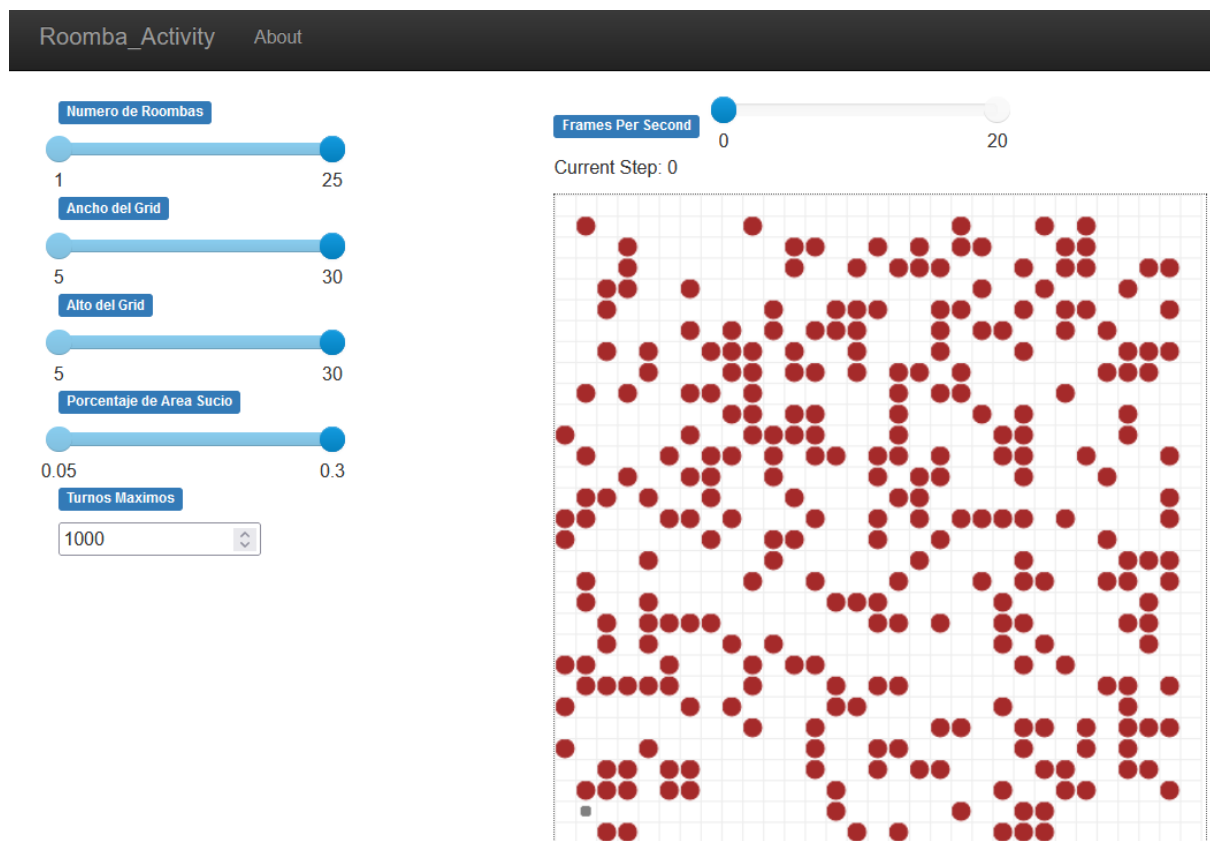
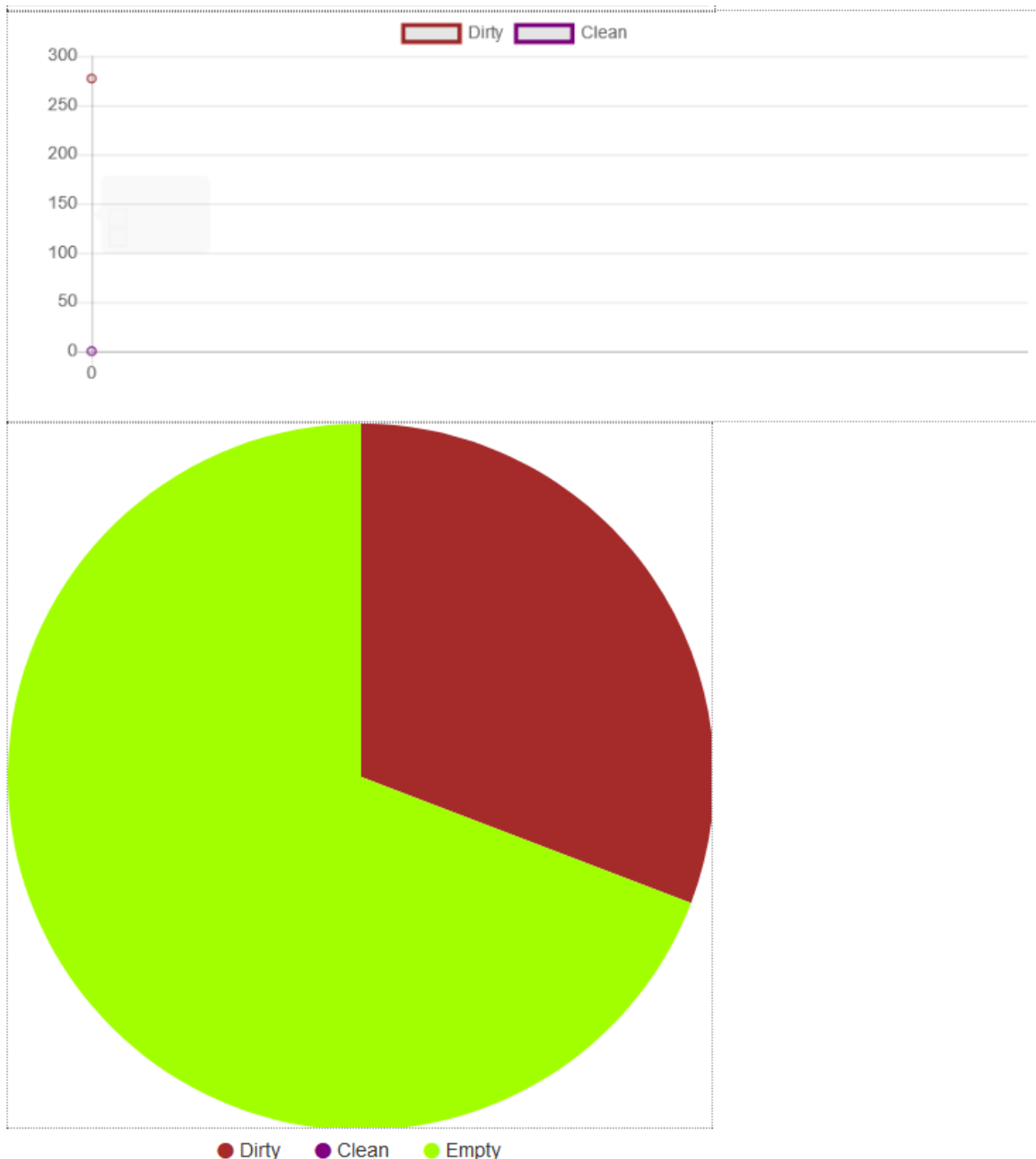


