

به نام خدا

گزارش چهارم درس آسخت افزار

گروه 6

آرمان زارعی

کیوان رضائی

سیدمحمد سیدجوادی

مقدمه

در این قسمت پروژه را بر روی Raspberry pi بردیم و از طریق آن و کتابخانه‌ای که مردمک چشم را تشخیص می‌داد، سیگنال‌های چپ و راست را به بازی‌ای که در گزارش قبلی پیاده‌سازی کردیم، می‌دهیم.

توضیح

در ابتدا می‌بایست کتابخانه‌های پیشنیازی که برای پروژه لازم بود را نصب کنیم. به دلیل کند بودن رزبری پای فقط کتابخانه‌هایی که در pip wheels وجود داشت را می‌توانستیم نصب کنیم. کتابخانه‌های دیگر مدت زیادی (مثلاً ۵ الی ۶ ساعت) برای بیلد شدن نیاز داشتند و در انتها نیز اکثراً با خطا مواجه می‌شدند. نهایتاً از پایتون ۳٫۵ استفاده کردیم تا شرایط مذکور را داشته باشد.

در فایل tracker.py یک تابع وجود دارد که به مدت ۵ ثانیه چشم بازیکن را در نظر می‌گیرد. سپس با توجه به عکس‌های این ۵ ثانیه دست راست یا چپ را در بازی گل یا پوچی که زدیم مشخص می‌کند.

در فایل server.py که قرار است روی raspberry pi اجرا شود، روی یک پورت منتظر می‌ماند تا کد مربوط به بازی به آن وصل شود و سرور برای تشخیص مردمک چشم از فایل tracker.py استفاده می‌کند و تصمیم نهایی کاربر را برای بازی می‌فرستد.

