**Name:** Application

**Description:** Runs a simple character-graphics based game.

**Requirements:**

The program should run a simple side scrolling game that features a player, non-moving objects and moving objects. The game should support save, pause and exit operations. Some statistics must be tracked.

**Classes:**

1. Application
   * Instance variables:

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Purpose** |
| in | BufferedReader | Reads console input |
| game | Game | Holds current game state |

* + Methods:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Parameters** | **Return** | **Description** | **Pre/post-conditions** | **Interaction** | **I/O** |
| main | String[] arg | void | The main function of the application. | None. | Creates a Game object and invokes its play() method. | Sends prompts to user via System.out, reads options via System.in |

1. Game
   * Instance variables:

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Purpose** |
| p | Player | The player’s character. |
| o | Obstacle | Object that holds the obstacles in the map. |
| m | Missile | Object that holds the missiles in the map. |
| map | char[][] | Holds a rendering of the area currently visible to the player. |
| score | int | Holds the players current score. |

* + Methods:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Parameters** | **Return** | **Description** | **Pre/post-conditions** | **Interaction** | **I/O** |
| play | BufferedReader in | boolean | This function handles the operations performed on every turn. | A valid game object must be created. | Uses the methods provided by the Player, Obstacle and Missile objects to manipulate them. | Accepts game controls from the console. Prints the map, score, and instructions to console. |
| refreshMap | -/- | boolean | Renders the visible area using the locations stored in the game objects. | The game must be initialized. After execution, the map[][] field holds a representation of the map. | None | None. |
| printMap | -/- | void | Prints the map. | Map must be refreshed. | None | Prints the map to console. |

1. Game object classes:
   * Instance variables:

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | | **Purpose** |
| ch | char | | The player’s character. |
| loc | **Player** | int | Holds the row number of the player. |
| **Obstacle** | boolean | Holds locations on the map that are occupied. |
| **Missile** |

* + Methods:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Parameters** | **Return** | **Description** | **Pre/post-conditions** | **Interaction** | **I/O** |
| play | BufferedReader in | boolean | This function handles the operations performed on every turn. | A valid game object must be created. | Uses the methods provided by the Player, Obstacle and Missile objects to manipulate them. | Accepts game controls from the console. Prints the map, score, and instructions to console. |
| refreshMap | -/- | boolean | Renders the visible area using the locations stored in the game objects. | The game must be initialized. After execution, the map[][] field holds a representation of the map. | None | None. |
| printMap | -/- | void | Prints the map. | Map must be refreshed. | None | Prints the map to console. |

**Test Descriptions:**

The program can be tested by first starting a new game.

It is then verified that the control options (up, down, stay, pause and exit) are working correctly.

At the same time, the motion of the obstacles and missiles is observed along with the score tracking, and these are verified to work correctly.

Then, the interactions between the player and environment are tested. It is verified that when the player collides with an object, the game ends and the final score is displayed.

Then, quit the game and save, and finally load the saved file. The file loads and the map is displayed as it used to be, verifying that the object serialization/deserialization and the file I/O code segments work correctly.