






گزارش نهایی پروژه

حرکت دسته جمعی ربات‌ها (Swarm Robotics)

 آکادمی هوش	حرکت دسته جمعی ربات‌ها (Swarm Robotics)			 مرکز تحقیق و توسعه هوش اول
	گزارش نهایی پروژه			
	نسخه	نام فایل	عنوان فاز	
	تاریخ گزارش: ۱۴۰۲/۰۸/۲۸	A1.0	LFT-AR-SCS-01/T[۱]/Rev[۳]	



شناسنامه مستند					
عنوان پروژه			حرکت دسته جمعی ربات‌ها (Swarm Robotics)		
عنوان مستند			گزارش نهایی		
کد مستند					
کارفرما			مرکز تحقیق و توسعه همراه اول		
پیمانکار			تیم نخبگان دانشجویی آکادمی همراه اول: سید سپهر المدنی، حسین فرحناکی، علی ترکمن، متین زارع بیدکی		
تاریخ تهیه			۱۴۰۲/۰۸/۲۸		
تعداد صفحات					
تاریخچه تغییرات					
ردیف	ویرایش	هدف انتشار	تهیه کننده	بررسی	تأیید
۱-		<div><input type="checkbox"/> جهت بررسی و اعلام نظر</div> <div><input checked="" type="checkbox"/> جهت تأیید</div> <div><input type="checkbox"/> جهت اطلاع</div> <div><input type="checkbox"/></div>			

	حرکت دسته جمعی ربات ها (Swarm Robotics)		
	گزارش نهایی پروژه		
	نسخه	نام فایل	عنوان فاز
	A1.0	LFT-AR-SCS-01/T[۱]/Rev[۳]	نهایی
تاریخ گزارش: ۱۴۰۲/۰۸/۲۸			





پیشگفتار

این پروژه با هدف بررسی فناوری Swarm و با تاکید بر شبیه سازی پرواز دسته جمعی پهپادها برای نمایش یک عبارت با شکل در آسمان جهت تبلیغات و اطلاع رسانی تعریف شده است. پروژه در اداره مدیریت سامانه ها و تجهیزات شبکه مرکز تحقیق و توسعه همراه اول تعریف شده و توسط تیم دانشجویان نخبه آکادمی همراه اول انجام گرفته است. در ابتدای کار، تحقیقاتی در مورد فناوری Swarm و الگوریتم های مختلف آن بیان شده و سپس کدها و نتایج شبیه سازی ها آورده شده است.

 آکادمی علوم و فناوری	حرکت دسته جمعی ربات‌ها (Swarm Robotics)			 مرکز تحقیق و توسعه
	گزارش نهایی پروژه			
	نسخه	نام فایل	عنوان فاز	
تاریخ گزارش: ۱۴۰۲/۰۸/۲۸	A1.0	LFT-AR-SCS-01/T[۱]/Rev[۳]	نهایی	


فهرست مطالب

صفحه	عنوان
ب	فهرست شکل‌ها
۱	فصل ۱- تحقیقات
۱-۱	مقدمه
۱-۲	ویژگی‌های کلی swarm
۱-۳	بررسی سخت‌افزار و نحوه اتصال به کدهای نرم‌افزاری
۱-۴	برآورد قیمت پهپاد تبلیغاتی
۱-۵	الگوریتم‌ها
۱-۵-۱	الگوریتم بهینه‌سازی توده‌ی ذرات (Particle Swarm Optimization یا PSO)
۱-۵-۲	الگوریتم بهینه‌سازی کلونی مورچگان (Ant Colony Optimization یا ACO)
۱-۵-۳	الگوریتم Boids
۱-۵-۴	الگوریتم جستجوی غذا (Foraging Algorithm)
۱-۵-۵	الگوریتم زنجیره مارکوف مونت کارلو (MCMC)
۸	فصل ۲- بکارگیری الگوریتم‌ها و نوشتن کد برای Drone Show
۲-۱	روش اول
۲-۲	روش دوم
۲-۳	روش سوم
۱۵	فصل ۳- شبیه‌سازی در محیط سه بعدی نرم‌افزار بلندر

 آکادمی علوم و فناوری	حرکت دسته جمعی ربات‌ها (Swarm Robotics)			 مرکز تحقیق و توسعه فرااولد
	گزارش نهایی پروژه			
	نسخه	نام فایل	عنوان فاز	
تاریخ گزارش: ۱۴۰۲/۰۸/۲۸	A1.0	LFT-AR-SCS-01/T[۱]/Rev[۳]	نهایی	

فهرست شکل‌ها

صفحه	عنوان
شکل ۱: خواندن مختصات و حرکت دادن یک جسم ۱۵	
شکل ۲: مدل‌سازی پهنپای ۱۶	
شکل ۳: شبیه‌سازی انیمشین تعدادی جسم و حرکت از دایره به مربع یا معادله دلخواه ۱۶	
شکل ۴: کد نهایی شامل خواندن از فایل، سیو کردن و اجرا کردن همزمان و حرکت دادن اجسام ۱۷	

	حرکت دسته جمعی ربات‌ها (Swarm Robotics)		
	گزارش نهایی پروژه		
	نسخه	نام فایل	عنوان فاز
تاریخ گزارش: ۱۴۰۲/۰۸/۲۸	A1.0	LFT-AR-SCS-01/T[۱]/Rev[۳]	نهایی

فصل ۱ - تحقیقات

۱-۱- مقدمه

در این فصل با توجه به نیاز اولیه پروژه و عدم آشنایی دقیق تیم به مدت حدوداً یک ماه فاز تحقیقات صورت گرفت که ابتدا به موردهای استفاده از swarm و همچنین الگوریتم‌های آن پرداخته شد. الگوریتم‌هایی که بسیار متنوع بوده و با توجه به موضوع کاری ما ممکن بود هر کدام پربازده‌ترین باشند، بنابراین نیاز به کسب اطلاعات و همچنین مطالعه‌ی بیشتر در این حوزه ضروری بود. در ادامه نتایج این مطالعات ارائه می‌گردد.

۱-۲- ویژگی‌های کلی swarm



به طور کلی، لزوم استفاده از این فناوری زمانی دیده می‌شود که ما نیاز به انجام عملیات‌های پیچیده در مکان‌هایی داریم که امکان حضور فرد کنترلگر وجود ندارد. عملیات‌هایی مانند جوشکاری لوله‌های انتقال گاز در اعماق اقیانوس‌ها و یا انجام عملیات بررسی و رسم نقشه برای مناطق پرخطر و یا معادن. در این ماموریت‌ها و یا عملیات‌های مشابه به دلیل لزوم انجام محاسبات پیچیده و همچنین وجود توانایی تصمیم‌گیری و هماهنگی ربات‌ها با یکدیگر از swarm استفاده می‌کنیم.

اولین و مهم‌ترین ویژگی swarm، این است که نحوه انجام عملیات به صورت decentralized می‌باشد به این معنی که ربات‌ها بدون استفاده از کنترل مرکزی و با هماهنگی یکدیگر عمل می‌کنند پس هر کدام از ربات‌ها باید دارای توانایی پردازش بوده و بتواند با دیگر ربات‌ها ارتباط برقرار کرده و با دریافت اطلاعات عمل کند.

دومین ویژگی، توانایی تک تک ربات‌ها در انجام ماموریت‌ها و محاسبات پیچیده به تنهایی است به نحوی که هر کدام از ربات‌ها بتوانند اطلاعاتی را از محیط دریافت کرده و با استفاده از اطلاعات خود و دیگر ربات‌ها تصمیم‌گیری کرده و عملیات را با هم انجام دهند.

سومین ویژگی به قابلیت مقیاس‌پذیری^۱ اشاره دارد به این معنی که این فناوری می‌تواند کاملاً efficient با یک یا ۱۰۰ ربات به کار خود ادامه دهد. به عبارت دیگر مهم نیست که عملیات ما به چند

^۱ Scalability

 آکادمی هاد	حرکت دسته جمعی ربات‌ها (Swarm Robotics)			 مرکز تحقیق و توسعه هاد اول
	گزارش نهایی پروژه			
	نسخه	نام فایل	عنوان فاز	
تاریخ گزارش: ۱۴۰۲/۰۸/۲۸	A1.0	LFT-AR-SCS-01/T[۱]/Rev[۳]	نهایی	

ربات داده می‌شود در هر صورت ربات‌ها قادر به انجام ماموریت به درستی خواهند بود و افزایش تعداد ربات‌ها تنها در بیشتر کردن اطلاعات و همچنین انجام سریع‌تر کارها به ما کمک می‌کند.