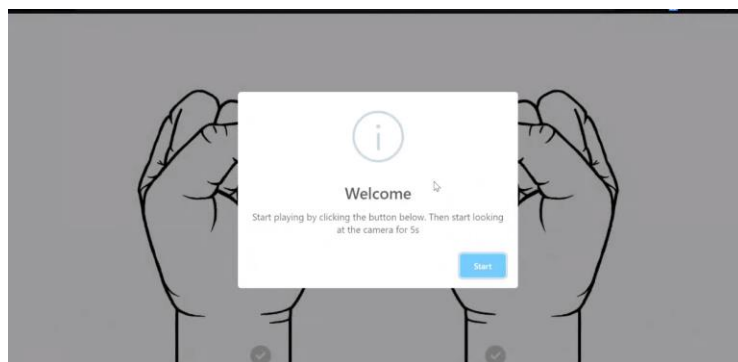
اجرای برنامه

در این قسمت یک بار اجرای کل سیستم را مشاهده می‌کنیم. در ابتدا کد server.py را بر روی رزبری پای اجرا کردیم.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  1
pi@raspberrypi:~/Desktop/project/project-team-6/Code/v2.0 $ /usr/bin/python3.5 server.py
Server is starting ...
(8017) wsgi starting up on http://0.0.0.0:4343
```

شکل 1 اجرای سرور

سپس بازی را بر روی کامپیوتر اجرا کردیم.



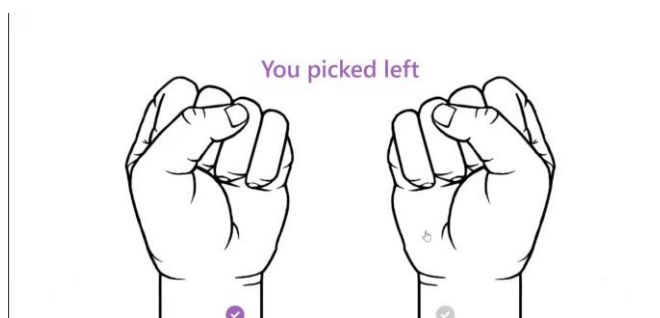
شکل 2 شروع بازی

بازی ۵ ثانیه به بازیکن مهلت می‌دهد که جهت مورد نظرش را نگاه کند.

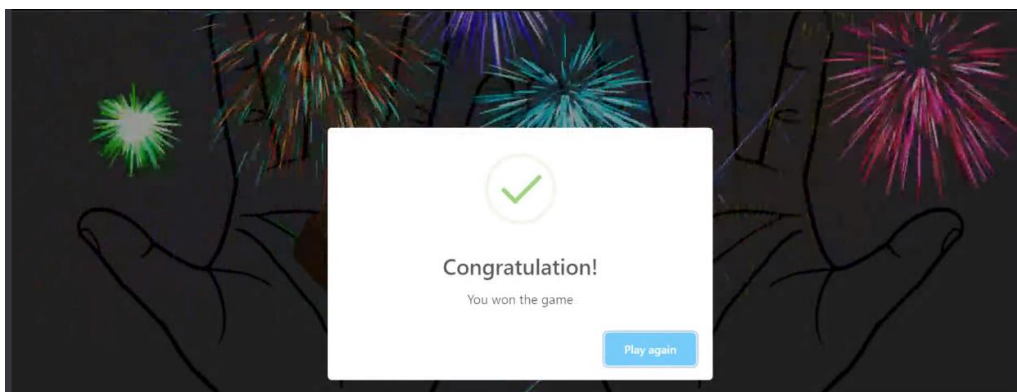


شکل 3 زمان برای انتخاب بازیکن

سپس نتیجه را اعلام می‌کند.



شکل 4 گرفتن نتیجه کاربر



شکل 5 نمایش نتیجه بازی

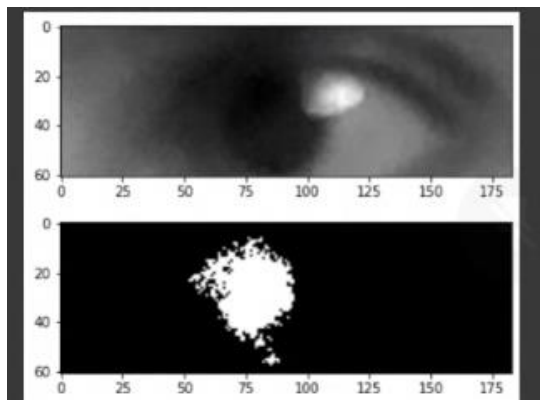
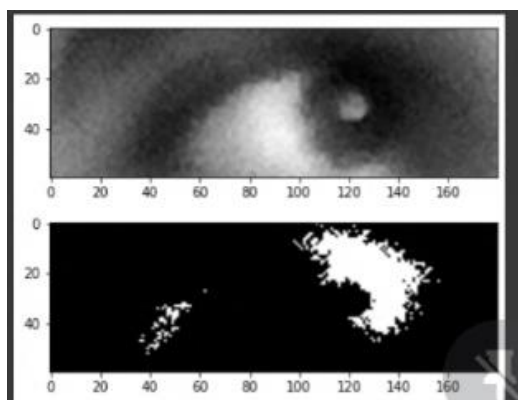
لاگ‌هایی که در سمت `server.py` نیز چاپ شده‌اند به شکل زیر می‌باشد. در ابتدا `handshake` انجام شده و در ۵ ثانیه ۴ عکس گرفته شده که اکثراً به سمت چپ بوده برای همین دست سمت چپ انتخاب شده.

```
(8017) accepted ('192.168.1.103', 51978)
connect 71d51f447227466292ee2617ffefdf68
handshake {'duration': 5}
Pupils' locations: None None
Direction: Looking center
Pupils' locations: (276, 281) (352, 275)
Direction: Looking left
Pupils' locations: (274, 273) (349, 263)
Direction: Looking left
Pupils' locations: (263, 277) (338, 266)
Send to Game ... LEFT
```

شکل 6 لاگ‌های سرور

تعویض قسمت تشخیص مردمک چشم

در انتها به دلیل اینکه کتابخانه‌ی مورد استفاده جهت تشخیص مردمک چشم به درستی کار نمی‌کرد، مجبور شدیم خودمان کد تشخیص مردمک چشم را پیاده‌سازی کنیم. برای این کار از تابع تشخیص چشم مربوط به `opencv` استفاده کردیم و قسمت مربوط به چشم را سیاه و سفید کرده و نقطه‌هایی که از یک مقدار سیاه‌تر هستند را مردمک چشم در نظر گرفتیم. در انتها `Median` این نقاط سیاه اگر سمت چپ تصویر بود بازیکن به سمت چپ نگاه می‌کند و بالعکس.



شکل 7 سیستم تشخیص مردمک جدید

نهایتاً کدی که زدیم هم سریعتر شد و هم دقت بالاتری داشت.

```
connect e9132376aa854c4c9089cbf8af7002e9
handshake {'duration': 5}
(480, 640, 3) <class 'numpy.ndarray'>
Eye Direction: LEFT
(480, 640, 3) <class 'numpy.ndarray'>
Eye Direction: LEFT
(480, 640, 3) <class 'numpy.ndarray'>
Eye Direction: LEFT
(480, 640, 3) <class 'numpy.ndarray'>
Eye Direction: LEFT
(480, 640, 3) <class 'numpy.ndarray'>
Eye Direction: LEFT
(480, 640, 3) <class 'numpy.ndarray'>
Eye Direction: LEFT
(480, 640, 3) <class 'numpy.ndarray'>
Eye Direction: LEFT
(480, 640, 3) <class 'numpy.ndarray'>
Eye Direction: LEFT
(480, 640, 3) <class 'numpy.ndarray'>
Eye Direction: LEFT
Result sent to game: LEFT
```

شکل 8 لاگ‌های مربوط به سرور با سیستم جدید تشخیص مردمک