Assignment 1

Asteroids

Created by group OP27-G59 for the course
CSE2115 Software Engineering Methods
of the Computer Science curriculum
at the Delft University of Technology

Group members:

Onur Gökmen
Joseph Catlett
Irtaza Hashmi
Helena Westermann
Ceren Uğurlu

CONTENT

1 Functional Requirements

The functional requirements of the game Asteroids are grouped together according to different components of the game: Spaceship, Asteroid, Laser, Account, Score, Sound, Levels, Multiplayer, Themes and Screens. In addition, they are ordered and categorized using the MoSCoW method. If a requirement is yet to be decided, we will use “To Be Decided” or TBD for short.

1.1 SPACESHIP

MUST: Upon start of new game, a Spaceship object is created of fixed size, at the centre of the screen. The tip of the spaceship will be facing upwards. The size of the asteroid is TBD. The spaceship will be drawn using lines initially. Later, we can add an image so the rocket looks nicer. MUST: It moves by pressing the up arrow in the direction the spaceship is pointing, it accelerates at a rate of 0.06 pixels/second until it reaches maximum velocity of 6.5 pixels/second.

MUST: When the W-key is released the spaceship begins to decelerate automatically at a rate of 0.99 pixels/second.

MUST: Pressing the A or D keys rotates the ship left or right with a maximum velocity of 0.05 pixels/second. The longer the player presses the key, the faster the ship will rotate.

MUST: The spaceship fires a laser which is triggered by pressing the spacebar. The spaceship can fire five shots consecutively lasers before taking before reaching an “overheated” state which lasts 0.5 second before the player can fire again.

MUST: A spaceship is created with three lives. The lives are shown at the top left of the game screen.

MUST: If the spaceship touches an asteroid it loses a life. After a life is lost, the number of lives decreases by one and the most right “rocket” disappears from the top left corner, signifying that a player has lost a life.

MUST: If all lives are lost, meaning that the number of lives are zero, the player dies.

SHOULD: If the player loses a life, they should be put back to the centre of the screen. The screen will be 700 by 700 pixels.

SHOULD: When respawned at the centre there should be a period of invincibility lasting for 5 seconds to avoid dying continuously because of crowding asteroids in the centre. At this point the ship will be blinking, hence showing that it is invulnerable and a collision won't kill the player.

COULD: Before starting a new game, the player can choose a different icon/skin for the spaceship from an options menu located on the secondary screen.

WON'T: The user can choose between different types of spaceships before starting a game. Each one will have a unique strength and weakness (e.g Some are faster but weaker, some can shoot more lasers but are larger etc.)

WON'T: The user can upgrade their spaceship by e.g. making it faster, shoot more lasers consecutively, etc.

1.2 ASTEROID

MUST: The asteroid objects have different sizes and attributes. The different sizes are Large, Medium, and Small worth 25, 50 and 100 points respectively. They also have different radiuses. The large one has a radius of 45 pixels, medium one is 35 pixels and the small one is 17 pixels.

MUST: When a new level or game begins, 4 asteroid objects are created at random locations. As level progresses, the number of asteroids = levelNumber + 3.

MUST: When asteroids are hit by a laser, if they are above size small, they are split into two smaller asteroids. For example, if a laser collides with a large asteroid, it splits into two medium asteroids and if a laser collides with a medium asteroid, it splits into two smaller asteroids. If an asteroid is small and is hit by a laser, it disappears.

MUST: The asteroids don't interact with each other, they pass through each other.

MUST: The asteroids move across the screen with a velocity ranging from 0.8 to 2 pixels/millisecond.

SHOULD: An asteroid's velocity is affected by being hit.

WON'T: If you kill the last asteroid of the level, you get bonus points.

WON'T: Asteroid's have an exploding animation once they are hit.

WON'T: Different skins for asteroid's

1.3 LASER

MUST: The laser object has a range and velocity attribute. The range of the laser is 65 pixels and the velocity is 7 pixels/millisecond.

MUST: After the laser travels it's range, it disappears.

MUST: Collision with an asteroid causes the laser to disappear.

MUST: Fired from the centre of the ship, they travel at a constant velocity of 65 pixels/second, with a fixed speed and a direction that depends on the spaceship's head.

SHOULD: Holding the spacebar shoots at most 5 lasers to prevent spamming.

COULD: Holding the spacebar for at least 1.5 seconds powers up the laser instead of firing multiple lasers. This laser is stronger than normal laser and can destroy any asteroid (not just break them into smaller asteroids).