چهارمین ویژگی این است که Swarm بتواند ماموریت را حتی با وجود وقوع اتفاقاتی نظیر مشکل فنی در یکی از ربات‌ها با موفقیت به پایان برساند. در واقع این ویژگی این امکان را می‌دهد که حتی در صورت از دست دادن یک یا چند ربات باز هم عملیات انجام پذیرد.

ویژگی پنجم، توانایی ربات‌ها در تطبیق‌پذیری با شرایط و مشکلات ماموریت‌های مختلف است. برای مثال ربات‌ها باید این توانایی را داشته باشند که در تمامی وضعیت‌های آب و هوایی و یا اقلیمی و محیطی ماموریت خود را با موفقیت به پایان برسانند.

۳-۱- بررسی سخت‌افزار و نحوه اتصال به کدهای نرم‌افزاری

در این بخش، تحقیقاتی بر روی بخش سخت‌افزاری و همچنین نحوه اتصال آن به کدهای موجود که از بخش شبیه‌سازی در اختیار ما قرار دارد، انجام شده است. با توجه به در دست داشتن پروژه واقعی توسط تیم پیمانکار برای مرکز تحقیق و توسعه همراه اول، تحقیقات در مورد نیازهای پروژه به طور پیش فرض بر روی رابط کاربری Pixhawk انجام شده است اما جایگزین‌هایی نیز در ادامه پیشنهاد شده است.



کتابخانه Pixhawk Python یک رابط کاربری پایتون برای سخت‌افزار Pixhawk است. این کتابخانه به کاربران اجازه می‌دهد تا با استفاده از پایتون با Pixhawk ارتباط برقرار کنند و داده‌ها را از آن بخوانند و بنویسند. کتابخانه Pixhawk Python شامل موارد زیر است:

- توابع و کلاس‌هایی برای اتصال به Pixhawk
- توابع و کلاس‌هایی برای خواندن داده‌ها از حسگرهای Pixhawk
- توابع و کلاس‌هایی برای نوشتن داده‌ها به Pixhawk
- توابع و کلاس‌هایی برای تنظیم پارامترهای Pixhawk

کتابخانه Pixhawk Python توسط جامعه Pixhawk توسعه یافته و به صورت رایگان در دسترس است. برای نصب کتابخانه Pixhawk Python می‌توان از دستور زیر استفاده کرد:

```
pip install pixhawk
```

پس از نصب کتابخانه، می‌توان شروع به نوشتن کد پایتون برای Pixhawk کرد. در ادامه یک نمونه ساده از نحوه استفاده از کتابخانه Pixhawk Python آورده شده است:

 آکادمی‌ها	حرکت دسته جمعی ربات‌ها (Swarm Robotics)			 مرکز تحقیق و توسعه هوادک
	گزارش نهایی پروژه			
	نسخه	نام فایل	عنوان فاز	
تاریخ گزارش: ۱۴۰۲/۰۸/۲۸	A1.0	LFT-AR-SCS-01/T[۱]/Rev[۳]	نهایی	

```
import pixhawk

# Connect to the Pixhawk
pixhawk = pixhawk.connect("/dev/ttyACM0")



# Read the gyroscope data
gyro_data = pixhawk.get_gyro()

# Print the gyroscope data
print("X:", gyro_data.x)
print("Y:", gyro_data.y)
print("Z:", gyro_data.z)
```

این کد ابتدا به پورت سریال /dev/ttyACM0 متصل می‌شود. سپس، داده‌های ژيروسکوپ را با استفاده از تابع `get_gyro()` می‌خواند. در نهایت، داده‌های ژيروسکوپ را چاپ می‌کند.

برای کسب اطلاعات بیشتر در مورد نحوه استفاده از کتابخانه Pixhawk Python و همچنین کسب اطلاعات بیشتر در مورد توابع و میزان دسترسی به سنسورها می‌توان به مستندات کتابخانه مراجعه کرد. در زیر چند نمونه دیگر از نحوه استفاده از کتابخانه Pixhawk Python آورده شده است:

- نوشتن کد برای کنترل موتورهای Pixhawk
 - نوشتن کد برای خواندن داده‌ها از حسگرهای دیگر Pixhawk، مانند حسگر Accelerometer، حسگر Magnetometer و حسگر Barometer
 - نوشتن کد برای تنظیم پارامترهای Pixhawk
- اگر رابط کاربری انتخاب شده برای انجام پروژه Pixhawk نباشد، باید رابط کاربری دیگری برای سخت-افزار خود انتخاب کرد. رابط کاربری باید بتواند با سخت‌افزار موجود در پروژه ارتباط برقرار کند و داده‌ها را از آن بخواند و بنویسد. چند گزینه برای انتخاب رابط کاربری وجود دارد:
- ❖ کتابخانه‌های آماده: چندین کتابخانه آماده برای ارتباط با سخت‌افزارهای مختلف وجود دارد. این کتابخانه‌ها می‌توانند کار برنامه‌نویسی را آسان کنند.
 - ❖ رابط کاربری سفارشی: اگر نیازهای خاصی در پروژه وجود داشته باشد، می‌توان یک رابط کاربری سفارشی ایجاد کرد. این کار می‌تواند زمانبر و دشوار باشد، اما کنترل بیشتری بر عملکرد رابط کاربری در اختیار ما قرار می‌دهد.
- در اینجا دو مورد از کتابخانه‌های آماده برای ارتباط با سخت‌افزارهای مختلف آورده شده است:
- ❖ کتابخانه QgroundControl: این کتابخانه برای کنترل و نظارت بر پرواز پهپادها استفاده می‌شود.

 آکادمی‌ها	حرکت دسته جمعی ربات‌ها (Swarm Robotics)			 مرکز تحقیق و توسعه هوادک
	گزارش نهایی پروژه			
	نسخه	نام فایل	عنوان فاز	
تاریخ گزارش: ۱۴۰۲/۰۸/۲۸	A1.0	LFT-AR-SCS-01/T[۱]/Rev[۳]	نهایی	

❖ کتابخانه Mavlink: این کتابخانه برای ارتباط با سخت‌افزارهای مبتنی بر Mavlink استفاده می‌شود.

برای انتخاب رابط کاربری مناسب، باید نیازهای پروژه به طور کامل در نظر گرفته شود. اگر نیاز به کنترل دقیق سخت‌افزار وجود داشته باشد، ممکن است به این نتیجه برسیم که بهتر است یک رابط کاربری سفارشی ایجاد شود. اگر نیاز به یک راه‌حل سریع و آسان وجود داشته باشد، می‌توان از یک کتابخانه آماده استفاده کرد. با کمی تحقیق درباره‌ی نیازهای پروژه، می‌توان رابط کاربری مناسبی برای سخت‌افزار پیدا کرد. در زیر چند نکته برای انتخاب رابط کاربری مناسب آورده شده است:

- ✓ نیازهای پروژه در نظر گرفته شود: چه کاری باید با سخت‌افزار انجام شود؟ آیا نیاز به کنترل دقیق آن وجود دارد؟ آیا نیاز به خواندن داده‌ها از آن وجود دارد؟
- ✓ قابلیت‌های رابط کاربری بررسی شود: آیا رابط کاربری می‌تواند نیازهای پروژه را برآورده کند؟ آیا مستندات کاملی برای آن وجود دارد؟ آیا پشتیبانی فنی در دسترس است؟


۴-۱- برآورد قیمت پهپاد تبلیغاتی

در این بخش، هزینه‌ی حدودی یک پهپاد^۱ و همچنین ابزار و امکانات مورد نیاز برای اجرای مأموریت پرواز دسته جمعی جهت تبلیغات آمده است. این بخش از تحقیقات، شامل دو بخش می‌شد که بخش اول آن به تعیین ابزار مورد نیاز و همچنین کارآمد برای پهپادها و انجام عملیات می‌پردازد و بخش دوم به پیدا کردن به صرفه‌ترین قیمت و کارآمدترین مدل از بین تمامی مدل‌ها اختصاص دارد.

