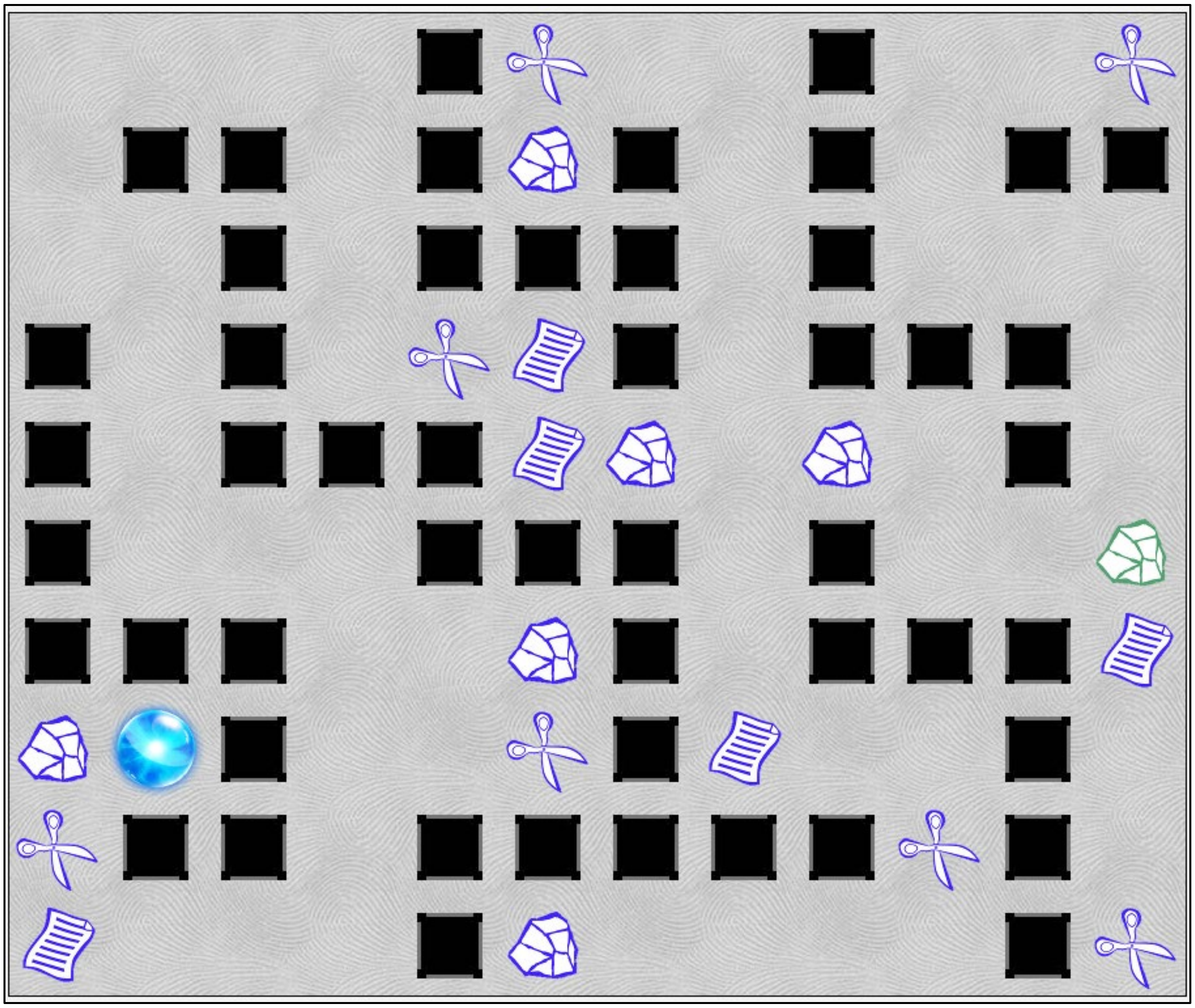
420-141-VA – Game Programming 1

Assignment 1 – *Amazing Rock, Paper, Scissors Maze!*

Worth: 10%
Due: Thursday, September 23th, end of day

Topics

- Implement a core mechanic in a game with the Greenfoot Game Engine
- Programming with Stride (recommended, but converting to Java is also accepted)



"*Amazing Rock, Paper, Scissors Maze!"* is a puzzle action adventure game where the player uses the "Rock, Paper, Scissors" mechanic in order to reach the end of a maze. The Greenfoot project provided implements features such as keyboard movement, level layout and collisions with walls. You will have to implement the Rock, Paper, Scissors mechanic and add an additional feature of your choice to complete the game.

Core Mechanic [5 points]

In a *"rock, paper, scissors"* duel, the rock crushes scissors, scissors cut paper and paper wraps the rock. Anything else would result in a tie. In this game, anytime the player collides with an item, it either wins the duel and destroys the item, or it the item wins, the player morphs into that item.

In the Greenfoot project provided, you must implement the method described below in each class and invoke them every time step:

RockPlayer – crush method	**PaperPlayer – cover method**	**ScissorsPlayer – cut method**
• It detects if the Rock Player intersects with Scissors, removes these scissors from the world and plays the crush.wav sound.	• It detects if the Paper Player intersects with a Rock, removes this rock from the world and plays the cover.wav sound.	• It detects if the Scissors Player intersects with a Paper, removes this paper from the world and plays the cut.wav sound.
Rock – crush method	**Paper – cover method**	**Scissors – cut method**
• It removes intersecting ScissorsPlayer from the world, replaces it by a RockPlayer at the same position and plays the crush.wav sound. • It also removes itself from the World when collision occurs.	• It removes intersecting ScissorsPlayer from the world, replaces it by a RockPlayer at the same position and plays the crush.wav sound. • It also removes itself from the World when collision occurs.	• It removes intersecting PaperPlayer from the world, replaces it by a ScissorsPlayer at the same position and plays the cut.wav sound. • It also removes itself from the World when collision occurs.

Game menus [2 points]

To make the game ready for production, you must also provide a start screen and victory screen. For both tasks, you can create all elements in an image editor and use this image as your world background.

<u>Start World</u>

It contains name of the game (feel free to change the name if you have a better one), and a mention that the game is made in Game Programming 1. It also includes the Vanier College logo. The game will start on this screen and transition to the MazeWorld when you press the Space Bar.

<u>Victory World</u>

When the player reaches the goal, the game must transition to a Victory screen. Be creative in this screen to reward the player for winning. Also press space to restart the game.

Additional task, implement one of the following [3 points]
(if you implement more than one task, the maximum marks for them is 3 points)

- Create your own assets:
  - Images for rock, paper, scissors, background
  - Replace all sound effects
- Make it Rock, Paper, Scissors, Lizard, Spock described in The Big Bang Theory (Youtube)
- Make it a 2-Player experience in a symmetrical level where the first player to reach the goal wins. Also one player can also beat the other player.
- Add a condition to win the game: all rocks, papers and scissors items must be destroyed to activate the goal.
- If you have other interesting ideas, verify with instructor it would be significant enough to get 3 marks.

Submission

You can zip your Greenfoot folder and submit it on Omnivox (Assignment 1). The assignment must be done individually. It is not acceptable to copy/paste or paraphrase your classmate's assignment. To collaborate, you can help a classmate debug his code, explain your approach, but never share your work, such practice is considered plagiarism.

Late submissions are not accepted. In any special circumstances where you need additional time to submit the assignment, let the instructor know at least 3 days before the due date.