**Criterion B - Design – What happens -relationship between objects and methods.**

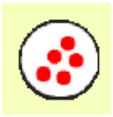
**Design of the Solution**

Three basic objects :

- board
- pot
- sprite

**board** has the following properties
holds 6 pot objects for the player
holds 6 for the computer
holds the sprite
scores for player and computer
two appearances:
one for the playing state with current scores of player and computer
one for the end of game state to show the winner

**pot** has the following properties
position on board
number of seeds in the pot
appearance of number of seeds in that position
for example the pot below holds 5 seeds



**sprite** has the following properties
current position on board
This changes as the game progresses and the sprite moves from pot to pot as a human hand would do in the board game when distributing the seeds.

**pot actions**

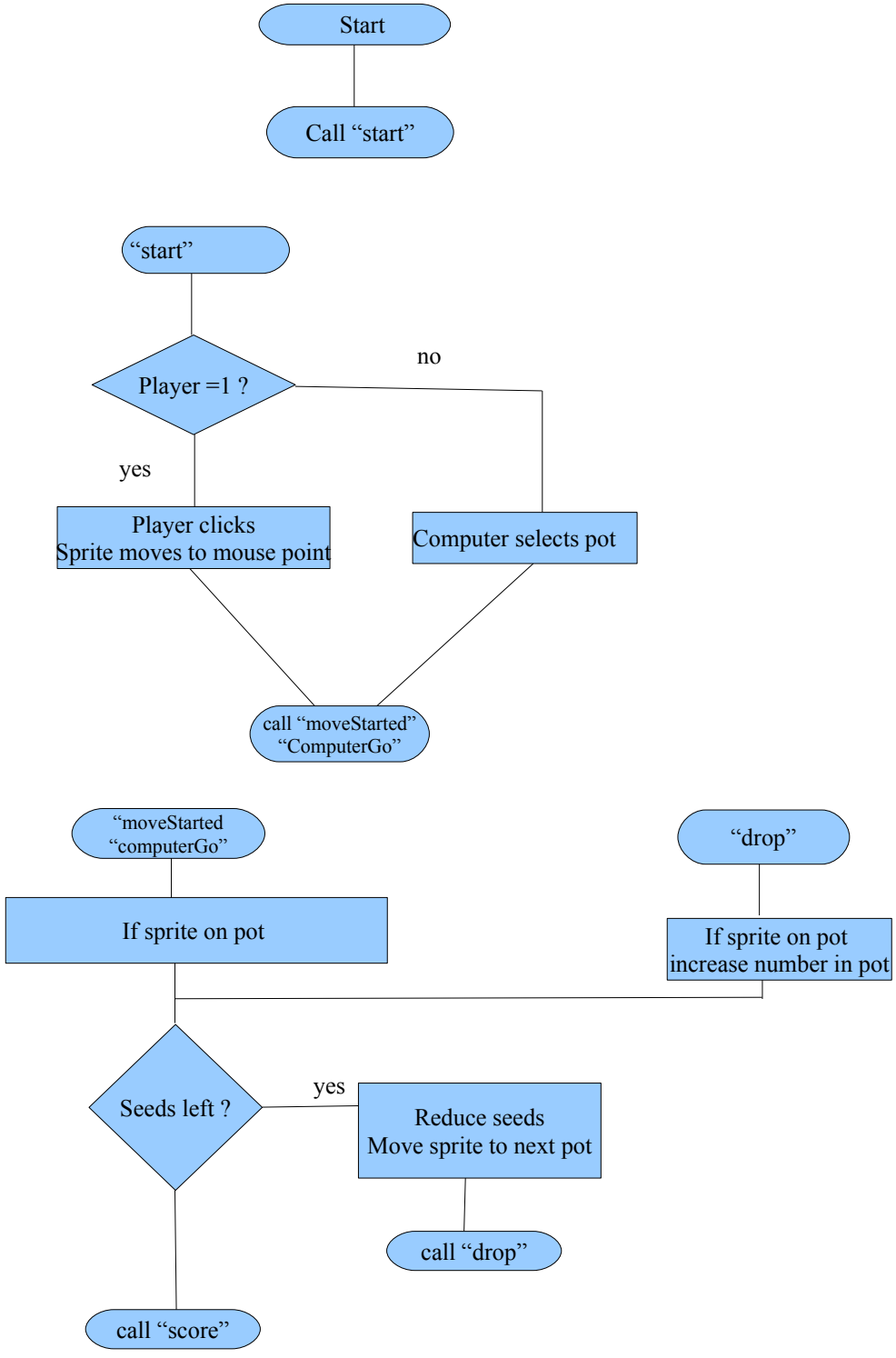
**player:** selected "moveStarted" empties pot when clicked

