Use Cases

This is the document where will right our use cases. Feel free to add your own. If you add one. YOU OWN IT. Please be sure to explain it so that it can be added into the requirements documents.

1. Player starts new game
2. Player Logs in to Facebook
3. Player post High Score to website
4. Player posts score on Facebook
5. Player loses, restarts
6. Player turns off the sound for game play
7. Player posts score on Twitter
8. Timer UI communicates with game manager object to trigger the win state
9. Database Saves score
10. Player clicks matching blocks - score updates, particle effects
11. Collision detection module - looks for blocks at top of screen

# **1. Informal Specification ( Start Game )**

**Actor(s):**Player

**Goal:**Begin Game

**Trigger(s):**Hit Start Button

**Wireframe Reference(s):**

**UC Diagram Reference(s):** 1

**Activity Diagram(s) Reference(s):**

**Sequence Diagram(s) Reference(s):**

**{ACTION NAME}**

Information about the action the user performs.

Player presses Start Button using mouse (Desktop) or finger (Mobile Platform)

**{ACTION NAME FOR SYSTEM}**

Information about the action the system performs.

The system enters Level One of game.

**{USE CASE ENDS}**

Use case ends.

# 

# 

# 

# **2. Informal Specification ( Posting Score to Website )**

# 

**Actor(s):**Player, Player Score Database

**Goal:** System post player’s score to Website Database

**Trigger(s):** End of Game

**Wireframe Reference(s):**

**UC Diagram Reference(s):** 2

**Activity Diagram(s) Reference(s):**

**Sequence Diagram(s) Reference(s):**

**{ACTION NAME}**

Information about the action the user performs.

Player wants to save game score.

Player enters his/her initials after game ends.

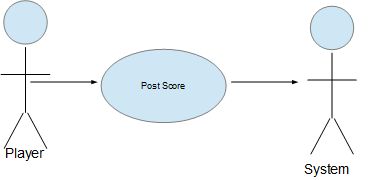
**{ACTION NAME FOR SYSTEM}**

Information about the action the system performs.

Systems stores player’s initial and associated gamescore to website player score database

**{USE CASE ENDS}**

Use case ends.



**3. Informal Specification ( Player Logs Into Facebook )**

**Actor(s):**Player, Website, facebook.com

**Goal:**Player logs in with facebook info

**Trigger(s): Player hits F Button**

**Wireframe Reference(s):**

**UC Diagram Reference(s):** 3

**Activity Diagram(s) Reference(s):**

**Sequence Diagram(s) Reference(s):**

**{ACTION NAME}**

Player accepts prompt to log into Facebook

Player allows posting to Facebook wall

**{ACTION NAME FOR SYSTEM}**

Information about the action the system performs.

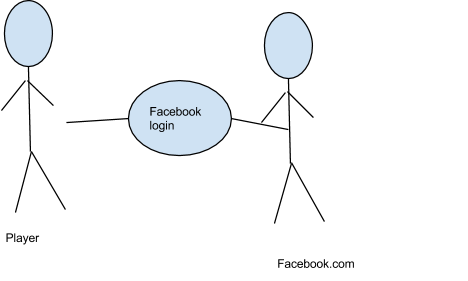
The System prompts the player to log in to Facebook when he visits the game’s website

If the prompt is accepted, then the website goes to Facebook.com

Facebook asks the player if he’d like to login

**{USE CASE ENDS}**

Once player is logged in, the use case ends



# 

# **4. Informal Specification ( Post Score On Facebook )**

**Actor(s):** Player, Facebook

**Goal:** To post player's score on his/her facebook account

**Trigger(s):** Player logs into Facebook

**Wireframe Reference(s):**

**UC Diagram Reference(s):**

**Activity Diagram(s) Reference(s):**

**Sequence Diagram(s) Reference(s):**

**{ACTION NAME}**

Information about the action the user performs

Player logins to his/her Facebook account to save their game score onto Facebook

**{ACTION NAME FOR SYSTEM}**

Information about the action the system performs.

System saves players game score to their Facebook account

**{USE CASE ENDS}**

Use case ends.

System logs off Facebook Account

# 

# 

# **5. Informal Specification Player loses and Restarts**

**{Lose Restart}**

**{Calls the UI and restarts the game}**

Information about the action the system performs.

