یا حمله، دایره برای پریدن، مثلث برای انجام یک کار و مربع برای نوع دیگری حمله استفاده می‌شود. با اهرم سمت راست، می‌توان زاویه دوربین را تغییر داد.



شکل (۷): تصاویر تزئینی مختلف برای اشاره‌گر موش متناسب با فضای پیتزافروشی

می‌کند. معمولاً این شکل، همانند تصویری است که در شکل ۵ قرار دارد. برخی بازی‌ها برای تغییر فضای بازی و افزایش جذابیت بصری آن، تصویر اشاره‌گر موش را نیز تغییر می‌دهند. این نمادها باعث می‌شود تا مخاطبان ارتباط بیشتری با بازی برقرار کنند و از تکرار یک شکل ساده و ثابت جلوگیری کنند. از لحاظ هنری نیز بسیار اهمیت دارد و شکل نماد باید با فضای داستان هماهنگی داشته باشد. حتی برخی بازی‌ها چندین شکل مختلف را نشان می‌دهند تا مخاطب یکی از آنها را که بیشتر دوست دارد؛ انتخاب نماید. هر چه بیشتر به جزئیات بازی و کیفیت بصری آن رسیدگی شود؛ امتیاز بازی بین مخاطبان و داوران بیشتر خواهد بود و تجربه بهتری را ثبت خواهد کرد. به عنوان مثال در شکل ۷ تصاویری به سبک رستوران‌داری و پیتزافروشی برای اشاره‌گر موش در نظر گرفته شده است. ممکن است نماد بصری موش، یک شمشیر، آبنبات، چوب یا یک شکل انتزاعی دیگر نیز باشد.

۳-۴-۳-۴ دسته‌های بازی

دسته‌های بازی نقشی اساسی در تجربه کاربری و تعامل بازیکن با پیشانه ایفا می‌کنند. این ابزار، به عنوان رابط اصلی و اختصاصی بازیکن با محیط بازی عمل می‌کند. طراحی دلچسب و چیدمان منطقی و در دسترس بودن کلیدها، از جمله عوامل کلیدی در بهبود کاربردپذیری و افزایش راحتی بازیکن در طول تجربه بازی است. از ابتدای دهه ۱۹۷۰ میلادی، انواع دسته‌های بازی با



شکل (۱۰): صفحه نمایش تور در کامپیوتر و تلویزیون کاربرد دارد.

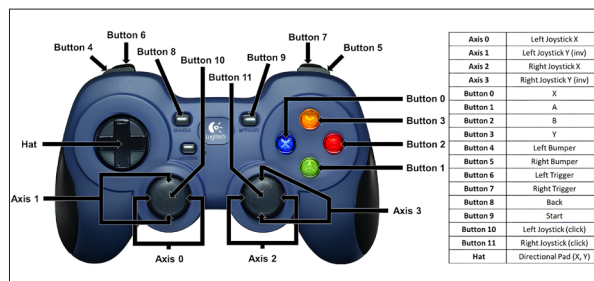
۵- دستگاه‌های خروجی

این دستگاه‌ها، اطلاعات را از کامپیوتر گرفته و به مخاطب می‌رسانند.

۵-۱ صفحه نمایش

صفحه نمایش، به عنوان صفحه نمایش اصلی، بستر نمایش تمامی عناصر بصری بازی، اعم از گرافیک، رنگ‌ها، حرکت‌ها و جزئیات محیط است. کیفیت آن مستقیماً بر تجربه بازی تأثیر می‌گذارد. چه در کامپیوتر، چه در پیشانه‌های بازی و چه در گوشی‌های هوشمند، صفحه نمایش وظیفه انتقال دقیق و واقعی محتوای بازی را بر عهده دارد. اگر صفحه نمایش از کیفیت پایینی برخوردار باشد؛ حتی بهترین گرافیک و طراحی بازی نیز به درستی نمایش داده نمی‌شود. این موارد نه تنها تجربه بصری را تحت تأثیر قرار می‌دهند بلکه باعث خستگی چشم، کاهش تمرکز و اختلال در تامل و تصمیم‌گیری بازیکن نیز می‌شوند. در واقع، صفحه نمایش ضعیف می‌تواند کیفیت واقعی بازی را مخدوش کند و حتی بهترین بازی‌ها را با نمایشی نامناسب و آزاردهنده همراه سازد. بنابراین، صفحه نمایش مناسب، به عنوان رابط اصلی بین بازیکن و بازی، نقش کلیدی در ارائه تجربه‌ای روان، دقیق و لذت‌بخش ایفا می‌کند و انتخاب آن باید متناسب با توانایی‌های سخت‌افزاری دستگاه و نیازهای بازی باشد.

