**Use Cases**

Group: Fark Etmez

**Use Case Name:** Move Shooter

**Use Case ID:** UC-1

**Scope:** KUVid-Phase 1

**Level:** User Goal

**Primary Actor:** Player

**Stakeholders and Interests:**

-Player: Wants to be able to move the shooter in the specified direction without delays, since even a millisecond of latency could mean health point or final score loss for the player.

-Developer: Wants a bug-free game which provides a good user experience.

**Preconditions:**

-Game is in running mode.

**Postconditions:**

-The shooter moved right or left synchronously according to user keystroke inputs.

**Main Success Scenario:**

1. Player pushes keys which allows the shooter to move.
2. Shooter moves left or right according to the key pressed by the player without any delay.
3. Shooter stays at its new place.
4. The shooter is still able to move in defined boundaries of the game.

**Extensions:**

\*a. At any time, System fails.

1. Player restarts the Game.

2a. Player reaches the visible boundaries of the game.

1. Shooter stops at its current location and does not move out of frame.

**Technology and Data Variations List:**

1a. Player uses keyboard to pick atoms

**Frequency of Occurrence:**

-Could be nearly continuous. Considering that objects fall in a random manner, the player will most likely have to move the shooter most part of the game.

**Use Case Name:** Hit a Molecule

**Use Case ID:** UC-2

**Scope:** KUVid-Phase 1

**Level:** User-goal

**Primary Actor:** Player

**Stakeholders and Interests:**

-Player: Wants the atom shooter to shoot an atom in an upward motion by providing an action to hit a falling molecule.

-Developer: Wants the atoms that are shot from the shooter to stay in the horizontalboundaries of the game screen.

**Preconditions:**

1. Game is in running mode.
2. The player has at least one atom of the randomly selected type to be shot.

**Postconditions:**

1. The atom moved up from the end of the atom shooter and started traveling up the game screen.
2. One atom, of whichever type was shot, was decreased from the inventory of the player.
3. The atom that was shot hit the falling molecule.

**Main Success Scenario:**

1. Player triggers the Shooter while an atom is selected.
2. Atom travels upwards through the game screen at a set speed.
3. Atom is traveling within the horizontal position of the compound.
4. Atom reaches the vertical position of the compound.
5. Atom collides with a molecule of the same type.
6. The atom and the molecule create a compound.
7. The newly formed compound disappears from the screen.
8. The player’s score is increased by one.

**Extensions:**

\*a. At any time, System fails.

1. Player restarts the game.

2a. The atom is traveling upwards but also moving horizontally due to the angle of the shooter.

1. The atom moves vertically with the set speed but also has horizontal speed, depending on the angle of the shooter.
2. The atom then may or may not collide with the molecule to form a compound similar with the main success scenario.

\*a At any time, Atom collides with a side border of the game screen.

1. The atom is reflected from the border.

4-5a. The atom does not collide with a molecule of the same type because there is no molecule present.

1. The atom keeps moving up the game screen
2. The atom reaches the ceiling of the game screen and disappears from the game screen and thus the player’s vision.

4-5b. The atom does not collide with a molecule of the same type but collides with a different type of molecule.

1. The atom keeps moving up the game screen.
2. The atom reaches the ceiling of the game screen and disappears from the game screen and thus the player’s vision.

**Special Requirements:**

Player uses a certain keyboard action to shoot atoms.

Some method to decide whether the atom and the molecule indeed collided, i.e. some sort of hit registry or collision detection.

**Frequency of Occurrence:** Nearly continuous, until the player runs out of atoms to shoot.

**Use Case Name:** Rotate Shooter

**Use Case ID:** UC-3

**Scope:** KUVid-Phase 1

**Level:** User-Goal

**Primary Actor:** Player

**Stakeholders and Interests:**

-Player: Wants to rotate the shooter.

**Preconditions:**

-Player specifies the number of each game object, the length unit and molecule structure in the  building mode.

-Player starts the game.

**Postconditions:**

-The location of the atom was updated.

-The angle of the shooter was updated.

**Main Success Scenario:**

1. System displays the shooter perpendicular to the horizon.
2. Player presses either one of the two desired buttons*.*
3. System rotates the shooter in the desired direction by some degrees.

*System repeats step 3 when the player repeats step 2.*

**Extensions:**

\*a. At any time, the game crashes.

1.Player restarts the game.