COULD: Power up lasers are different colours to regular lasers.

WON'T: Collisions between lasers and asteroids play a little explosion animation.

WON'T: If you get hit by your own laser (they can loop around the screen) you lose a life.

WON'T: The user can buy faster and stronger lasers with the number of points they have achieved.

WON'T: An inventory where user can change between lasers.

WON'T: Different skins for laser.

1.4 ACCOUNT

MUST: Upon running the program, a user can register as a new user.

MUST: They click the REGISTER button and are brought to a register page.

MUST: the user can type in a username and password.

MUST: The username must be compared to all the usernames in the database to ensure it is available.

MUST: The password must be compared to a set of security criteria (between 8-16 characters long and contains at least 1 number).

MUST: This user and their credentials are saved to a local database.

MUST: In the opening screen the user must log in with their credentials.

MUST: If they enter the correct information, they proceed into the game or a second window.

SHOULD: If they've entered incorrect information, the text fields should flash/turn red.

COULD: The user name is a combination of provided "Clan Names" and user chosen integers.

COULD: The user can be changed at any point.

WON'T: A computer can only register once.

1.5 SCORE

MUST: The score is calculated based on the number of asteroids destroyed. If a user destroys a big asteroid, that's 25 points. A medium asteroid is worth 50 points and the small one is worth 100 points.

MUST: Every score is saved to the local database with the corresponding user account and nickname entered by the user after the game.

MUST: After finishing the game, the user's score is shown on the screen.

MUST: The top 5 high scores are retrieved from the database after each game and the top five high scores are displayed on screen, with their corresponding players' initials.

SHOULD: Smaller asteroids are worth more points than larger asteroids.

COULD: There is a page you can navigate to via the end game screen where you can see every score saved in the database and the nicknames of the players who achieved the scores.

WON'T: The scores are converted into coins (1:1 ratio of pints to coins) and you can use these coins to purchase ship skins/themes/ power ups/ new ships.

WON'T: There will be a bonus point system.

WON'T: When a user supasses the high score, the user will get a notification.

1.6 SOUND

MUST: The background music begins when a new game is started.

SHOULD: Firing Lasers makes a “pew-pew” sound effect.

SHOULD: Hitting Asteroids makes an explosion sound effect.

COULD: Starting a new game, entering a new level, ending a game all have distinct tones or sound effects.

COULD: When a new level is entered the music reflects the change in difficulty and increases in speed.

COULD: Music stops when the character dies.

WON’T: There is sci-fi music played in the all the other screens (login screen, register page, end game screen, score saving screen).

WON’T: The player can mute the music.

WON’T: Different music

1.7 LEVELS

MUST: Maximum number of levels so the game has a stopping point and doesn’t continue infinitely.

SHOULD: When all the asteroids are destroyed, you enter a new level.

SHOULD: Each succeeding level is harder than the previous.

SHOULD: The level number is displayed on the screen before each level is starting

SHOULD: The number of Asteroids increases.

COULD: The highest level reached is shown on the death/end screen.

WON’T: Each level has a unique background and music.

WON’T: The user can skip the next level if they finish a level in a certain period of time.

1.8 MULTIPLAYER

WON’T: Starting the program opens a server connection and establishes a connection to said server.

WON'T: Multiple people can connect to the server and create two player games.

WON'T: When a player starts a game they are automatically paired with another user also trying to start a game at that time.

1.9 THEMES

WON’T: In either the opening window or some secondary window, the user will be shown a button clearly labelled “Select Theme” and be given the option of choosing a theme for the game.

WON'T: Once the theme is chosen, icons, graphics and backgrounds will change to the chosen theme.

WON'T: The theme will be reset to some default every time the program is run.

1.10 SCREENS

MUST: There must be an opening screen to log in with the games title, a username text field, a password text field, a login button and a register button, all placed in the center of the screen, clearly visible to the user.

MUST: There must be a register screen with a username text field, a password text field, a textfield to confirm the password and a button to register the player.

MUST: Once the game has been started, you should see the Asteroids, the Spaceship, the score, the number of lives remaining.

MUST: The game screen should be looped, so if you exit the screen on the right at some y-value, you should re-enter on the left at the same y-value (and the same for all other directions).

MUST: When you die, your score should be displayed on screen with the words “Game Over” above.

MUST: The end screen should provide a keyboard of buttons to type in ones initials. There should be a button labelled “Enter” next to the keyboard to save the score with the nickname chosen by the player.

