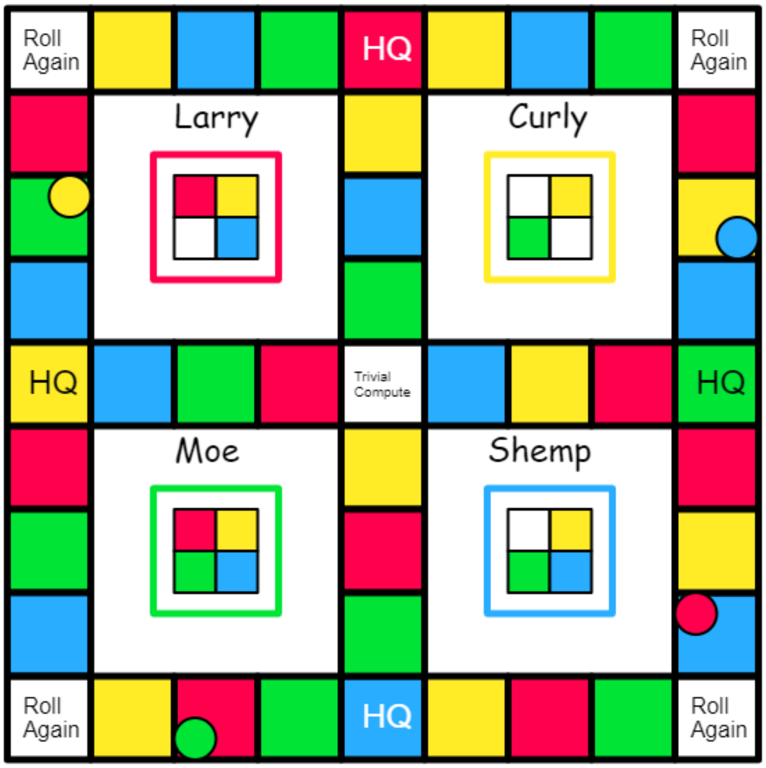
Trivial Compute – a computer game for teaching and learning

On-line, simplified version of Trivial Pursuit.

In this project, you will develop a computer game called Trivial Compute. The rules are very similar to the classic Hasbro game, Trivial Pursuit, with several simplifications. We recommend that you purchase or borrow one of the box games, and spend some time playing it to become familiar with the rules. Here is a video explaining the rules for classic Trivial Pursuit: https://www.youtube.com/watch?v=4VfDkkXefQY&pp=0gcJCdgAo7VqN5tD

The main simplifications in Trivial Compute are the shape of the board, the number of categories, the shape of the tokens, and the mechanism for asking and answering questions. In Trivial Compute, the board is arranged in a square, with nine squares on each side. Spokes emanate from a center square and meet the sides in "category headquarters" squares. The number of categories has been reduced to four accordingly. There should be some way to indicate the current location of each player. The score for each player should be shown on the user interface.



The game can be played by one to four players. The above diagram is a suggestion for the look of the GUI. Trivial Compute will provide in its user interface a random number generator such as six-sided die.

Instead of boxes of cards containing the questions and answers, Trivial Compute keeps the questions and answers in a data repository on the server.

Rules of Play

The order of players signing on to the game will be the order of play.

Before the game starts, the players will need to select the four categories of questions and assign the colors. The names of the categories and assigned colors should then be displayed on the screen.

All players start in the center square. When it is a player’s turn, they will roll the die. Trivial Compute will indicate valid destination squares based on the current location and the amount on the die. There may be as few as two or as many as eight valid destinations. The player will select one of the destination locations.

- If it is a “roll again” square, the player repeats this process.
- If the destination selected is a colored square, Trivial Compute will display a question in a text box. The category of the question is determined by the color of the square. If the player answers correctly, they get another turn. If they are incorrect, they lose their turn, and play continues with the next player.
- If the destination is a headquarters square, and the player answers correctly, the player is awarded a colored “chip.” If they answer incorrectly, they must move away from the headquarters square in the next move before returning in subsequent moves.
- If the destination is the center square, Trivial Compute will display a dialog asking for a category.
    - If the player who lands on the center square has been awarded all four colored chips, then the opponents will decide on the category of the question to be posed to the player. If they answer correctly, they win the game.
    - If they do not have all four chips, that player will select the category of the question they must answer.
    - If the player answers incorrectly, they must move away from the center square in the next move before returning in subsequent moves.

Implementation

The implementation of the game is a web app. The user interface is in a browser, but the control of the game is in a server. The questions should be stored in some sort of repository such as a database, spreadsheet, or flat files. All players in a game will play on the same computer browser. There should be some sort of visual display to indicate the names of the players and who the current player is. The server should be able to support multiple games at once, each game in a separate browser.

Trivial Compute is a simplified version of the classic Trivial Pursuit. As the game is played, questions should be displayed in a window on the screen. The player will answer the question orally. Then the players will push a button to reveal the correct answer. The opponents will decide whether the player answered correctly or not and will indicate the decision to the game. If correct, the player rolls again. If incorrect, the player loses their turn and the next player takes their turn. For the center square, either the player or the opponents will need to select the category of the question to be asked, so you will need a dialog for this.

In this computerized version of the game, questions and answers will be textual (as in Trivial Pursuit). Some questions may have an accompanying media window to display a picture, an audio clip or a video clip. That is, you might show a picture and ask a question about it. Or you might play a sound recording or video and ask about it.

The Question Editor

Trivial Compute is intended to be a game for teaching and learning. We expect that this application will be used by elementary and high school teachers as a way of quizzing their students. Thus, you will need to provide a convenient way for teachers to create categories and questions. There should be a user's manual for teachers to guide them on the creation of the question-and-answer sets. The category/question editor should be implemented as a separate stand-alone application.

Incremental Development

We recommend that you implement this using an incremental approach.

- The first increment should be a skeletal system consisting of a rudimentary client, server and data repository. You should demonstrate the ability of the subsystems to communicate with each other. No graphical user interface is necessary in the skeletal increment. You may limit the number of sessions to one in this increment. You should also develop the question editor in the skeletal increment.
- The next delivery is the minimal system. In this increment the users should be able to play a game following the game rules. The user interface can be text based. The questions and answers may be text based in this version. The system should support multiple sessions at once.
- In the target system, there should be a full graphical user interface and multimedia questions. There should be a user manual for the question editor.
- You may define additional features in a dream system increment. Implementation of any of the dream features would exceed expectations, resulting in a higher grade.