      2a. When the system displays the shooter 90 degrees rotated to left *(to right)* and the player presses the desired button for rotating to left *(to right)*.

1.System does not change the display of the shooter.

      2b. Player presses any arbitrary button.

1.System does not rotate the shooter. Display of the shooter stays the same.

**Technology and Data Variations List:**

2a. Player uses a keyboard to rotate the shooter.

**Frequency of Occurrence:** Could be nearly continuous.

**Use Case Name:** Pick Atom

**Use Case ID:** UC-4

**Scope:** KUVid-Phase 1

**Level:** User-Goal

**Primary Actor:** Player

**Stakeholders and Interests:**

-Player: Wants to pick an atom to shoot.

**Preconditions:**

-Player specifies the number of each game object, the length unit and molecule structure in the  building mode.

-Game is in running mode.

**Postconditions:**

-Desired atom was displayed on top of the shooter.

**Main Success Scenario:**

1. System displays a random atom on the top of the shooter.
2. Player presses the desired button.
3. System randomly selects atoms.
4. System displays the selected atom on the top of the shooter.

*System repeats steps 3-4 in every step 2.*

**Extensions:**

\*a. At any time, the player is out of atoms.

1.Player can press the “blender” icon.

      \*b. At any time, the game crashes.

1.Player restarts the game.

      2a. Player presses an arbitrary button.

1.System does not change the atom.

**Technology and Data Variations List:**

2a. Player uses a keyboard to pick atoms.

**Frequency of Occurrence:** Could be nearly continuous

**Use Case Name:** Blend Atoms

**Use Case ID:** UC-5

**Scope:** KUVid-Phase 1

**Level:** User-Goal

**Primary Actor:** Player

**Stakeholders and Interests:**

-Player: Wants the blender to blend specific atoms to convert them to a different atom.

**Preconditions:**

1. Game is in running mode.
2. Player picks the blender.
3. There should be enough atoms to blend them together.

**Postconditions:**

1. Number of source atoms decreased.
2. Number of produced molecules increased.

**Main Success Scenario:**

1. Player picks the blender icon from the statistics screen.
2. Player presses the desired button and then respectively types the rank of the source atom and the rank of the atom to be produced.
3. System decreases the number of source atom
4. System increases the number of produced atom

**Extensions:**

\*a. At any time, System fails.

1. Player restarts the game

2a. There isn’t enough number of source atoms to produce the target atom

1. System gives warning

**Frequency of Occurrence:** Depends on the player. Player can do this action as long as there is enough product atom.

**Use Case Name:** Break Atoms

**Use Case ID:** UC-6

**Scope:** KUVid-Phase 1

**Level:** User Goal

**Primary Actor:** Player

**Stakeholders and Interests:**

-Player: Wants to have smaller atoms by breaking bigger atoms. Breaking atoms are crucial since the atoms must be in correct size to be able to collide with corresponding molecules.

**Preconditions:**

-Game is in running mode.

-Player picks the blender.

-Player has at least 1 sigma or gamma or beta atom.

**Postconditions:**

-Number of the source atom has decreased in inventory.

-Number of produced atoms have increased in inventory.

**Main Success Scenario:**

1. Player picks the blender icon from the statistics screen.
2. Player presses the desired button and then respectively types the rank of the source atom and the rank of the atom to be produced.
3. System decreases the number of the source atom.
4. System increases the number of the produced atoms.

**Extensions:**

\*a. At any time, System fails.

1. Player restarts the game

2a. There isn’t any sigma or gamma or beta atoms to be chosen as a source atom.

1. System gives warning

**Frequency of Occurence:** The action can be done as long as there are at least one sigma or gamma or beta atoms to be broken.

**Use Case Name:** Catch Powerup

**Use Case ID:** UC-7

**Scope:** KUVid-Phase 1

**Level:** User-Goal

**Primary Actor:** Player

**Stakeholders and Interests:**

-Player: wants to catch a powerup and save it. The player needs powerups to destroy reaction blockers.

-Developer: wants to drop powerups at appropriate time intervals.

**Preconditions:**

-Powerup is on the game screen and falling.

**Postconditions:**

-Powerup recorded.

-The number of the corresponding Powerup type calculated.

-Powerup vanished.