MUST: Then, the top 5 high scores should be displayed and there should be exit and new game buttons clearly labelled and visible beneath the highest scores.

SHOULD: There should be a secondary screen that should have the options for the themes, an option to turn off sound, and the buttons to start a game or exit the program.

COULD: If the user has got a high score, after dying the words “High Score" should be shown on the screen above there score instead of the words “Game Over.

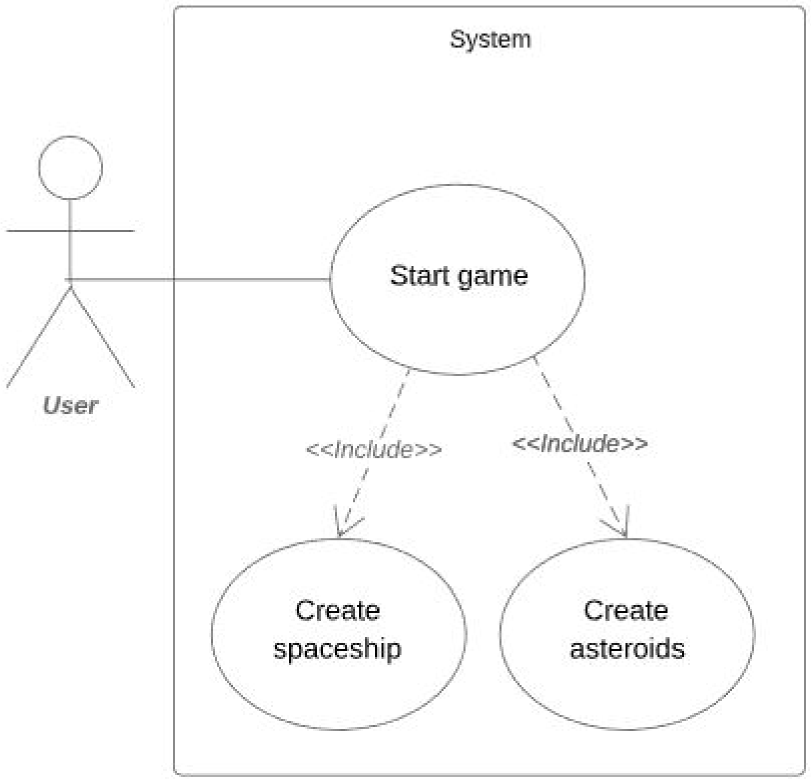
2
Non-Functional Requirements

In addition to the functional requirements, there are a number of constraints that apply to the game development and system.

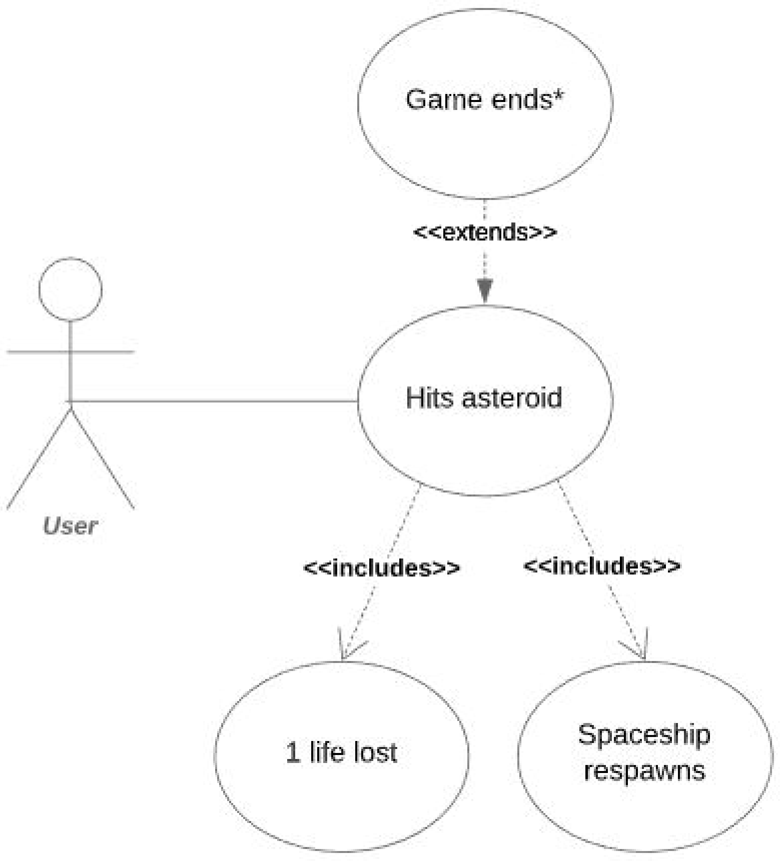
- The authentication should be implemented with the use of SQL and JDBC.
- Java should be used for the game logic.
- JavaFX (FXML) and CSS styling should be used for any windows that do not hold game functions (e.g. login windows, score display, register window etc.).
- Prepared statements should be used to prevent code injection.
- There should be secure username and password storage.
- The password is saved in a hashed format.
- The game should be non-exploitable (e.g. has no cheat codes).
- The game should have intuitive game controls.
- The code should contain clearly distinguished classes and there is cohesion in-between them.
- Have a working prototype by December 6th with 70% code coverage