در بخش اول با توجه به انتظارات از drone و همچنین عملیات مورد نظر، نیاز به وجود یک ابزار موقعیت‌سنجی دقیق در کنار GPS احساس می‌شود که برای این کار از RTK استفاده می‌شود. در بخش ارتباط بین نرم‌افزار از بوردهای برند Pixhawk استفاده می‌گردد که یک برد open source و دارای reliability بسیار بالا بوده و کار با آن بسیار راحت است. مدل‌های pixhawk 4 و همچنین مدل پیشرفته‌تر و البته گران‌تر pixhawk cube کاملاً توانایی اجرای مأموریت‌های دلخواه ما برای تبلیغات را دارند.

همچنین نیاز به یک برد فرستنده-گیرنده‌ی ساده برای انتقال و دریافت پیام از مرکز کنترل به پهپادها وجود دارد که این برد به راحتی و با قیمت به نسبت پایین قابل تهیه است. برای بخش بدنه و سنسورهای حرکت نیز ما در بخش برآورد هزینه از یک مدل از پیش ساخته شده استفاده کردیم که البته

^۱ Drone

	حرکت دسته جمعی ربات‌ها (Swarm Robotics)		
	گزارش نهایی پروژه		
	نسخه	نام فایل	عنوان فاز
تاریخ گزارش: ۱۴۰۲/۰۸/۲۸	A1.0	LFT-AR-SCS-01/T[۱]/Rev[۳]	نهایی

می‌توان این مدل را نیز با استفاده از قطعات اولیه در داخل کشور مونتاژ کرد که باعث صرفه‌جویی در هزینه نهایی خواهد شد. در نهایت با توجه به تمام موارد گفته شده و همچنین برآورد قیمت‌ها برای خرید از سایت‌هایی مانند آمازون و یا سایت اصلی تولیدکننده قطعه در مجموع هزینه‌ی مورد نیاز برای بکار بردن یک پهپاد با قابلیت‌های گفته شده چیزی نزدیک به ۱۰۰۰ تا ۱۲۰۰ دلار تخمین زده می‌شود.

۱-۵- الگوریتم‌ها

در مسائلی مانند مدیریت و نگهداری تعداد زیادی ربات مثل پهپادها یا ربات‌های انبارداری یا خط تولید و غیره، سیستم پردازش مرکزی و مدیریت آن‌ها بزرگ و نابهینه بود و زمان زیادی برای توسعه سیستم پردازش صرف می‌شد. Swarm intelligence، با الگوبرداری از طبیعت مسیری جدیدی برای حل این مسائل معرفی می‌کند. به جای یک سیستم مرکزی، سیستم پراکنده و نامتمرکز می‌شوند. هر ربات برای خود عمل می‌کند و صرفاً با پروتکل‌های ارتباطی ساده در کنار هم کارهای بزرگ را انجام می‌دهند.


یکی از نمونه‌های این تکنولوژی در یک پروژه تحقیقاتی که ربات‌هایی کرم‌مانند با قابلیت حرکت و اتصال به دیگر ربات‌ها ساخته شده بودند، به نمایش گذاشته شد. در این پروژه، ربات‌ها همواره می‌توانستند به جلو حرکت کنند به جز وقتی که مسیری در جلوی ربات نبود و یا ربات دیگری به ربات اصلی متصل می‌شد. با استفاده از این روش، مسئله‌ی عبور ربات‌ها از موانع به جای محاسبات پیچیده‌ی متمرکز به راحتی با استفاده از هوش‌های ساده و غیرمتمرکز حل شد. در ادامه، به بررسی و بیان الگوریتم‌های Swarm پرداخته شده است.

۱-۵-۱- الگوریتم بهینه‌سازی توده‌ی ذرات (Particle Swarm Optimization یا PSO)

یک تکنیک بهینه‌سازی الهام گرفته از طبیعت است که رفتار یک گله پرندگان یا یک توده ذرات را شبیه‌سازی می‌کند تا مسائل بهینه‌سازی پیچیده را حل نماید. در PSO، جمعیتی از راه‌حل‌های پتانسیل، که به صورت ذرات نمایان می‌شوند، در فضای جستجوی چندبعدی حرکت کرده و مکان‌های خود را به‌طور مداوم بر اساس تجربیات خود و تجربیات هم‌نوعان‌شان تنظیم می‌کنند. حرکت هر ذره توسط بهترین راه‌حل پیشین خود و بهترین راه‌حلی که توسط هر ذره دیگری در توده یافته شده است، هدایت می‌شود. با گذر زمان، PSO به‌طور کارآمد فضای جستجو را بررسی کرده و به سوی راه‌حل بهینه همگرا می‌شود. سادگی، سرعت و توانایی مدیریت گسترده‌ای از مسائل بهینه‌سازی، PSO را به یک انتخاب محبوب در زمینه‌هایی مانند مهندسی، یادگیری ماشین و رباتیک تبدیل کرده است.

۱-۵-۲- الگوریتم بهینه‌سازی کلونی مورچگان (Ant Colony Optimization یا ACO)

یک رویکرد بهینه‌سازی مشتق شده از طبیعت است که الهام خود را از رفتار جستجوی غذا توسط مورچه‌ها می‌گیرد. در ACO، جمعیتی از مورچه‌های مصنوعی از طریق یک فضای حل حرکت می‌کنند و

	حرکت دسته جمعی ربات‌ها (Swarm Robotics)		
	گزارش نهایی پروژه		
	نسخه	نام فایل	عنوان فاز
تاریخ گزارش: ۱۴۰۲/۰۸/۲۸	A1.0	LFT-AR-SCS-01/T[۱]/Rev[۳]	نهایی


سعی می‌نمایند راه‌حل‌های بهینه برای مسائل پیچیده را پیدا کنند. این مورچگان مسیرهای عطری را ترسیم کرده و ارتباط برقرار می‌کنند و مسیر خود را ثبت می‌نمایند که به آن‌ها امکان بهبود نسبی مسیر جستجو به نسبت مدت زمان را می‌دهد. مورچه‌ها تحت تأثیر عطرها و یک تابع شهودی قرار می‌گیرند که جذابیت مسیرهای مختلف را ارزیابی می‌کند. ACO در حل مسائل بهینه‌سازی ترکیبی به خوبی عمل می‌کند. انعطاف‌پذیری و کارایی آن در یافتن راه‌حل‌های نزدیک به بهینه آن را به یک ابزار ارزشمند در زمینه‌های مختلف اعمالی مانند حمل و نقل، مسیریابی شبکه و برنامه‌ریزی می‌کند.

۳-۵-۱ الگوریتم Boids

یک مدل محاسباتی است که تحت تأثیر رفتار گله‌گیری پرندگان و سایر حیوانات اجتماعی ایجاد شده است. در این الگوریتم، جمعیتی از عوامل مصنوعی به نام "Boids" حرکات افراد در یک گروه را شبیه‌سازی می‌کنند. هر Boid با پیروی از قوانین ساده به منظور رفتار جمعی عمل می‌کند؛ جداشدگی، تطابق و واحدیت. جداشدگی Boids را ترغیب می‌کند تا حداقل فاصله‌ای از همسایگان خود حفظ کنند تا از برخورد جلوگیری کنند. تطابق آن‌ها را راهنمایی می‌کند تا سرعت و جهت Boids نزدیک را همسان کنند و تمرکز گروه را تشویق کند. واحدیت باعث می‌شود Boids به سمت مرکز جرم همسایگان خود حرکت کنند و ساختار کل گله را حفظ نمایند. این سه قانون به Boids امکان نمایش رفتارهای پیچیده‌ای مانند گله‌گیری را می‌دهد. الگوریتم Boids در زمینه‌های گرافیک کامپیوتری، انیمیشن و شبیه‌سازی کاربردهایی دارد.

۴-۵-۱ الگوریتم جستجوی غذا (Foraging Algorithm)



یک تکنیک بهینه‌سازی الهام گرفته از طبیعت است که رفتار جستجوی حیوانات اجتماعی مانند باکتری‌ها و زنبورها را شبیه‌سازی می‌کند تا مسائل پیچیده بهینه‌سازی را حل کند. در این الگوریتم، جمعیتی از عوامل مصنوعی که نمایندگان جستجوگرها هستند، به طور جمعی فضای جستجو را برای یافتن راه‌حل‌های بهینه اکتشاف می‌کنند. این جستجوگرها از طریق تبادل اطلاعات مانند عطرها یا سیگنال‌ها ارتباط برقرار می‌کنند تا تلاش‌های خود را هماهنگ کنند و دانش خود را درباره مناطق مشتاقانه فضای جستجو به اشتراک بگذارند. الگوریتم‌های جستجو معمولاً مکانیسم‌هایی برای اکتشاف و بهره‌برداری شامل می‌شوند تا نیاز به کشف نواحی جدید با بهره‌برداری از راه‌حل‌های پیش‌شناخته تعادل یابند. با تقلید استراتژی‌های همکارانه و تطبیقی جستجوی مشاهده شده در طبیعت، این الگوریتم در زمینه‌های مختلف از جمله رباتیک، مسیریابی شبکه و هوش گله برای یافتن راه‌حل‌های کارآمد برای مسائل پیچیده استفاده می‌شود.