System call GUI, and pauses game. Prompts player with option to quit or play again

**{Game Restarts}**

Use case ends.

Player chooses quit: goes to main screen

Player chooses restart: Restarts the current level

**Actor(s):** Player, Game ui

**Goal:** Player resarts or quits

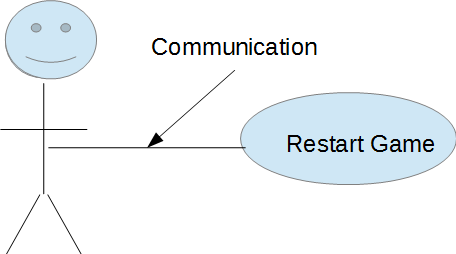
**Trigger(s):** Loss State

**Wireframe Reference(s):**

**UC Diagram Reference(s):**

**Activity Diagram(s) Reference(s):**

**Sequence Diagram(s) Reference(s):**



# **6) Informal Specification ( Sound on/off )**

**Actor(s):** Player

**Goal:** Allow the player choose whether they want to hear the game play sound

**Trigger(s):** button on main menu that allows the user to have control of game play sound

**Wireframe Reference(s):**

**UC Diagram Reference(s):**

**Activity Diagram(s) Reference(s):**

**Sequence Diagram(s) Reference(s):**

# 

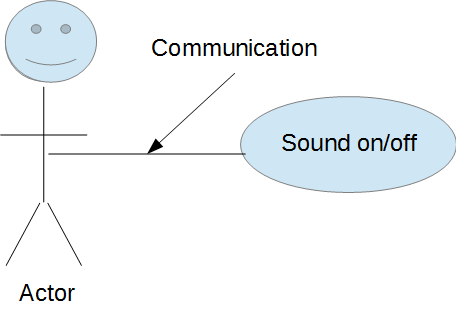
**Turn Sound Off**

**User turns off the volume from the menu screen so no sound is heard during game play.**

Sound interface

**System sets the sound off.**

Use case ends.



# **7) Informal Specification ( Post Score on Twitter )**

**Actor(s):** Player

**Goal:** post score on twitter for others to see

**Trigger(s):** player selects post score to twitter

**Wireframe Reference(s):**

**UC Diagram Reference(s):**

**Activity Diagram(s) Reference(s):**

**Sequence Diagram(s) Reference(s):**

**{ACTION NAME}**

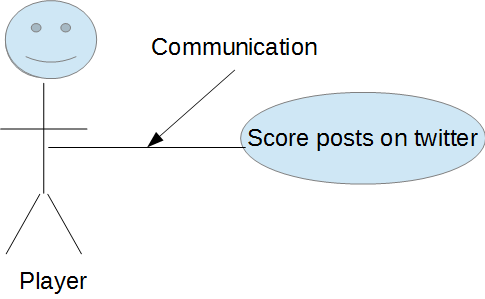
**Player logs into their twitter to tweet out their new score.**

**{ACTION NAME FOR SYSTEM}**

**System allows the user to post score on twitter.**

**{USE CASE ENDS}**

Use case ends.