3
Modelling Use Cases

Use Case: Start game
Author: HW
Date: 27/11/2019
Description: After the user logs in, the system initiates the game at level 1. It creates the spaceship with 3 lives and places it in the centre of the screen. Additionally, the system creates 4 asteroid objects at random locations and assigns them a random velocity between 0.8 and 2 pixels/millisecond.



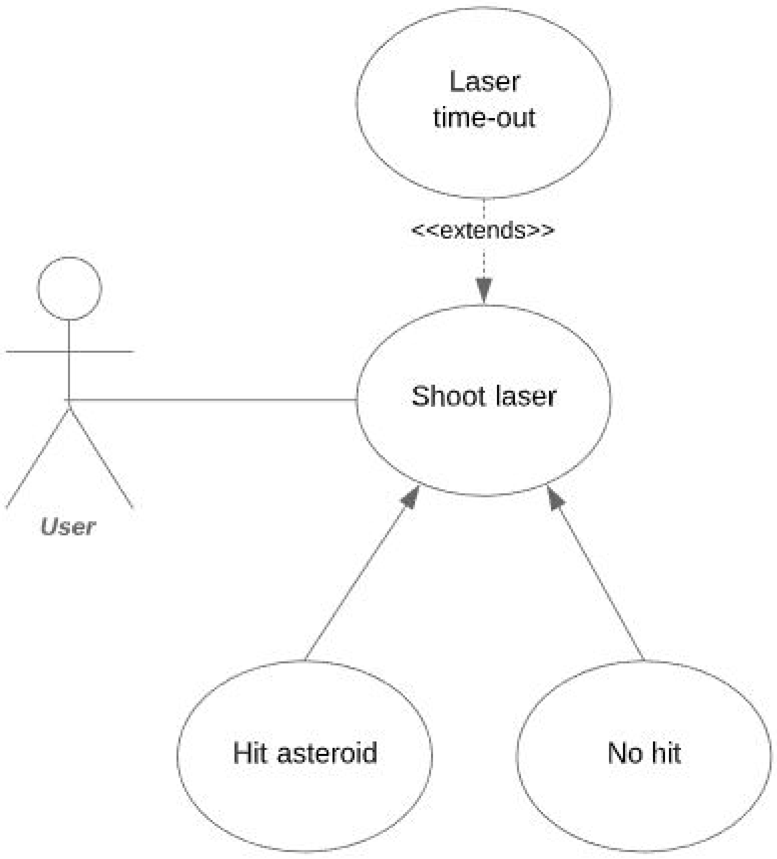
Use Case: Spaceship hits an asteroid
Author: HW
Date: 27/11/2019
Description: When the spaceship is hit by an asteroid, the system checks how many lives the spaceship has. If it only has no lives remaining, the game ends*.