Ignacio Joaquín Moral
A01028470

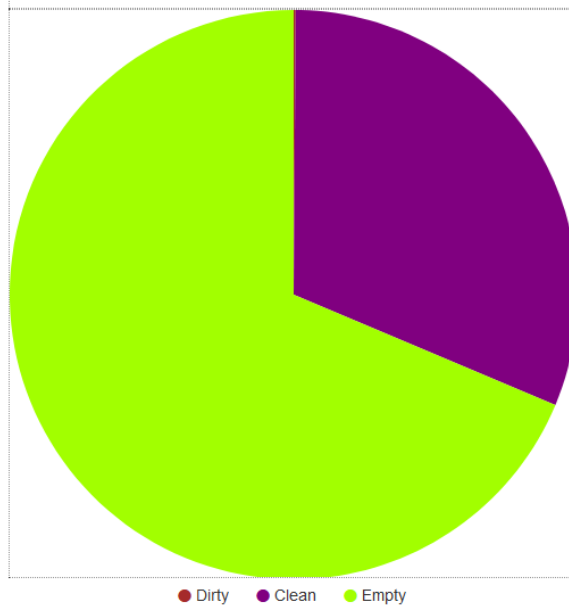
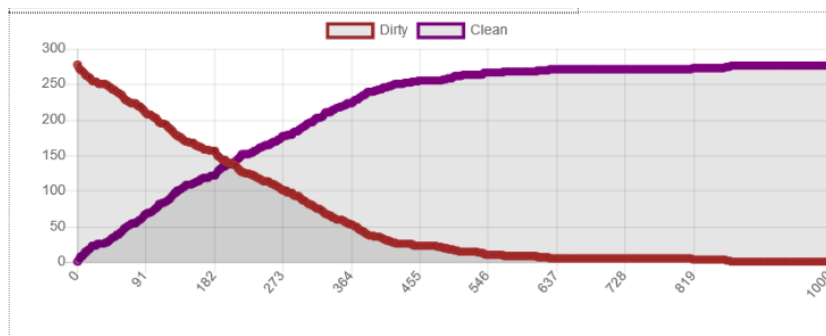
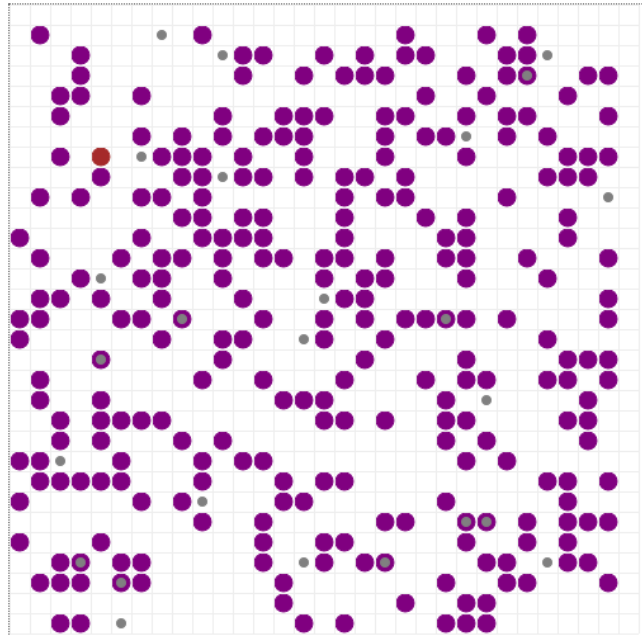
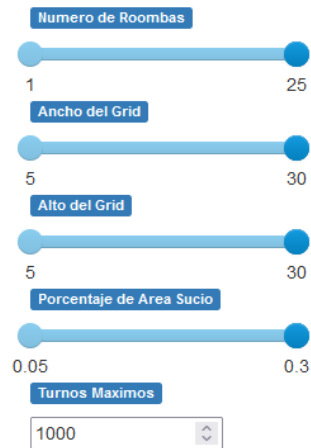
Para esta actividad, se nos pidió generar un modelo en Mesa que simule un grupo de aspiradoras, especialmente las de modelo Roomba, paseando por una sala limpiando. Se pidió que, una vez que un Roomba caiga en una celda sucia, la limpie, y si no, se mueva. Se pidió que el modelo tenga altura y anchura modificable, además de que se pueda escoger la cantidad porcentual de celdas sucias y el número máximo de turnos. Finalmente, se pidió hacer un análisis con las herramientas que ya ofrece Mesa. Para esto, generé un gráfico que se mueve en tiempo real que muestra cómo se eliminan las células sucias