	حرکت دسته جمعی ربات‌ها (Swarm Robotics)		
	گزارش نهایی پروژه		
	نسخه	نام فایل	عنوان فاز
تاریخ گزارش: ۱۴۰۲/۰۸/۲۸	A1.0	LFT-AR-SCS-01/T[۱]/Rev[۳]	نهایی

۱-۵-۵ الگوریتم زنجیره مارکوف مونته کارلو (MCMC)

یک تکنیک محاسباتی قدرتمند که برای نمونه‌برداری و تقریب توزیع‌های احتمال پیچیده استفاده می‌شود. اگرچه به صورت دقیق الگوریتم یک "گله" نمی‌باشد، MCMC اغلب همراه با روش‌های هوش گله‌ای استفاده می‌شود. MCMC با تکرار تولید دنباله‌ای از نمونه‌های تصادفی از یک توزیع هدف عمل می‌کند که هر نمونه به نمونه قبلی وابسته است و یک زنجیره مارکوف ایجاد می‌نماید. از طریق اکتشاف توزیع به این شیوه، MCMC می‌تواند خواص توزیع را مانند میانگین و واریانس تخمین بزند و مناطق با احتمال بالا را شناسایی کند. این الگوریتم دارای کاربردهای گسترده‌ای در زمینه‌های آمار، یادگیری ماشین و استنباط بیزی است و به عنوان یک ابزار ارزشمند به همراه الگوریتم‌های گله‌ای برای حل مسائل پیچیده که نیاز به مدل‌سازی و نمونه‌برداری احتمالاتی دارند، مورد استفاده قرار می‌گیرد.

در نهایت باتوجه به زمان پروژه و همچنین علاقه‌ی تیم به کار بر روی پروژه‌ی موجود در همراه اول، تصمیم بر آن شد که تحقیقات بر روی استفاده از پهنپادها برای تبلیغات و همچنین ساخت تصاویر و انیمیشن‌های بصری ادامه پیدا کند که نمونه‌هایی از آن‌ها در ادامه‌ی گزارش دیده خواهد شد.

 آکادمی علوم و فناوری	حرکت دسته جمعی ربات‌ها (Swarm Robotics)			 مرکز تحقیق و توسعه
	گزارش نهایی پروژه			
	نسخه	نام فایل	عنوان فاز	
تاریخ گزارش: ۱۴۰۲/۰۸/۲۸	A1.0	LFT-AR-SCS-01/T[۱]/Rev[۳]	نهایی	

فصل ۲ – بکارگیری الگوریتم‌ها و نوشتن کد برای Drone Show

در این فصل، برای مسیریابی پهپادها از پیاده‌سازی هوش مرکزی استفاده می‌شود.

۲-۱- روش اول

گرافی در نظر بگیرید از شبکه دکارتی ۳ بعدی که پهپادها در آن نمایش را اجرا می‌کنند. نقاط شروع و مقصد مشخص هستند. ابتدا مسیری با bfs برای ربات اولی پیدا می‌کنیم و سپس برای ربات دوم مسیری را طوری پیدا می‌کنیم که در طی مسیر فاصله حداقلی مشخص شده با ربات اول حفظ شود و سپس به سراغ ربات بعدی می‌رویم.

Time Complexity: $O(\text{Graph}^{\text{drones}})$



۲-۲- روش دوم

از همان گراف قبلی استفاده می‌کنیم اما این بار ربات‌ها را یک به یک فقط یک قدم به جلو می‌بریم به جای اینکه به ترتیب مسیر پیدا کنیم. این راه پیچیدگی بیشتری دارد ولی بهینه‌ترین راه را پیدا می‌کند و از همان Time Complexity قبلی هست.

۲-۳- روش سوم

باید از گراف اولی یک زیردرخت پیدا کنیم که دو راه قبلی بهینه شود. در مرحله اول گراف را compress می‌کنیم. اگر بتوانیم گراف را تبدیل به درختی شامل رئوس مبدا و مقصد کنیم به دلیل یکتا بودن مسیر الگوریتم مسیریابی از $O(\text{Graph})$ می‌شود. برای compress کردن گراف و تبدیل به گرافی سبک‌تر یک راه به این صورت است که هر بار یک راس تصادفی از فضا که تا به حال انتخاب نشده برداریم و نزدیک‌ترین راس به آن را از رئوس انتخابی پیدا کنیم. با بررسی رعایت شدن فاصله حداقلی، آن را به نزدیک‌ترین راس متصل کنیم. این کار را تا جایی ادامه دهیم که همه رئوس مبدا و مقصد در گراف حضور پیدا کنند.

برای پرواز و نمایش تبلیغاتی، برنامه پرواز قبل از پرواز آماده و تست می‌شود. به وسیله انیمیشن ساخته شده در محیطی مانند blender مسیر پرواز و برنامه پرواز هر پهپاد آماده شده است. برای توقف اضطراری و فرود پهپادها در حین برنامه نیاز به الگوریتمی است که با توجه به مکان فعلی پهپادها مسیر را سریع پیدا کرده و با ارتباط با پهپادها برنامه پروازی جدید را ارائه کند.

 آکادمی علوم و فناوری	حرکت دسته جمعی ربات‌ها (Swarm Robotics)			 مرکز تحقیق و توسعه هواداول
	گزارش نهایی پروژه			
	نسخه	نام فایل	عنوان فاز	
تاریخ گزارش: ۱۴۰۲/۰۸/۲۸	A1.0	LFT-AR-SCS-01/T[۱]/Rev[۳]	نهایی	



الگوریتم گفته شده مسیرهایی ارائه می‌دهد که ربات در هر مرحله تنها در یکی از جهت‌های بردارهای مکانی حرکت می‌کند. برای بهتر بودن ظاهر مسیر نیاز به الگوریتم بهینه‌سازی مسیر است که مسیر گسسته الگوریتم را ورودی بگیرد و به پیمایشی پیوسته در فضا تبدیل کند. توضیحات کد برنامه در ادامه آمده است.

در این بخش از کد ابتدا متغیرهای لازم برای کار کردن کد تعریف شده و همچنین تابعی برای محاسبه فاصله‌ی دو نقطه نوشته شده است چون که در هر لحظه برای جلوگیری از برخورد drone در دنیای واقعی به یکدیگر نیاز است که در شبیه‌سازی نیز نقطه‌های ما همواره فاصله‌ی بیشتری از یک مقدار کمینه داشته باشند.

```

1  #include <bits/stdc++.h>
2  using namespace std;
3  const int maxn = 1e3;
4  // structure for saving points
5  struct point3d
6  {
7      double x, y, z;
8      point3d(int x = 0, int y = 0, int z = 0)
9      {
10         this->x = x;
11         this->y = y;
12         this->z = z;
13     }
14     bool operator==(point3d &b)
15     {
16         if (x == b.x && y == b.y && z == b.z)
17             return 1;
18         return 0;
19     }
20     void print()
21     {
22         cout << x << " " << y << " " << z << endl;
23     }
24 } defpoint;
25 // function to calculate distance of two
    points
26 double calcDistance(point3d &p1, point3d
    &p2)
27 {
28     double dx = p2.x - p1.x;
29     double dy = p2.y - p1.y;
30     double dz = p2.z - p1.z;
31     return sqrt(dx * dx + dy * dy + dz * dz);
32 }

```

 آکادمی هوش	حرکت دسته جمعی ربات‌ها (Swarm Robotics)			 مرکز تحقیق و توسعه هوش اول
	گزارش نهایی پروژه			
	نسخه	نام فایل	عنوان فاز	
تاریخ گزارش: ۱۴۰۲/۰۸/۲۸	A1.0	LFT-AR-SCS-01/T[۱]/Rev[۳]	نهایی	