* See Use Case: Game ends

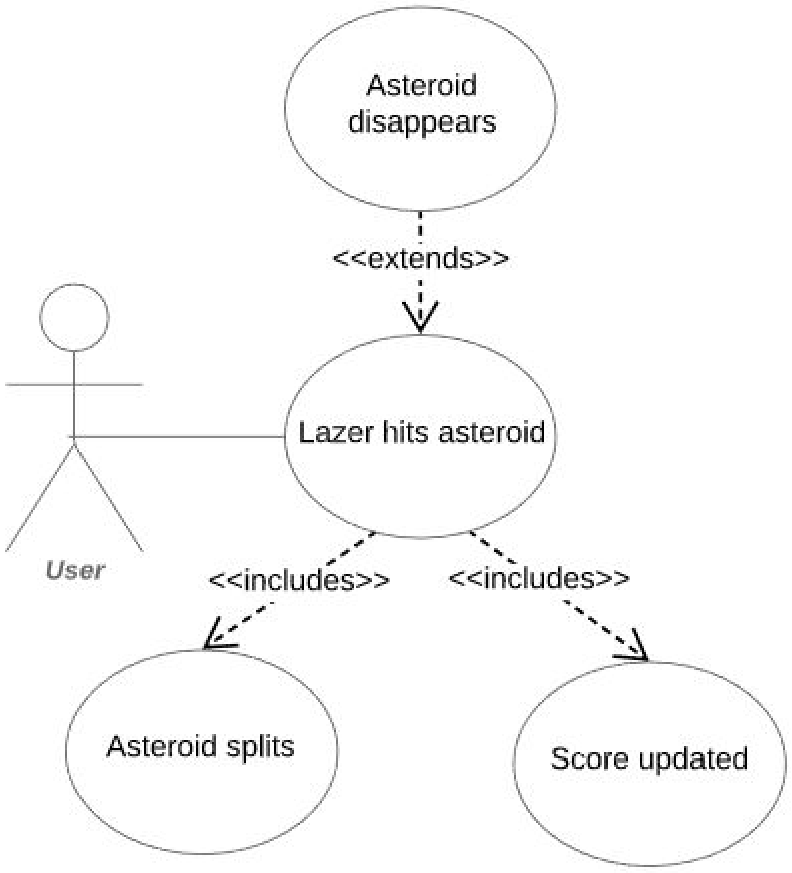
Use Case: Shoot laser
Author: HW
Date: 27/11/2019
Description: When the user shoots a laser, the system checks if 5 lasers have already been shot consecutively. If this is the case, nothing happens. Else a laser is created at the centre of the spaceship with a velocity of 7 pixels/millisecond in the direction that

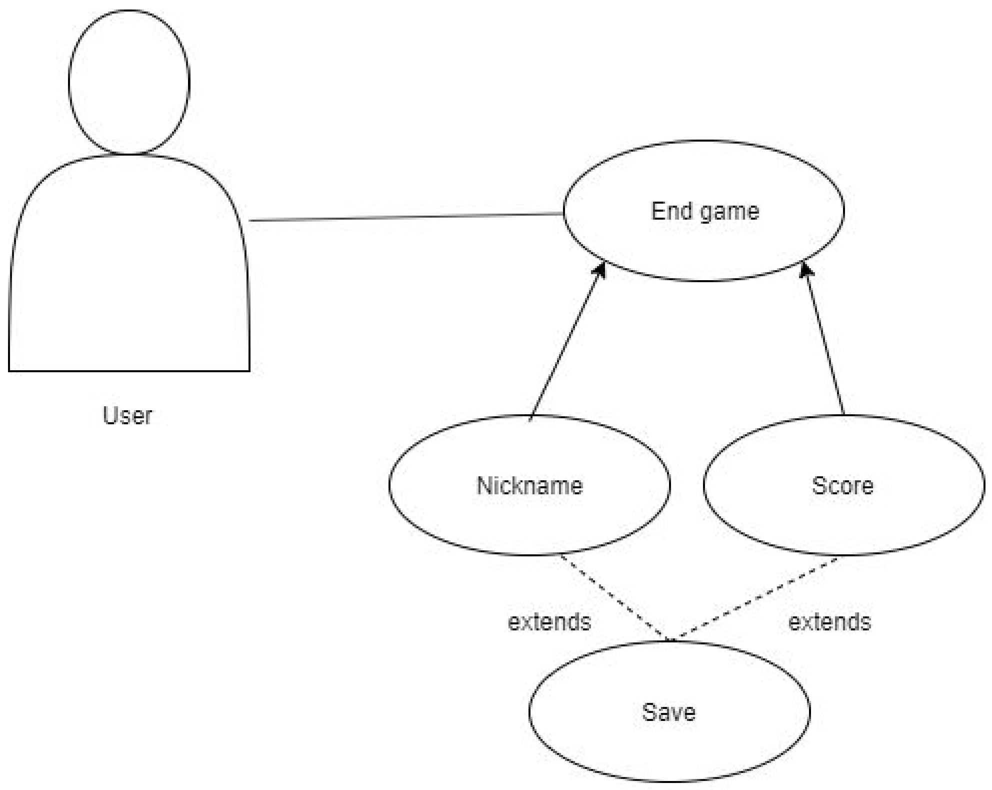
the spaceship is facing. If the laser touches an asteroid, it disappears. If it has travelled for 65 pixels, it also disappears.



