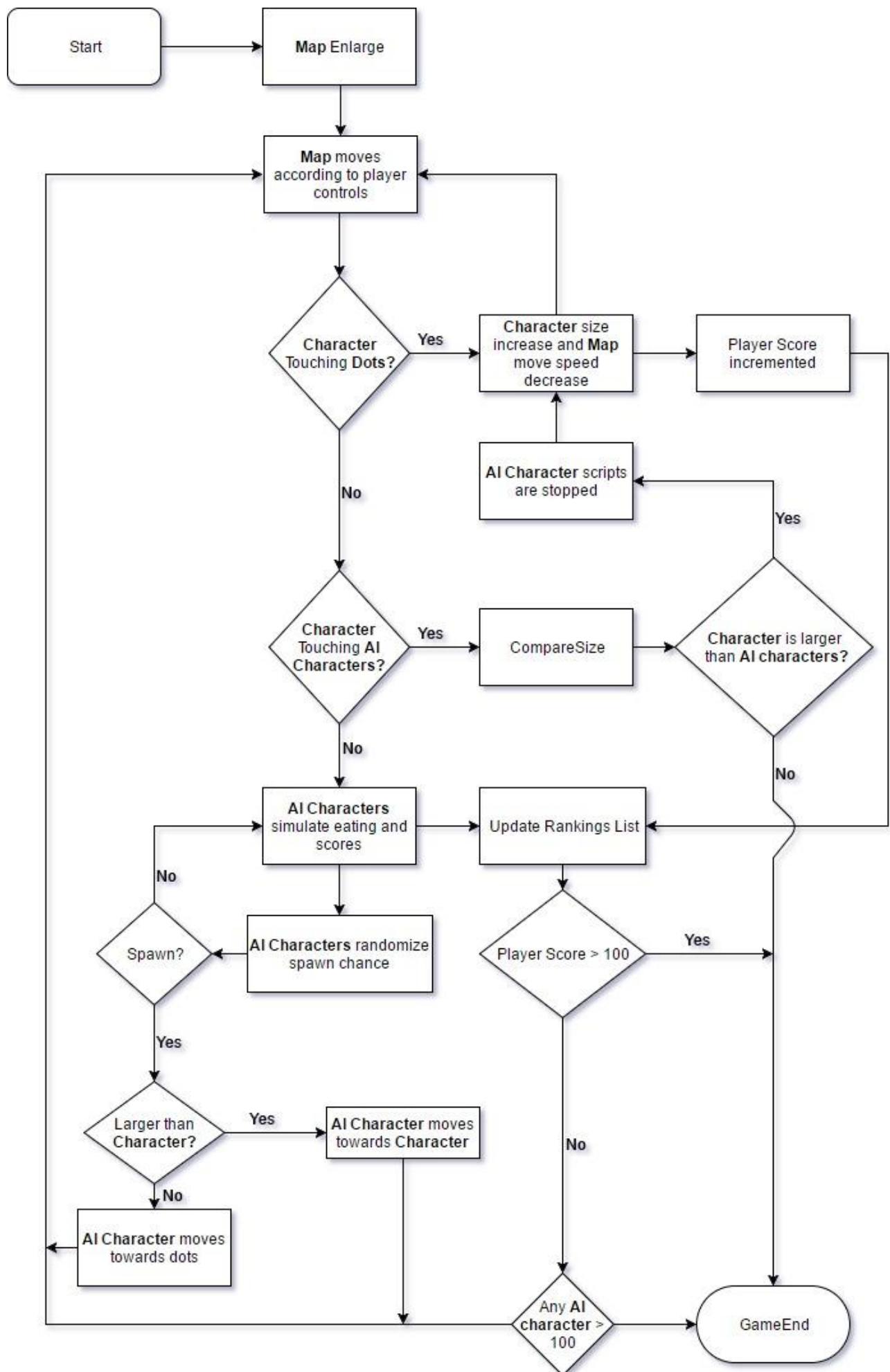


Criterion B: Design

Design of the Solution

Four Basic Objects	Properties	Actions
Map	- a grid costume, enlarged at the beginning of the game	enlarge: a background grid image costume is enlarged to create an area for objects to interact move: since Character remains center-stage, it is the map that is moving that gives the perception that the Character is moving
Player Character	- remains at the center of the stage - can eat dots when in contact with them - gain size when eating the dots - cannot exist beyond Map boundaries	incrementSize: after receive DotEat call or CharacterIdentifier + Eat size is increased decrementSpeed: after receive DotEat call, speed is decreased CompareSize: when in contact, compares size and determines which Character is larger, and eats the smaller one, calls CharacterIdentifier + Eat accordingly SortRankings: Generates and sorts a displayable List that holds the score of all Characters in descending order AtBorders: if Character touches the border of the Map, this is set to 1, and the Map will not move in the direction that allows the Character to be beyond the borders GameLose: if CompareSize determines that Character is smaller or if any AI Character score reaches 100, player loses and game stops GameWin: if Character score reaches 100, player wins and game stops
Dots	- can clone itself and spawn randomly within the stage - disappear when Character or AI Characters are in contact	createClone: a randomizing algorithm spawns a dot at a random location near edges of stage and within the map Eat: Hides the dot and calls DotEat to the

	- cannot exist beyond Map boundaries	character in contact with the dot
AI Characters (Three in Total)	- has a chance to spawn within the stage - engages or disengages Character depending on their sizes - can eat dots when in contact with them - gain size when eating the dots - when in contact with Character, if larger, player loses, if smaller, disappears - cannot exist beyond Map boundaries	incrementSize: after receive DotEat call or CharacterIdentifier + Eat size is increased decrementSpeed: after receive DotEat call, speed is decreased CompareSize: when in contact, compares size and determines which Character is larger, and eats the smaller one, calls CharacterIdentifier + Eat accordingly AppearChance: a randomizing algorithm decides when an AI Character will spawn SimulateEating: if AI Character does not spawn within stage, its score and size are incremented according to a randomizing algorithm Eliminated: A method that stops this character's scripts if CompareSize determines that another character is larger PointTowards: an algorithm that compares the current sizes of Character and his own, and determines whether this character should move towards Character or another dot