En este modelo, decidí usar Random Activation. Esto es porque consideré el hecho de que algunas aspiradoras intenten limpiar la misma basura, y ninguna de las dos se mueva. Entonces, usando RandomActivation, una aspiradora se activará primero, limpiará la celda, y las otras se podrán mover. También genere que si se limpia el DirtyTile, se elimine del modelo.

Por lo general, el modelo tiende a limpiar mucho al inicio, y se alenta mucho al final. Esto es natural, pues al inicio hay muchas celdas sucias, y el movimiento es generado al azar. Por ende, al inicio que hay muchas celdas, se van a limpiar más seguido. Sin embargo, acercándose al final, se tarda muchos más pasos en limpiar. En algunos casos, no llega a limpiar todo el lugar antes que se termine el número de pasos.



En pruebas, alterando porcentaje de celdas sucias, salen las siguientes tablas. Todas las tablas asumen un grid de 15x15 con 10 roombas disponibles.

Porcentaje Sucio	Celdas Sucias	Turnos	Promedio Turnos
5%	19	285	212.2
	10	225	
	11	187	
	7	197	
	21	167	
10%	25	146	202.4
	29	186	
	24	301	
	15	104	
	21	275	
15%	41	213	201.4
	39	327	
	27	106	
	37	208	
	34	153	
20%	50	347	326
	40	330	
	45	240	
	39	420	
	47	293	
25%	56	265	292
	60	383	
	54	170	
	53	272	
	60	370	
30%	76	249	279
	61	212	
	87	351	
	75	289	
	72	294	

En estas pruebas, se puede ver una distribucion normal muy curiosa. Despega mucho en 20%, y se reduce siguiendo adelante. Esto puede ser debido a que, mientras más lugares sucios haya, más posibilidades de limpiar la celda existan. Además, es menos probable que se limpien las celdas si están muy separadas, ya sea del punto inicial, o del orden general de los agentes de limpieza.

Este modelo tiene muchas posibilidades de mejora. Una posible mejora involucra que se marquen todas las celdas ya visitadas, para evitar que se vayan hacia esas. Aunque pueda llegar al instante en donde un agente se atrape, la idea es evitar que visiten celdas ya visitadas. Otra posibilidad es agregar mas agentes. En teoría, esto llevaría a menos tiempo sin visitar una celda sucia. Aunque no es siempre cierto, sí es cierto que, en misma cantidad

de celdas totales y celdas sucias, se limpian muchas más en menos tiempo. Otra posibilidad podría ser una preferencia por una dirección. En vez de moverse al azar, que tenga un movimiento donde, si se movió el movimiento anterior a la derecha, que prefiera moverse nuevamente a la derecha. Si se movió arriba a la derecha, que prefiera esa dirección general. Y si llega a una pared en movimiento diagonal, que se regrese en la dirección perpendicular. Eso cubriría más parte del tablero. Finalmente, una mejora que se podría implementar es generar búsquedas binarias para encontrar los agentes sucios en cada lado, y puedan las aspiradoras preferir esa posición.