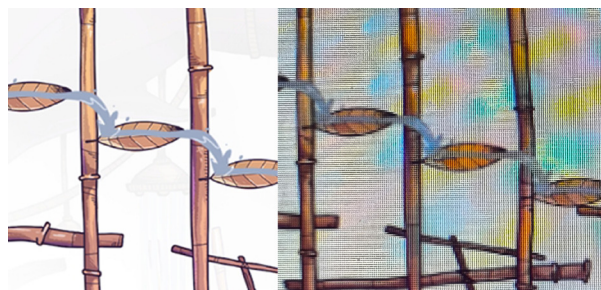
همان‌طور که در شکل ۱۰ آورده شده است؛ صفحه نمایش، نه فقط برای کامپیوتر، بلکه برای گوشی و کامپیوتر و تلویزیون می‌تواند ایفای نقش کند. در واقع تمامی آنها از



شکل (۹): اطلاعات یک دسته بازی

همچنین دو کلید R1 و R2 در بالای بخش راست وجود دارند که معمولاً باید هم‌زمان با یکی از دکمه‌های دیگر فشرده شوند تا کاری را انجام دهند. بخش سمت چپ نیز شبیه به بخش راست است اما در ازای ۴ کلید، ۴ جهت حرکتی را دارد. از آن دکمه‌ها برای حرکت در محیط بازی استفاده می‌شود. البته که اهرم (منظور همان گوی متحرک است) نیز وجود دارند که می‌توان با آن حرکت نمود. در بخش بالای سمت چپ، دو دکمه L1 و L2 وجود دارد که معمولاً با دیگر دکمه‌های دسته باید هم‌زمان فشرده شود تا کاری انجام دهد.

همان‌طور که در شکل ۹ مشاهده می‌شود؛ محورهای اهرم‌ها با اعداد (۰ و ۱) و (۲ و ۳) مشخص می‌شوند. این محورها بر اساس مختصات X و Y مقداردهی می‌شوند. دقت این مقادیر بستگی به نوع کدگذاری دارد که تک‌بایتی (۸بیتی) یا دوبایتی (۱۶بیتی) باشد. در گذشته، معمولاً از روش تک‌بایتی استفاده می‌شد. هر بایت از ۸ بیت تشکیل شده است که هر بیت می‌تواند مقدار ۰ یا ۱ بگیرد. بدین ترتیب، ۲۵۶ حالت ممکن وجود دارد که محدوده مقادیر از ۱۲۸- تا ۱۲۷+ متغیر است. به عنوان مثال، با حرکت اهرم به سمت بالا، مقدار محور عمودی افزایش می‌یابد. در محور افقی نیز تغییرات مشابهی رخ می‌دهد و در نهایت این تغییر مقادیر، به پیشانه یا کامپیوتر ارسال می‌شوند. در روش دوبایتی که در اکثر دسته‌های امروزی استفاده می‌شود؛ دقت به‌طور قابل توجهی افزایش می‌یابد و محدوده مقادیر از ۳۲۷۶۸- تا ۳۲۷۶۷+ گسترش می‌یابد. افزایش دقت باعث روان‌تر شدن عملکرد می‌شود. اگرچه این تفاوت‌ها ممکن است برای کاربر قابل تشخیص نباشد.



شکل (۱۲): سمت راست: عکس‌برداری شده از صفحه نمایش و سمت چپ: چیزی که چشم ما می‌بیند.

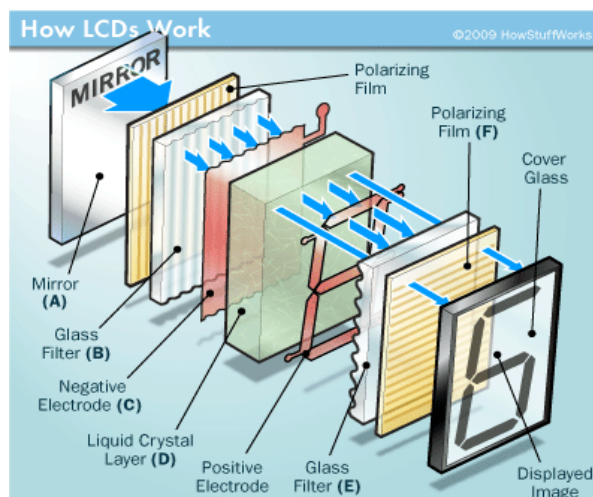
از صفحه نمایش عکسی گرفتیم و آن را با چیزی که نمایش می‌داد مقایسه نمودیم.