**Timer Communicates When Player Wins**

# **8) Informal Specification ( Timer Triggers When Condition )**

**Actor(s):** Timer, Game Manager, UI

**Goal:** Player Wins the game, if he survives until the end of the level

**Trigger(s):** Timer object hits 0

**Wireframe Reference(s):**

**UC Diagram Reference(s):**

**Activity Diagram(s) Reference(s):**

**Sequence Diagram(s) Reference(s):**

**{ACTION NAME}**

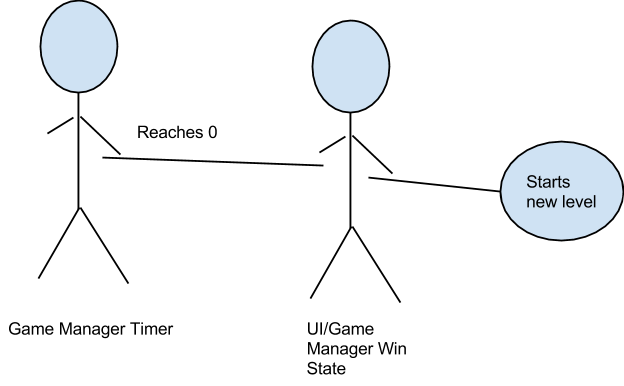
**Game Manager Win Condition Trigger**

**{ACTION NAME FOR SYSTEM}**

**Win Condition on Level timer End**

**{USE CASE ENDS}**

When new level starts



# **9) Informal Specification ( Database Saves Score )**

**Actor(s):** Database; Score Manager

**Goal:** Score manager updates the top score database at the end of every level

**Trigger(s):** Timer object hits 0

**Wireframe Reference(s):**

**UC Diagram Reference(s):**

**Activity Diagram(s) Reference(s):**

**Sequence Diagram(s) Reference(s):**

**{ACTION NAME}**

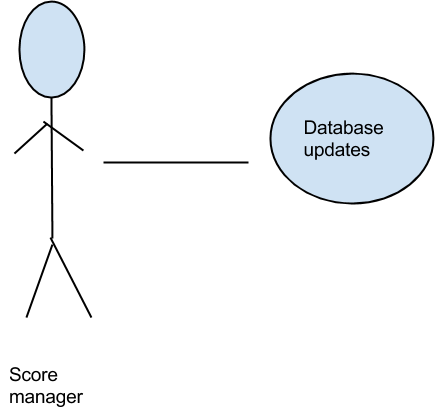
**dATABASE sAVES**

**{ACTION NAME FOR SYSTEM}**

**Database score saving**

**{USE CASE ENDS}**

Ends when new level is updated



**10) Informal Specification ( Player pushes block when there is a match )**

**Actor(s):** Game Manager, Display Manager, Score Manager

**Goal:** Color updates, blocks move down, score updates

**Trigger(s):** player clicks a match

**Wireframe Reference(s):**

**UC Diagram Reference(s):**

**Activity Diagram(s) Reference(s):**

**Sequence Diagram(s) Reference(s):**

**{ACTION NAME}**

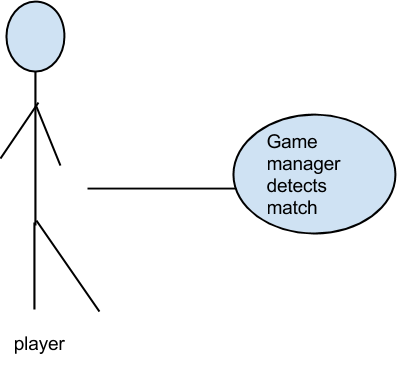
**Player click**

**{ACTION NAME FOR SYSTEM}**

**Match**

**{USE CASE ENDS}**

Ends when row updates



# **Test Steps**

**Scenario #: 1**

Description: Player Presses Start Button

Expected: System begins game from level 1

**Scenario #2**

Player Logs into Facebook

Description: Player logs onto website. The Website fires a pop up alert asking if the player wants to log into Facebook.

Expected: Player can post his Score

**Scenario #: 3**

Description: Player Post High Score by entering His/Her Initials after game ends.

Expected: System saves Player’s score and initials to Website player score database.

**Scenario 4:**  Player post Score to his/her Facebook Account

Description: System posts player’s score to his/her Facebook Account

Expected: Player’s Facebook Account show his/her highest game score.

**Scenario 5:**  Player Restart Game After Failed level

Description: Restart game immediately after failing a level

Expected: Player exits if quit is selected. Player starts same level if restart level is selected

**Scenario 6:**  Toggle Sound On/Offl

Description: Player can Interact with the UI to toggle sound at any point

Expected: Sound goes off or on at the push of a button

**Scenario 7:**  Post Score to Twitter

Description: System posts player’s score to his/her Twitter Account

Expected: Player’s Twitter Account posts the score

**Scenario 8:**  Win Condition Timer

Description: When the timer reaches 0, the level should end

Expected: When Timer runs down, UI should pop up saying that the player won the level

**Scenario 9:**  Database Saving

Description: At a certain interval the score is written to a text file or database

Expected: Player score is updated at least once per session

**Scenario 10:**  Block matching

Description: Player pushes the button, Game manager detects if there is a match

Expected: Blocks get removed when the player clicks onn a match, score updates