در دومین بخش کد، شروع به دریافت اطلاعات و ورودی‌ها کرده و همچنین مقادیر خروجی داده شده به کد به عنوان ورودی را از لحاظ معتبر بودن چک می‌کند.

```

54 // check that the point is valid in space and
   drones can go to there
55 bool valid(point3d &a)
56 {
57     if (a.x >= X_MIN && a.y >= Y_MIN && a.z >=
       Z_MIN && a.x <= X_MAX && a.y <= Y_MAX &&
       a.z <= Z_MAX)
58         return 1;
59     return 0;
60 }
61
62 int main()
63 {
64     // reading input files
65     ifstream f("config.txt"); // number of drones
   and simulation variables
66     int droneNumber;
67     f >> droneNumber;
68     point3d starting[droneNumber];
69     // data of starting positions and ending
   positions
70     ifstream f2("starting_data.txt");
71     ifstream f3("ending_data.txt");
72

```

بخش سوم کد، فایلی را ایجاد می‌کند تا مسیر تک تک drone‌ها را در این فایل ذخیره کرده و سپس به عنوان خروجی این فایل را به ما می‌دهد که آن‌را به کد بلندر داده و استفاده می‌کنیم. همچنین در این بخش، یک مقدار ثابت به تمامی نقاط وارد می‌کنیم تا مطمئن شویم که تمامی نقاط ما در گراف جا می‌گیرند.



 آکادمی هوش	حرکت دسته جمعی ربات‌ها (Swarm Robotics)			 مرکز تحقیق و توسعه هوش اول
	گزارش نهایی پروژه			
	نسخه	نام فایل	عنوان فاز	
تاریخ گزارش: ۱۴۰۲/۰۸/۲۸	A1.0	LFT-AR-SCS-01/T[۱]/Rev[۳]	نهایی	

```

73 // file to output calculated paths
74 ofstream f4("paths.txt");
75 f4 << droneNumber << endl;
76 drone Drones[droneNumber];
77
78 for (int i = 0; i < droneNumber; i++)
79 {
80     f2 >> Drones[i].starting.x >> Drones[i].
starting.y >> Drones[i].starting.z;
81     f3 >> Drones[i].ending.x >> Drones[i].
ending.y >> Drones[i].ending.z;
82
83     X_MIN = min({X_MIN, Drones[i].starting.x,
Drones[i].ending.x});
84     Y_MIN = min({Y_MIN, Drones[i].starting.y,
Drones[i].ending.y});
85     Z_MIN = min({Z_MIN, Drones[i].starting.z,
Drones[i].ending.z});
86
87     X_MAX = max({X_MAX, Drones[i].starting.
x, Drones[i].ending.x});
88     Y_MAX = max({Y_MAX, Drones[i].starting.
y, Drones[i].ending.y});
89     Z_MAX = max({Z_MAX, Drones[i].starting.
z, Drones[i].ending.z});
90 }
91 // adding offset to be sure graph will be
connected
92 Z_MAX += 20;
93 X_MAX += 20;
94 Y_MAX += 20;
95 cout << "start finding path" << endl;

```

در این بخش الگوریتم اصلی کد ما (bfs) شروع به کار می‌کند. ابتدا تمامی آرایه‌های ممکن آماده شده و پس از آن با یافتن مسیرها، الگوریتم اصلی به پایان می‌رسد.


 آکادمی هاه	حرکت دسته جمعی ربات‌ها (Swarm Robotics)			 مرکز تحقیق و توسعه هاه اولک
	گزارش نهایی پروژه			
	نسخه	نام فایل	عنوان فاز	
تاریخ گزارش: ۱۴۰۲/۰۸/۲۸	A1.0	LFT-AR-SCS-01/T[۱]/Rev[۳]	نهایی	

```

96 // start bfs
97 for (int i = 0; i < droneNumber; i++)
98 {
99     cout << "Finding path for" << i << endl;
100     // resetting arrays
101     for (int k1 = 0; k1 < MAX_CORDINATES;
102          k1++)
103         for (int k2 = 0; k2 < MAX_CORDINATES;
104              k2++)
105             for (int k3 = 0; k3 <
106                  MAX_CORDINATES; k3++)
107             {
108                 dis[k1][k2][k3] = INT_MAX;
109                 par[k1][k2][k3] = defpoint;
110             }
111             queue<point3d> q;
112             q.push(Drones[i].starting);
113             dis[int(Drones[i].starting.x)][int(Drones[i].
114             starting.y)][int(Drones[i].starting.z)] = 1;
115             while (!q.empty())
116             {
117                 point3d x = q.front();
118                 if (x == Drones[i].ending)
119                 {
120                     // detection for finding path and
121                     ending algorithm
122                     break;
123                 }
124             }
125             q.pop();

```

در این بخش و با استفاده از تابعی که در بالا تعریف کردیم فاصله‌ی بین هر دو drone را چک کرده و پس از آن از نزدیک شدن drone‌ها بیشتر از حد مجاز جلوگیری می‌کنیم.


	حرکت دسته جمعی ربات‌ها (Swarm Robotics)		
	گزارش نهایی پروژه		
	نسخه	نام فایل	عنوان فاز
تاریخ گزارش: ۱۴۰۲/۰۸/۲۸	A1.0	LFT-AR-SCS-01/T[۱]/Rev[۳]	نهایی

```

121 // checking neighbors vertex to go
122 for (int j = 0; j < 6; j++)
123 {
124     point3d v(x.x + dx[j], x.y + dy[j], x.z +
125     dz[j]);
126     int ds = dis[int(x.x)][int(x.y)][int(x.z)] +
127     1;
128     if (valid(v) && dis[int(v.x)][int(v.
129     y)][int(v.z)] > ds)
130     {
131         // check collision with previous
132         drone(s) path
133         bool can = 1;
134         for (int k = 0; k < i; k++)
135         {
136             if (Drones[k].path.size() <= ds)
137             {
138                 int ln = Drones[k].path.size() -
139                 1;
140                 if (calcDistance(Drones[k].
141                 path[ln], v) < COLLISION_MIN)
142                 {
143                     can = 0;
144                     break;
145                 }
146             }
147             else if (calcDistance(Drones[k].
148             path[ds], v) < COLLISION_MIN)
149             {
150                 can = 0;
151                 break;
152             }
153         }
154     }
155 }

```

در بخش آخر این کد نیز بعد از استفاده‌ی نهایی از الگوریتم اصلی شروع به سیو کردن مسیرها کرده و در نهایت فایل مسیر تک تک مسیرها را به عنوان خروجی می‌دهد.

	حرکت دسته جمعی ربات‌ها (Swarm Robotics)		
	گزارش نهایی پروژه		
	نسخه	نام فایل	عنوان فاز
	A1.0	LFT-AR-SCS-01/T[۱]/Rev[۳]	نهایی
تاریخ گزارش: ۱۴۰۲/۰۸/۲۸			



```

147         if (can)
148         {
149             // after sure about the vertex
adding to the bfs queue
150             dis[int(v.x)][int(v.y)][int(v.z)] = ds;
151             par[int(v.x)][int(v.y)][int(v.z)] = x;
152             q.push(v);
153         }
154     }
155 }
156 }
157 // reseting queue for next cycle
158 while (!q.empty())
159 {
160     q.pop();
161 }
162 // get path and store it on drone struct
163 point3d x = Drones[i].ending;
164 while (!(x == Drones[i].starting))
165 {
166     Drones[i].path.push_back(x);
167     x = par[int(x.x)][int(x.y)][int(x.z)];
168 }
169 Drones[i].path.push_back(x);
170 reverse(Drones[i].path.begin(), Drones[i].
path.end());
171 // saving the path
172 f4 << Drones[i].path.size() << endl;
173 for (auto x : Drones[i].path)
174 {
175     f4 << x.x << " " << x.y << " " << x.z << endl;
176 }
177 }
178 f4.close();
179 }