با استفاده از رابطه ۳ می‌توان کیفیت خروجی بازی در صفحه نمایش را تنظیم نمود. عدد ۱۲۸۰، تعداد نقطه تصویری در طول و ۷۲۰ تعداد نقطه تصویری در عرض (ارتفاع) است. کلمه true/false برای تمام صفحه بودن/نبودن بازی است که اکثراً در بازی‌های کامپیوتری مورد استفاده قرار می‌گیرد [۱].

رابطه (۳): یک نمونه تنظیمات نمایش صفحه بازی

Screen.SetResolution(1280,720,true);

رابطه ۳ کیفیت خروجی بازی در مانیتور را تنظیم می‌کند. اگر تعداد نقطه تصویری (مربع) های نمایش داده شده کم باشد؛ بازی کم‌کیفیت می‌شود و مربع-مربعی بنظر می‌رسد. جهت افزایش کیفیت، تعداد نقطه تصویری را بیشتر می‌کنند تا کیفیت نمایشی اثر افزایش یابد. البته باید دقت داشت که این افزایش تعداد نقطه تصویری نباید بدون قاعده باشد. اگر تعداد نقطه تصویری را بدون قاعده زیاد کنیم؛ باعث می‌شود تا بازی به اصطلاح سنگین شود و سرعت آن کاهش یابد. سنگینی، در اینجا منظور از پردازش بازی است. اگر تعداد نقطه تصویری بسیار شود؛ پردازش آن بسیار سخت است و پردازنده نمی‌تواند کار خود را به صورت کامل انجام دهد. بدین ترتیب تعداد قاب‌ها پایین می‌آید. به عنوان مثال در یک بازی باید هر ثانیه، ۳۰ قاب داشته باشد اما پردازنده که قدرت پردازش ۳۰ قاب با کیفیت بالا را ندارد؛ تعداد قاب پایین‌تری را پردازش می‌کند. مثلاً ۱۵ قاب را آماده و در صفحه نمایش پخش می‌شود. بدین‌صورت، انگار که سرعت بازی کم شده است



شکل (۱۱): مراحل تولید تا نمایش تصویر در صفحه نمایش

استانداردهای LCD پیروی می‌کنند.

۵-۱- نحوه کارکرد صفحه نمایش

صفحه نمایش‌ها از ساختار پیچیده‌ای برخوردار هستند. شاید صفحه‌کلید و تعداد دکمه‌های آن را بتوان با ماتریس نشان داد اما یک صفحه نمایش، بسیار پیچیده است. داخل هر صفحه نمایش، چند منبع نور سفید قرار دارد. سپس این نور از لایه‌هایی عبور می‌کند تا کاملاً یکسان شود. در مرحله بعدی، از یک صفحه می‌گذرد که تمامی نقاط تصویری در این صفحه وجود دارند. به عنوان مثال برای کیفیت HD حدود ۱۲۸۰ در ۷۲۰ نقطه تصویری وجود دارد و به عنوان استاندارد 720P معروف است. روی این صفحه، ۹۲۱۶۰۰ نقطه تصویری وجود دارد. هر نقطه تصویری، ۳ زیر نقطه دارد که رنگ‌های اصلی یعنی قرمز و سبز و آبی هستند. نور سفید به این لایه می‌رسد و هر نقطه تصویری، اطلاعات خود را از کارت گرافیک دریافت می‌کند. سپس تصمیم می‌گیرد که چه جریانی را به هر زیر نقطه بدهد. جریان برق مختلف، باعث می‌شود که شدت سختی آنها متفاوت شود. بدین‌صورت، نوری که از آنها خارج می‌شود نیز متفاوت است و رنگ‌های مختلف به نمایش در می‌آیند. در شکل ۱۱ به صورت خلاصه این مراحل نمایش داده شده است.

همان‌طور که در شکل ۱۲ می‌بینید؛ با یک دوربین قوی

روانشناختی مهم در طراحی بازی است. صداهای مناسب می‌توانند همراه واکنش‌های مخاطبان باشند. این یعنی یک تعامل موفق صورت پذیرفته است. از حس انتقال ترس تا دلنشینی و شاد بودن یک موسیقی، توانسته احساسات را منتقل کند و با مخاطب تعامل داشته باشد. بنابراین، انتخاب و پیاده‌سازی صحیح صداها، نه تنها به بهبود کیفیت فنی بازی کمک می‌کند بلکه تجربه احساسی و عاطفی مخاطب را نیز عمیق‌تر می‌سازد. در نتیجه، بلندگوها و سیستم‌های صوتی، به عنوان رابطی بین دنیای واقعی و مجازی، نقش اساسی در ایجاد یک تجربه بازی فراموش‌نشدنی ایفا می‌کنند.

