**Operation Contracts**

Group: Fark Etmez

**Operation:** moveShooter(shooter: Shooter)

**References: Use Cases:** Move Shooter

**Pre-conditions:** Game is in running mode.

**Post-conditions:** The shooter has moved right or left synchronously according to

 user keystroke inputs. ‘shooter.location’ is updated.

**Operation:**  pickAtom()

**References: Use Cases:** Pick Atom

**Pre-conditions:**  The game is in the running mode.

**Post-conditions:**  Initial atom was not displayed on the shooter. The atom that is

 randomly selected by the system was displayed on the shooter.

**Operation:**  rotateShooter()

**References:**  Use Cases: Rotate Shooter

**Pre-conditions:**  The game is in the running mode.

**Post-conditions:**  The initial angle of the shooter was not valid. The angle was

 updated.

**Operation:**  checkTheVerticalLevel(powerup: Powerup, shooter: Shooter)

**Cross References: Use Cases:** Catch Powerup

**Preconditions:**  Powerup is on the game screen and falling.

 Powerup arrives at the vertical location of Shooter.

**Postconditions:** A boolean created isSameVertical

  isSameVertical associated with instance of Powerup

**Operation:** checkTheHorizontalLevel(powerup: Powerup, shooter: Shooter)

**Cross References: Use Cases:** Catch Powerup

**Preconditions:**  Powerup is on the game screen and falling.

 Powerup arrives at the vertical location of Shooter.

**Postconditions:** A boolean created isSameHorizontal

 isSame associated with instance of Powerup

**Operation:**  increasePowerupNumber(powerup: Powerup)

**Cross References: Use Cases:** Catch Powerup

**Preconditions:**  Powerup is caught.

**Postconditions:** powerup.quantity increased by 1.

**Operation:** vanishPowerup(powerup: Powerup)

**Cross References: Use Cases:** Catch Powerup

**Preconditions:**  Powerup is caught.

**Postconditions:**  Powerup.visible became false.

**Operation:**  pickPowerup()

**Cross References: Use Cases:** Pick Powerup

**Preconditions:** Player must have at least one powerup.

**Postconditions:**  powerup.type associated with Shooter.

**Operation:**  triggerShooter(powerup: Powerup)

**Cross References: Use Cases:** Shoot Powerup

**Preconditions:**  Player must have picked the powerup.

**Postconditions:**  powerup.shot became true.

 powerup.location changed.

 powerup.type disassociated with Shooter

**Operation:** triggerShooter(atom: Atom)

**Cross References: Use Cases:** Hit a Molecule

**Preconditions:**  Player needs to have at least one atom of the chosen type.

**Postconditions:** atom.quantity is decreased by 1.

 atom.location has changed.

**Operation:** detectCompound()

**Cross References: Use Cases:** Hit a Molecule

**Preconditions:** An atom has collided with a molecule of the same type.

**Postconditions:** Compound was formed and has disappeared from the screen.

**Operation:**  decreasePowerupNumber(powerup: Powerup)

**Cross References: Use Cases:** Catch Powerup

**Preconditions:**  Powerup is shot.

**Postconditions:**  powerup.quantity decreased by 1.

**Operation:** checkReactionBlockers(powerup: Powerup)

**Cross References: Use Cases:** Shoot Powerup

**Preconditions:** Powerup is on the game screen and moving upward.

**Postconditions:**  A boolean created isAnyReactionBlocker

 isAnyReactionBlocker associated with instance of Powerup

**Operation:** destroyReactionBlockers(powerup: Powerup)

**Cross References: Use Cases:** Shoot Powerup

**Preconditions:** Corresponding type of Reaction Blocker is in the range of

 Powerup

**Postconditions:**  reactionBlocker disassociated with GameScreen

**Operation:**  breakAtom()

**Cross References: Use Cases:** Break Atoms

**Preconditions:**  Blender is selected.

 There is at least 1 sigma or gamma or beta atom.

**Postconditions:**  inventory.Atom.type (selected) decreased by 1.

 inventory.Atom.type increased with corresponding value.

**Operation:**  pauseGame()

**References: Use Cases:** Pause/Resume Game

**Pre-conditions:**  The game is in running mode.

**Post-conditions:**  Game is paused.

**Operation:**  resumeGame()

**References: Use Cases:** Pause/Resume Game

**Pre-conditions:**  The game is paused.

**Post-conditions:**  Game is in running mode.