Use Case: Laser hits asteroid
Author: HW
Date: 28/11/2019
Description: The user shoots a laser. When the laser hits an asteroid, the system checks if the size of the asteroid is above “small” (17 pixels). If this is not the case, the asteroid disappears. Else, the score

is updated according to the size of the asteroid (25 for small, 50 for medium or 100 for large) and the asteroid splits into 2 asteroids which are one size smaller than the original.



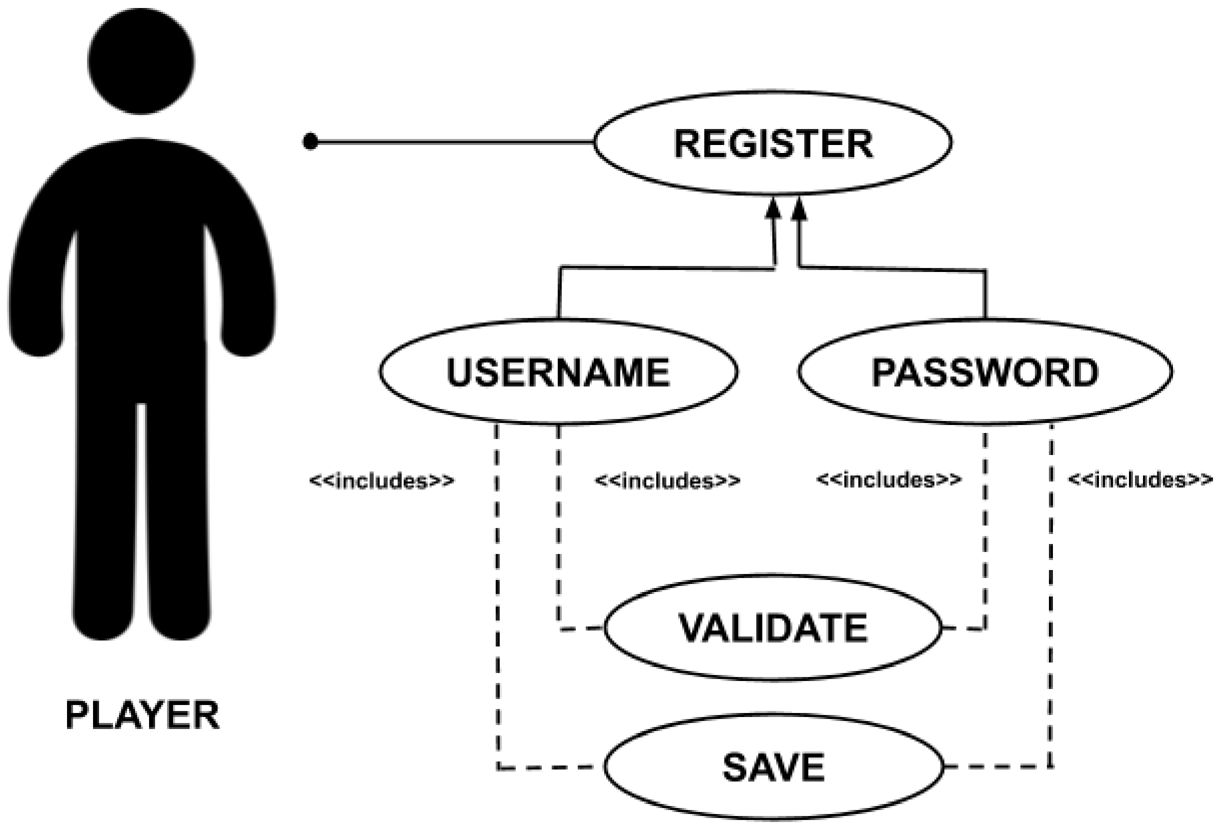


Use case: Ending a game

Date: 29-11-2019

Purpose: Ends the Asteroids game

Overview: When the game ends, The user is able to view the score and is able to fill a nickname to save the score.



USE CASE: Creating an account
DATE: 28/11/2019
DESCRIPTION: The user should be able to create an account. They click register and are shown a username and password field. They enter details into these fields and these entries should be validated (compared to criteria we provide i.e. password should between 8-16 characters and contain a number, the username cannot already be taken). Once these criteria are met, the username and password are saved and then the player is registered.