۶- نتیجه‌گیری

تعامل مؤثر کاربر با بازی، مستلزم درک دقیق سازوکارهای ورودی و خروجی و پیاده‌سازی بهینه آنها در موتورهای بازی‌سازی است. این مقاله با بررسی دستگاه‌های ورودی و خروجی رایج و تحلیل کدهای نمونه در یونیتی، نشان داد که چگونه ورودی‌های فیزیکی کاربر به عملیات درون بازی ترجمه می‌شوند و چگونه خروجی‌های بصری و صوتی، تجربه بازی را به مخاطبان نمایش می‌دهند و این تجربه را غنی‌تر می‌سازند. آشنایی با یونیتی و اصول سخت‌افزاری، بازیسازان را قادر می‌سازد تا رابط‌های کاربری پاسخگو و تجربه‌های بازی جذاب‌تری را خلق نمایند.

مراجع

۱. اردلان، امیرحسین. (۱۴۰۴). «دستگاه‌های ورودی و خروجی در تعامل و ساخت بازی (یونیتی)»، نماشا [ویدیو]. قابل دسترسی در: <https://www.namasha.com/v/aXo8OsGc>
2. Hocking, Joseph. (2015). Unity in Action: Multiplatform game development in C# with Unity 5, Manning Publications, Shelter Island.
3. Stallings, William. (2018). Computer Organization and Architecture: Designing for Performance, Eleventh Edition, Pearson, New York.
4. Universal Serial Bus Implementers Forum (USB-IF). (2004). HID Usage Tables, Version 1.12, October 28, Available at: usb.org.
5. Patterson, David A., Hennessy, John L. (2013). Computer Organization and Design: The Hardware/Software Interface, Fifth Edition, Morgan Kaufmann.

و تصاویر با اختلال نمایش داده می‌شوند. این امر باعث بازخوردهای منفی از بازی می‌شود و فقط کسانی که بودجه پردازنده‌های قوی را دارند از بازی لذت می‌برند و این امر اصلاً مطلوب نیست. نکته دیگری که باید به آن توجه داشت؛ کیفیت خود صفحه نمایش است. آیا دستگاه موردنظر کیفیت در نظر گرفته شده را می‌تواند پخش کند یا قادر به آن نیست؟ اگر برای یک بازی، کیفیت بسیار بالا را در نظر بگیرید اما نمایشگر قدرت پخش آن را نداشته باشد؛ در حقیقت کیفیت بالا پخش نخواهد شد و بازی بازخوردهای ضعیفی را خواهد گرفت.

۵-۲ بلندگو

صوت در بازی‌های رایانه‌ای تنها به عنوان یک عنصر جانبی نیست بلکه یکی از مهمترین ابزارها برای ایجاد ارتباط عمیق بین مخاطب و دنیای مجازی است. بلندگوها و سیستم‌های صوتی، با انتقال دقیق و باکیفیت صداهای محیطی، موسیقی زمینه، دیالوگ‌ها و جلوه‌های ویژه صوتی، نقش کلیدی در شکل‌گیری احساسات و تجربه کاربر ایفا می‌کنند. برای مثال، صدای قدم‌های یک شخصیت در یک محیط تاریک و مرموز، می‌تواند حس ترس و تنش را در مخاطب تقویت کند، در حالی که موسیقی دلنشین و شاد در یک بازی ماجراجویی، انرژی و هیجان را افزایش می‌دهد. این امر در کیفیت نهایی بازی و تعامل با مخاطبان بسیار اهمیت دارد و باید جدی گرفته شود.

در موتور بازی‌سازی یونیتی، امکانات گسترده‌ای برای مدیریت صوتی وجود دارد. از تنظیم حجم و کیفیت صداهای مختلف گرفته تا استفاده از ابزارهای مختلف صوتی، همه و همه به سازندگان بازی این امکان را می‌دهند تا تجربه صوتی منحصربه‌فردی برای مخاطب خلق کنند. همچنین، استفاده از تکنیک‌های صوتی سه‌بعدی در یونیتی، باعث می‌شود مخاطب بتواند جهت و فاصله منابع صوتی را در محیط بازی تشخیص دهد که این امر حس غوطه‌وری در بازی را به شدت افزایش می‌دهد.

ارتباط بین صوت و احساسات مخاطب، یکی از جنبه‌های