These tests need to be carried out several times by playing the game to fulfill relevant requirements of the game.

Success Criteria	Action To Test	Method of Testing/Expected Result
A	Common operating techniques are used to move Player Character	Alpha test by myself to see if controls are intuitive, then have advisor and client beta test and ask for their opinion on the controls
B	The map moves in perception to the Player Character	The Character should be stationary, while the map background should be moving, creating the illusion that the Character is moving with a focused camera
C	Player Character cannot move beyond map	Play the game and attempt moving beyond map
D	Dots spawn randomly on the stage	Play the game with advisor and wait to see if dots are indeed randomly appearing near all four sides of stage
D	Dots cannot exist beyond map	Play the game with advisor and go to edge of map and wait to see if any dots spawn beyond the map
E	Player Character can eat dots, grow in size, and decrease in speed	Play the game multiple times and eat dots and use variable tracking to see if interactions are happening correctly, or physically through observation. Have client play the game to see if the speed and size change is similar to original game.
F, G	AI Characters simulate Eating process and spawn according to algorithm	Play the game with advisor several times, see if AI score is increasing while unseen. Check if AI spawns randomly by observation with client.
F	AI Characters will move towards Player Character if it is bigger and move towards dots if it is smaller	Play the game and wait until an AI character is spawned, if it is larger, see if it is chasing Character, and eat dots until Character is larger than the AI and see if AI will move towards other dots
F, H	All Characters interact correctly while in contact	Play the game and observe if when two characters are in contact, the larger one eliminates the smaller character and gains its size
I	Sorted Scoreboard correctly ranks player and AI	Play the game, eat dots and AI, and see if the scoreboard updates itself correctly
H, I	Game ends correctly	Play the game with client, when a character reaches score 100, or when player Character is eliminated, the script stops.