```

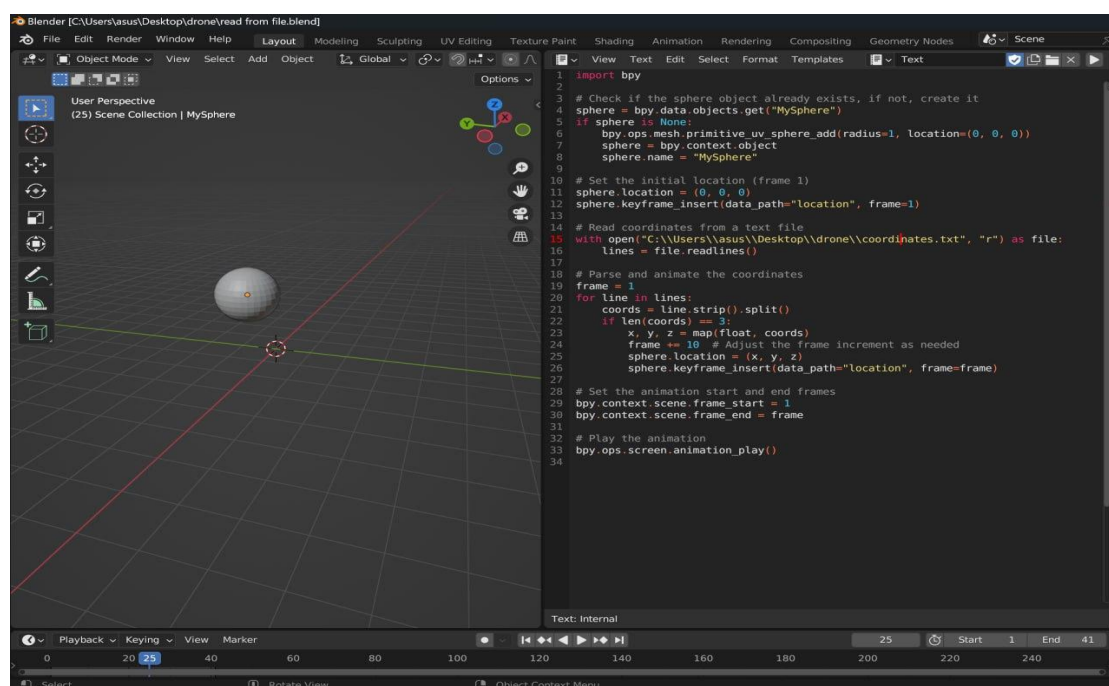
در یک مرحله جلوتر نیز باید وقتی روی هوا متنی نوشتیم متن را عوض کنیم و بعد از آن نیز پهپادها را نیز در جایی دیگر فرود بیاوریم که دقیقاً مانند کد قبلی است که مختصات‌های ابتدا و انتها مسیر را به برنامه می‌دادیم. تنها فرق این دفعه این است که باید چهار فایل نقطه بدهیم، یک فایل مختصات اولیه پهپادها روی زمین، یک فایل متن اول در آسمان، فایل متن بعدی در آسمان و در نهایت هم محل فرود پهپادها روی زمین.

تنها چالشی که پیش آمد این است که تعداد پهپادها باید در دو حالت یکی باشد چون مسیر با توجه به تعداد پهپادها تعیین می‌شود.

 آکادمی ربات‌ها	حرکت دسته جمعی ربات‌ها (Swarm Robotics)			 مرکز تحقیق و توسعه هوادک
	گزارش نهایی پروژه			
	نسخه	نام فایل	عنوان فاز	
تاریخ گزارش: ۱۴۰۲/۰۸/۲۸	A1.0	LFT-AR-SCS-01/T[۱]/Rev[۳]	نهایی	



فصل ۳ – شبیه‌سازی در محیط سه بعدی نرم‌افزار بلندر

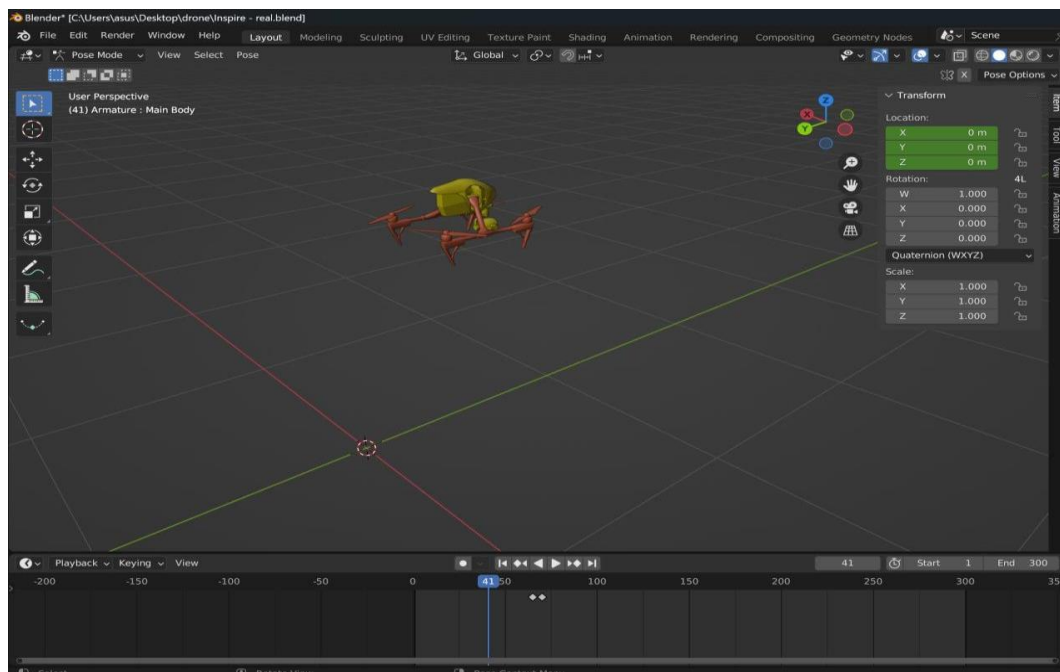
ابتدا در مرحله اول کمی آموزش نرم‌افزار بلندر دیده شده و آشنایی اولیه با این محیط صورت گرفت. انجام کارهایی از قبیل ساختن اجسام، حرکت دادن، نشان دادن انیمیشن و ... بخشی از این قسمت بودند. در مرحله بعد برنامه‌نویسی در نرم‌افزار شروع شد. با استفاده از زبان پایتون یک جسم را طی مختصات هایی که در یه فایل txt قرار داشت حرکت دادیم.



شکل ۱: خواندن مختصات و حرکت دادن یک جسم

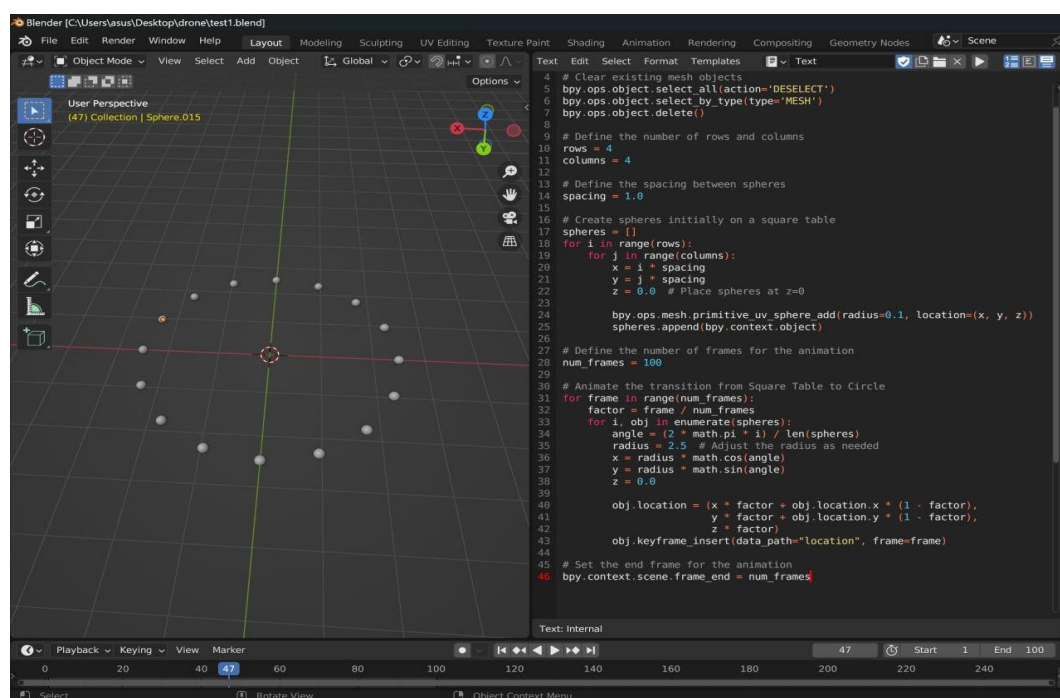
در بخش بعدی، موضوع عوض کردن جسم و جایگزین کردن آن با مدل سه بعدی پهباد انجام شد. در این بخش، یک مدل ساده از حرکت یک پهباد که به دنبال نشانگر موس حرکت می‌کند ساخته شده و حتی تاثیر باد را هم در حرکت آن نشان دادیم.