**computer:** selected by algorithm "computerGo" empties pot

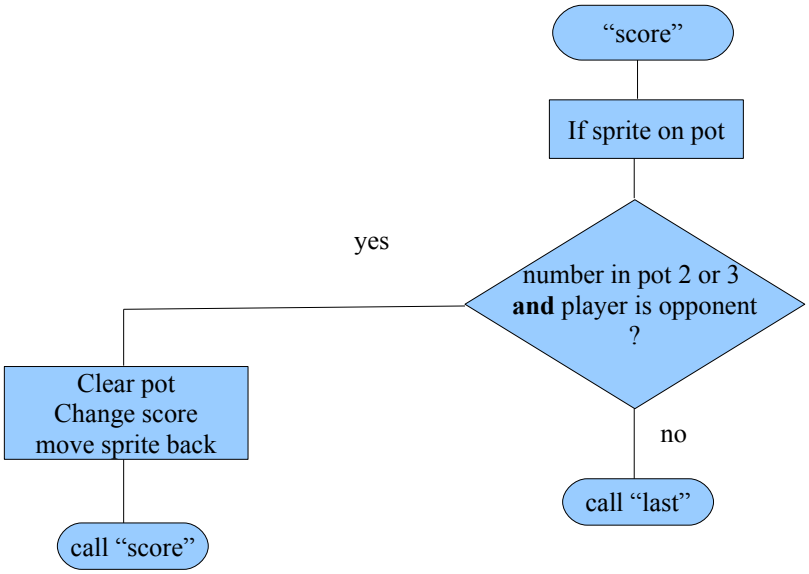
These methods are the same for **player** and **computer** pots – see below for details
"drop" – seeds added to pot
"score" – check if the move results in a score
"start" - become visible with costume 4 seeds in correct place on board.
"gameOver"- hide

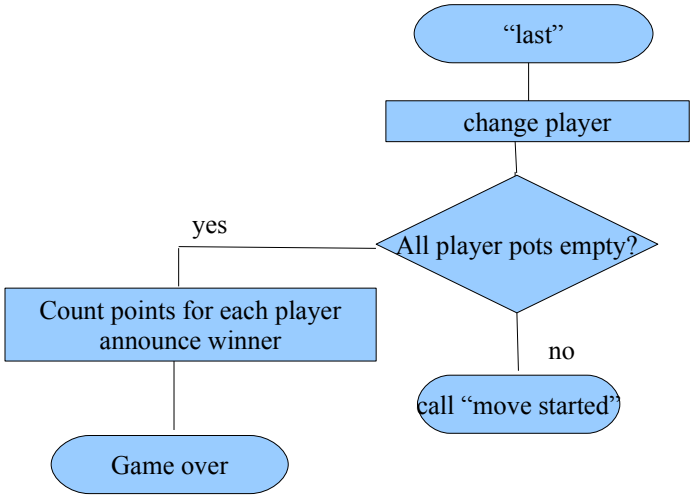
**sprite actions**
"last" - swap between **player** and **computer** (or vice versa)
"move" – go to next pot anti-clockwise
"move back" – go to next pot clockwise
"moveStarted" - if **computer** turn then calculates best move

**board actions**
"gameOver" – changes costume colour and displays text with name of winner.


Start
Call “start”
“start”
Player =1 ?
no
yes
Player clicks
Sprite moves to mouse point
Computer selects pot
call “moveStarted”
“ComputerGo”
“moveStarted
“computerGo”
If sprite on pot
“drop”
If sprite on pot
increase number in pot
Seeds left ?
yes
Reduce seeds
Move sprite to next pot
call “drop”
call “score”

2


"score"
If sprite on pot
yes
number in pot 2 or 3 **and** player is opponent ?
Clear pot
Change score
move sprite back
call "score"
no
call "last"


"last"
change player
yes
All player pots empty?
Count points for each player
announce winner
Game over
no
call "move started"

These tests need to be carried out by playing the game several times and adopting an appropriate strategy.

| Action to test | Method of testing - result |
|---|---|
| Player chosen is random for the first time. | Play multiple times and check that player or computer occurs evenly |
| Player changes after each move | Try waiting and not going to see if computer goes twice in succession. Try to go when not player turn. |
| Clicking on pot activates seed distribution | Yes – every time |
| Check that pots change appearance correctly when seed content changes. | Note changes for diverse number of seeds. Check with array holding number of seeds in each pot. |
| Check number of seeds in the pot is correct after each move | Count manually as the seeds are distributed. |
| Check that the pots are "taken" correctly. | Observe every time computer takes pots. Arrange player to take seeds from one pot only, two and three successive and in particular watch situation where the turn goes into the opposite row. |
| Check that the scores are correctly applied after pots are taken | Count and compare. Add a script to display the number of seeds in each pot when space bar pressed. |
| Check that the number of seeds on the board is correct. | Add the two scores and the total number in pots and check this is 48 |
| Does the computer algorithm for best choice work? | This can be seen from multiple plays and also comparing pot chosen with the other possibilities. |
| The game ends when either player or computer has no seeds left | Check the number visually and also in the array of seeds in pots at the end of each game. |
| Correct winner allocated and displayed | See for each game. |

This criterion was awarded 6 marks.

The record of tasks form is used and indicates the development of the solution clearly.

The designs and test plan are thorough and give a clear indication of how the product was developed including the relationship between the objects and their methods / actions and the way these connect. There is an indication of the techniques and algorithmic thinking used to develop the product (criterion C) plus the extensibility of the product (criterion D).