**Main Success Scenario:**

1. Powerup arrives at the vertical level of Shooter.
2. Shooter arrives at the horizontal level of Powerup.
3. Shooter and Powerup are at the same point.
4. Player catches the powerup when Shooter and Powerup are at the same point.
5. System calculates the number of the corresponding Powerup type.
6. Powerup vanishes.
7. Shooter leaves the place.

**Extensions:**

\*a. At any time, System fails.

1. Player restarts the Game.

2a. Shooter does not arrive at the horizontal level of Powerup

1. Shooter and Powerup do not meet at the same point.
2. Powerup passes beyond the bottom boundaries.

3-4a. Player is not at the same point with Powerup.

1. Powerup passes beyond the bottom boundaries.

6a. Powerup does not vanish.

1. Powerup continues to fall.

**Technology and Data Variations List:**

Player uses a keyboard.

**Frequency of Occurrence:**

By default, there are 20 powerups. If an average time limit of game is ten minutes, frequency of occurrence would be once every thirty seconds approximately.

**Use Case Name:** Pick Powerup

**Use Case ID:** UC-8

**Scope:** KUVid-Phase 1

**Level:** User-Goal

**Primary Actor:** Player

**Stakeholders and Interests:**

-Player: wants to pick a Powerup to shoot it.

**Preconditions:**

-Player must have at least one powerup.

**Postconditions:**

-Powerup was located on the top of Shooter.

**Main Success Scenario:**

1. Player clicks on the powerup.
2. Powerup places on the top of Shooter.

**Extensions:**

\*a. At any time, System fails.

1. Player restarts the Game.

1a. Player clicks on the powerup which Player does not have.

1. System gives a warning to Player to indicate Player does not have the corresponding powerup.

**Technology and Data Variations List:**

Player uses a keyboard.

**Frequency of Occurence:**

By default, there are 20 powerups. If an average time limit of game is ten minutes, frequency of occurence would be once every thirty seconds approximately.

**Use Case Name:** Destroy Reaction Blocker

**Use Case ID:** UC-9

**Scope:** KUVid-Phase 1

**Level:** User-Goal

**Primary Actor:** Player

**Stakeholders and Interests:**

-Player: Wants to shoot a powerup to destroy the corresponding reaction blocker.

**Preconditions:**

-Powerup must be placed on the top of Shooter.

**Postconditions:**

-The number of the Corresponding Powerup type calculated.

-Reaction Blocker and Powerup are vanished.

**Main Success Scenario:**

1. Player triggers the Shooter.
2. Placed Powerup leaves the Shooter.
3. Powerup moves up across the Game Screen.
4. Any corresponding Reaction Blocker enters Powerup’s field.
5. Powerup destroys the reaction blocker.
6. Powerup is vanished.

**Extensions:**

\*a. At any time, System fails.

1. Player restarts the Game.

4-5a. No corresponding Reaction Blocker enters Powerup’s field.

1. Powerup continues to its path.

**Technology and Data Variations List:**

Player uses a keyboard.

**Frequency of Occurrence:**

By default, there are 20 powerups. If an average time limit of the Game is ten minutes, frequency of occurrence would be once every thirty seconds approximately.

**Use Case Name:** Pause/Resume Game

**Use Case ID:** UC-10

**Scope:** KUVid-Phase 1

**Level:** User-Goal

**Primary Actor:** Player

**Stakeholders and Interests:**

Player: Wants to pause the game than successfully continue from the paused moment

**Preconditions:**

1. Game is in running mode.

**Postconditions:**

1. Game was paused/resumed

**Main Success Scenario:**

1. Player started the game and it's in running mode
2. Player presses a button to pause the game
3. Game is paused
4. Player presses a button to resume the game.

**Extensions:**

\*a. At any time, System fails.

1. Player restarts the game

2a. After pressing the resume button the game is still in pause mode

1. Player restarts the game

2b. After pressing the pause button the game is still running

1. Player restarts the game

3a. The player presses the desired button while the game is paused.

3b. The player presses the desired button while the game is paused.

**Frequency of Occurrence:** Depends on the player as long as the system works the player can pause and resume.

**Start Game:** Actor is Player, player triggers the game to start. The game starts with Building Mode.

**Build Game:** Player is the actor. The player can choose game difficulty, and according to the game difficulty fall speed of objects change. The player can specify the number of each game object, unit L and shape of specific molecules.

**Run Game:** Actor is Player, player triggers the game to start Running Mode. It finishes Building mode. Running mode lasts until the end of the game.