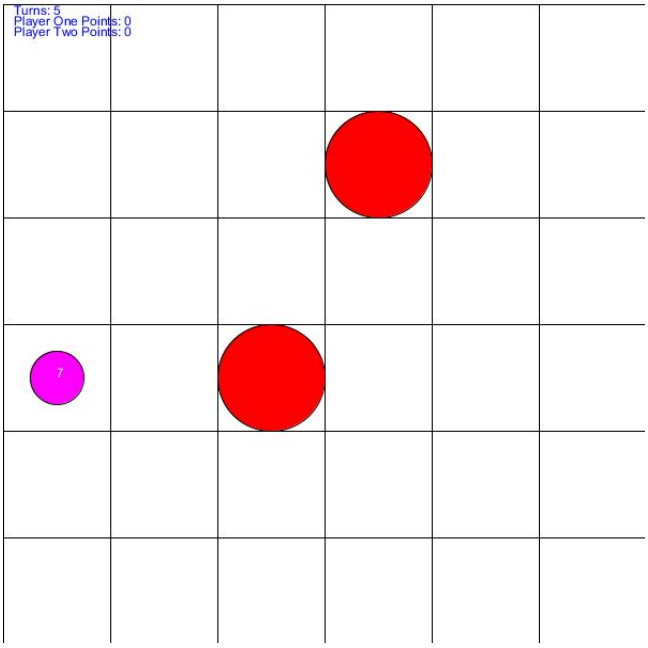
Intro

We are building a **turn-based** game, where two players take turns moving their characters to collect points/power-ups. A very basic version of the game might look like this:



The **required** features are as follows:

- Two players on-screen, represented by shapes or sprites.
- A limited amount of time (or number of moves) for player 1 to move, followed by a limited amount of time (or number of moves) for player 2 to move. Control then reverts back to player 1, and so on.
    - We recommend using different controls for each player, eg WASD for player 1 and IJKL for player 2.
- Points tokens/power-ups, or similar, which appear randomly, and which players can collect to receive points or other bonuses.
- Some win/loss condition.

Additional **suggested** features of the game:

- Decaying points/power-ups - they are less valuable the longer they stay on-screen, and eventually disappear.
- An increased board size (relative to the screenshot above) with multiple power-ups/points available at any given time, increasing the strategy of the game.
- Power-ups which give additional bonuses beyond points (increased move time, double points for a given period of time, paralyse the opponent, etc)

All games must be submitted for grading within the time alloted for the class, in a zip file by email to diarmuid+online@theacademyofcode.com.

Grading Criteria

- Quality of code:
    - Appropriate variable use, names
    - Appropriate use of functions, classes, etc.
    - General readability
    - Computational efficiency
- Quality of presentation:
    - Graphics used (if any)
    - Appropriate formatting of text on-screen
    - Start/end screens
- Originality
    - Any non-suggested features implemented
- Overall quality of gameplay
- DOES IT WORK?? Code which is submitted with errors which prevent it running is unlikely to be the prize-winner.

How to Start

Unsure how to get started? The most important game mechanic is moving the players, and deciding whose turn it is. Start by creating the players on-screen, adding controls to move them, and then figure out how to switch from one player's turn to the other's (we'll talk about this in the session).

Other Notes

Daire's test code (which is **not** available to you) was built with the following major elements (this is NOT an exhaustive list of everything you might include, or an optimal solution - merely a starting point if you aren't sure how to approach the challenge):

- Class "Bites" for the points/power-ups
    - Constructor which assigns a random position and pointValue
    - int for xPos, yPos, pointValue
    - void display() - for displaying the bite
    - void pointsDecay() - for implementing the decaying points game mechanic
- Class "Player" for the players
    - Constructor which assigns a random position
    - int for xPos, yPos, points
    - void display - for displaying the player
    - Ideally this class would also include some of the other player-related code (movement, etc), although in the limited time available this may not be the best use of your time
- ArrayList for Bites (points/power-ups) and Players.
- void drawBites() - for drawing the points/power-ups
- void removeBites() - for removing "decayed" bites
- void detectCollision() - for checking if a Player has collected a Bite
- void keyReleased() - triggers once each time a player hits a key
    - "Decays" the Bites
    - Decides at random whether to add another Bite to the screen
    - Checks whose turn it is
    - Triggers the movement code
- boolean move()
    - Daire decided to return a *boolean* each time a player tries to move. This way he avoided triggering the pointDecay() function, reducing moves left and adding new Bites if the move wasn't valid.