

Criterion A: Planning

The Scenario

My cousin Kevin plays a game called Agario in his free time. In Agario, the player's character attempts to enlarge itself by eating stationary "dots". A player's character can also eat another player's character when his or hers is larger, and gains the size of the eaten character. The larger the character's size, the more points it has. Players are ranked according to their points.

However, the game is hosted online, meaning that without internet Kevin will be unable to play the game. Since he enjoys travelling, there are many instances where he would be unable to enjoy the game. As the client, he advised me that he would like to see an offline version that "restores the game" and also attempts to improve the game by "developing AI." My computer science teacher has also agreed to be my advisor.

Rationale for Proposed Solution

In the course of my study, I have covered Scratch 2, Visual Studio, and Java. In my scenario, I will be working with graphics and simple object-oriented programming. I have chosen Scratch 2 as my primary programming language to create a simple offline version of Agario for the following:

- Easy visualization of program objects represented through sprites – characters and dots can be created and real-time testing of the game can be done
- Offers a default stage to work with – suitable to create a map for the game
- In Scratch 2, the clone function can create multiple instances of one object – suitable in generating many, random dots to appear in the map
- Simple default control mechanics – Scratch can track cursor position which can help create a movement system that matches the online version
- Offers easy randomizing blocks – helpful in generating randomized dots
- Scratch is free to download, multi-platform, and has an online community to research for tips and help files
- I am familiar with the language, having previously created simple programs with it throughout my studies in Computer Science

Success Criteria

- A. The client can control the main character with common operating techniques like arrow keys or mouse.
- B. The client's character exists at the center and has limited field of view.
- C. A map exists for the client's character to travel around, and cannot travel past the map boundaries.
- D. Visualized "dots" spawn on the map randomly.
- E. The client's character can eat the "dots" by moving towards them and grow in size.
- F. The game includes AI characters that will interact similarly to the main character.
- G. The AI characters will interact even when they are not in the field view of the player.

- H. When an AI character makes contact with the player, their size is compared, and the larger character will eliminate and gain the size of the smaller character. Player loses if he is eliminated.
- I. A score system will keep track of the number of dots/characters each character eats and will rank the characters accordingly. When player reaches 100, player wins. If any AI character reaches 100 first, player loses.

Word Count 492