 آکادمی‌ها	حرکت دسته جمعی ربات‌ها (Swarm Robotics)			 مرکز تحقیق و توسعه هوادک
	گزارش نهایی پروژه			
	نسخه	نام فایل	عنوان فاز	
تاریخ گزارش: ۱۴۰۲/۰۸/۲۸	A1.0	LFT-AR-SCS-01/T[۱]/Rev[۳]	نهایی	





شکل ۲: مدل‌سازی پهباد

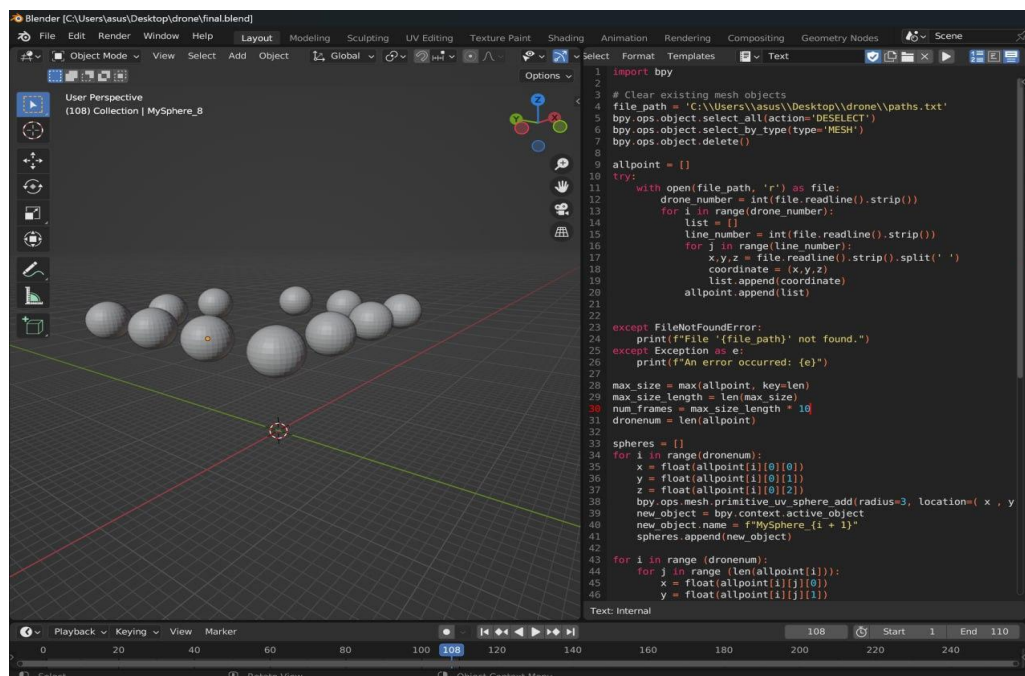
در بخش بعد، تعداد اجسام در شبیه‌سازی را برای واقعی‌تر شدن شبیه‌سازی از ۱ به ۱۰ افزایش دادیم. سپس کد پایتون برای خواندن همه مختصات‌های n جسم و مسیرهای‌شان تعمیم داده شد و کل مختصات‌های مسیر پهبادها را در یک فایل قرار دادیم و از روی آن توانستیم همه اجسام را همزمان با هم حرکت دهیم.



شکل ۳: شبیه‌سازی انیمشین تعدادی جسم و حرکت از دایره به مربع یا معادله دلخواه



 آکادمی‌ها	حرکت دسته جمعی ربات‌ها (Swarm Robotics)			 مرکز تحقیق و توسعه هوادک
	گزارش نهایی پروژه			
	نسخه	نام فایل	عنوان فاز	
تاریخ گزارش: ۱۴۰۲/۰۸/۲۸	A1.0	LFT-AR-SCS-01/T[۱]/Rev[۳]	نهایی	

در نهایت با همه این مراحل به جایی رسیدیم که شبیه‌ساز بلندر فقط یک فایل را که از الگوریتم می‌آید می‌گیرد و شبیه‌سازی را به ما نشان می‌دهد.



شکل ۴: کد نهایی شامل خواندن از فایل، سیو کردن و اجرا کردن همزمان و حرکت دادن اجسام

در آخرین مرحله از شبیه‌سازی‌ها هم کد تولید مختصات‌های نهایی با استفاده از فونت‌های نقطه‌ای و با استفاده از pygame نوشته شد که قابلیت گرفتن هر متن و نمایش آن در شبیه‌ساز را می‌دهد.

 آکادمی هاله	حرکت دسته‌جمعی ربات‌ها (Swarm Robotics)			 مرکز تحقیق و توسعه هاله‌اول
	گزارش نهایی پروژه			
	نسخه	نام فایل	عنوان فاز	
تاریخ گزارش: ۱۴۰۲/۰۸/۲۸	A1.0	LFT-AR-SCS-01/T[۱]/Rev[۳]	نهایی	

```



main.py ×
31 text_rect = text_surface.get_rect()
32
33 # Center the text on the screen
34 text_rect.center = (screen_width // 2, screen_height // 2)
35
36 1 usage
37 def scan_and_print_pixels():
38     num = 0
39     pixels = 170
40     print(screen_width)
41     print(screen_height)
42     for y in range(3, screen_height-4):
43         for x in range(3, screen_width-4):
44             color = 0
45             for i in range(2):
46                 for j in range(2):
47                     pixel_color = screen.get_at((x+i, y+j))
48                     color += pixel_color.r
49             color /= 4
50             if (color != 0):
51                 print(f"color{color}")
52             if ((color > 237) & ((screen_height - y) > 21)):
53                 points.append((x, screen_height - y))

```

while running > if check > with open("C:\\Users\\asus\\Des... > for xy in points

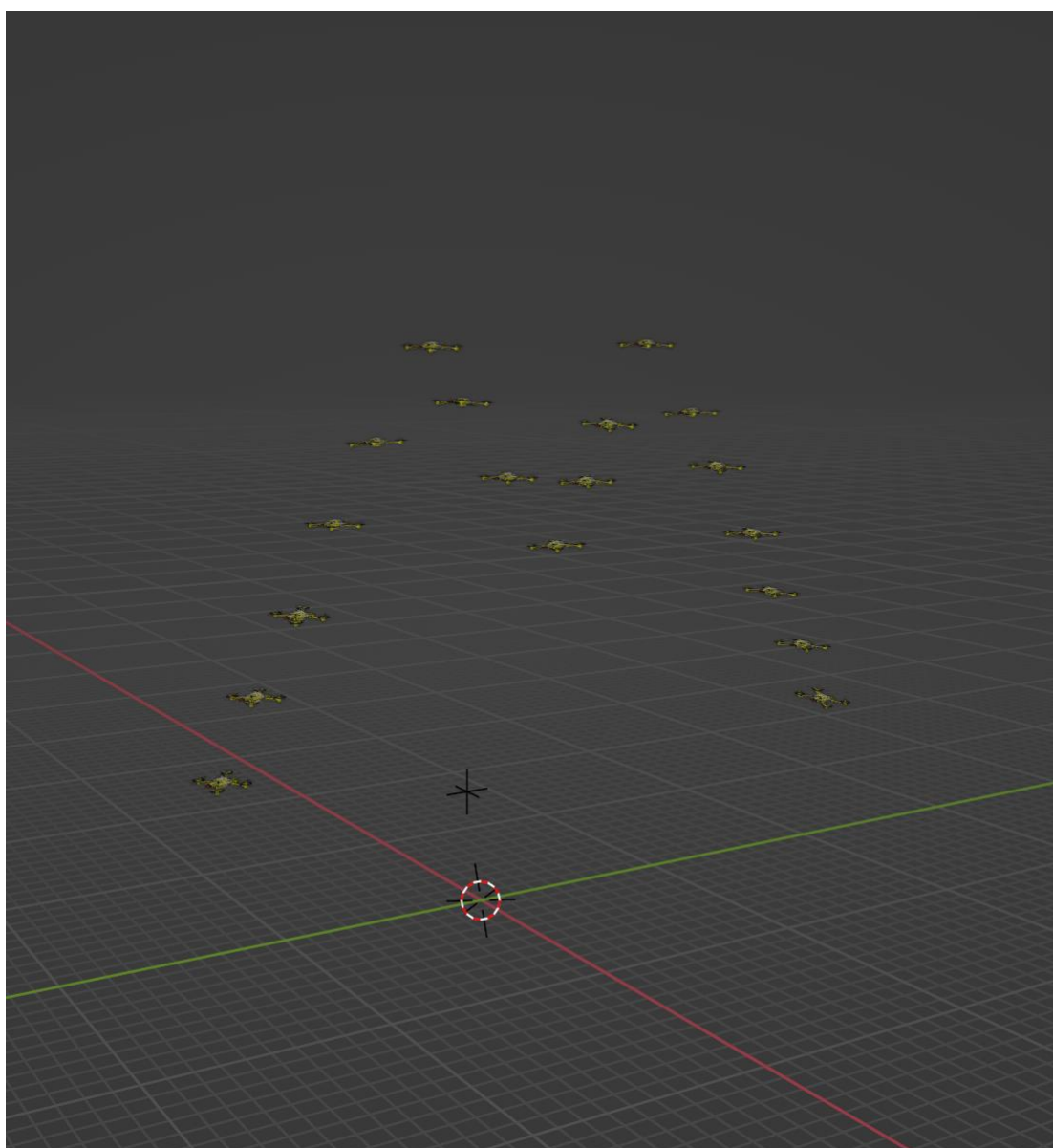
در این الگوریتم ابتدا متن دریافت شده را به صورت زیر روی صفحه نشان می‌دهیم:





 آکادمی‌ها	حرکت دسته جمعی ربات‌ها (Swarm Robotics)			 مرکز تحقیق و توسعه هوادک
	گزارش نهایی پروژه			
	نسخه	نام فایل	عنوان فاز	
تاریخ گزارش: ۱۴۰۲/۰۸/۲۸	A1.0	LFT-AR-SCS-01/T[۱]/Rev[۳]	نهایی	

سپس از پیکسل‌های صفحه rgb می‌خوانیم و میانگین هر ۴ پیکسل را در یک خانه سیو می‌کنیم و اگر مقدار این رنگ از عددی بیشتر بود آن مختصات را به عنوان یه نقطه نهایی از مسیر در نظر می‌گیریم. در نهایت، فایل کانفیگ که تعداد پهپادهای لازم را دارد و در کنار آن دو فایل نقاط شروع و پایان را می‌سازیم. بعد الگوریتم را اجرا می‌کنیم و مسیر را بدست می‌آوریم. در آخر هم در بلندر شبیه‌سازی را مشاهده می‌کنیم.

یک حالت دیگر هم در بلندر شبیه‌سازی شد که در آن مسیر را مشخص کرده و تاثیر باد و نویز را هم در نظر گرفتیم. تمام مراحل در بلندر انجام شد و شکل نهایی خروجی به صورت زیر است:



در آخر از هر دو روش الگوریتم و بلندر به نوشتن متن دلخواه رسیدیم.

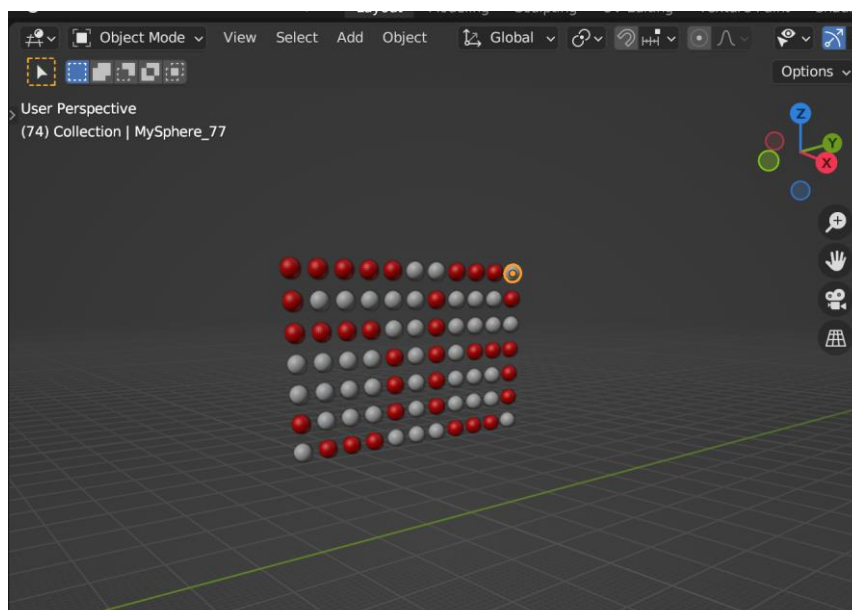
 آکادمی ربات‌ها	حرکت دسته جمعی ربات‌ها (Swarm Robotics)			 مرکز تحقیق و توسعه هواداول
	گزارش نهایی پروژه			
	نسخه	نام فایل	عنوان فاز	
تاریخ گزارش: ۱۴۰۲/۰۸/۲۸	A1.0	LFT-AR-SCS-01/T[۱]/Rev[۳]	نهایی	



در مرحله بعد می‌خواستیم پهباده‌ها در یک شکل ثابت در آسمان یک شبکه درست کنند که با خاموش و روشن شدن چراغ‌های‌شان متن را نمایش دهند. در مرحله اول برای این کار، مختصات‌هایی را که در قسمت قبلی داشتیم بررسی می‌کنیم و ماکزیمم و مینیمم مختصات‌ها را بدست می‌آوریم و یک شبکه دو بعدی مثل دیوار درست می‌کنیم.

```
for i in range(int((ymax-ymin)/3)+1):
    for j in range(int((zmax-zmin)/3)+1):
        x = 21
        y = ymin + 3*i
        z = zmin + 3*j
        list = [x,y,z]
        wallpoints.append(list)
print(wallpoints)

k = 0
for i in range(len(points)):
    for j in range(len(wallpoints)):
        yy = abs(points[i][1] - wallpoints[j][1])
        zz = abs(points[i][2] - wallpoints[j][2])
        if ((yy≤2)&(zz≤2)):
            k = k + 1
```

سپس مختصات‌هایی که برای متن داریم با مختصات‌های دیوار مقایسه می‌کنیم و آن‌ها را روشن می‌نماییم.



 آکادمی علوم و فناوری	حرکت دسته جمعی ربات‌ها (Swarm Robotics)			 مرکز تحقیق و توسعه فرااولد
	گزارش نهایی پروژه			
	نسخه	نام فایل	عنوان فاز	
تاریخ گزارش: ۱۴۰۲/۰۸/۲۸	A1.0	LFT-AR-SCS-01/T[۱]/Rev[۳]	نهایی	

```

for i in range(len(points)):
    for j in range(len(wallpoints)):
        yy = abs(points[i][1] - wallpoints[j][1])
        zz = abs(points[i][2] - wallpoints[j][2])
        if ((yy<=2)&(zz<=2)):
            #code
            matching_sphere = None
            for k, sphere in enumerate(spheres):
                if (round(sphere.location.y) == wallpoints[j][1]) and (round(sphere.location.z) == wallpoints[j][2]):
                    matching_sphere = sphere

            if matching_sphere is not None:
                matching_sphere = None
            for k, sphere in enumerate(spheres):
                if (round(sphere.location.y) == wallpoints[j][1]) and (round(sphere.location.z) == wallpoints[j][2]):
                    matching_sphere = sphere
                    break
            if matching_sphere is not None:

                object_name_to_find = matching_sphere.name

                all_objects = bpy.data.objects

                object_index = next((index for index, obj in enumerate(all_objects) if obj.name == object_name_to_find), -1)

                #code

                mat = bpy.data.materials.new(name="RedMaterial")
                matching_sphere.data.materials.append(mat)

                bpy.context.scene.frame_set(50)
                mat.diffuse_color = (1.0, 1.0, 1.0, 1.0)
                mat.keyframe_insert(data_path="diffuse_color", frame=50)

                bpy.context.scene.frame_set(60)
                mat.diffuse_color = (1, 0, 0, 1.0)
                mat.keyframe_insert(data_path="diffuse_color", frame=51)

```