

Brooklyn College Invaders Specifications

Group 1

Table of Contents

- 1. Introduction**
 - 1.1 Overview**
 - 1.2 Background**
 - 1.3 Software Specs**
- 2. Game**
 - 2.1 UI**
 - 2.2 Game Description**
 - 2.3 Game Environment**
 - 2.4 Characters**
- 3. Use Cases**

1. Introduction

1.1 Overview

We are reimplementing the game, Space Invaders (1978), using Javascript as a new game with references to Brooklyn College faculty. The game will start by opening the application and bringing the user to the main menu.

1.2 Background

Brooklyn College Invaders mimics the original Space Invaders gameplay with a set of rows and columns of enemies. For reference, we will be using the people of Brooklyn College as a reference. The player can choose between playing as a faculty member or a student. The enemies are the trustees and administrators. The player will have three (3) lives to score points and try to reach new high scores.

2. Game

2.1 User Interface

When we first start the game, we should see a menu screen containing three options:

Play, Options, High-Score

The 'Play' button will bring you into the game once clicked on.

The 'Options' button should have a way to change the volume.

The 'High Score' button will contain the 10 highest recorded scores along with their names.

2.2 Game Description

The game has a player unit which can either be a faculty member or a student and an enemy unit which is either an administrator or a trustee. The player will have 3 lives and will be battling against waves of trustees/administrators. The enemies are arranged in five (5) rows by eleven (11) columns that move from left to right. If the trustee/admins make it to the left of rightmost corner they will drop down a row closer to the bottom.

The player will be able to fire at the enemies and will be able to move horizontally as the battle continues. Enemies will be able to shoot at the player and if they hit the player, they lose 1 life. Protecting the player will be four (4) barriers that each have 20 life and will be destroyed once their life depletes. Once you destroy an enemy you gain points: 10 points for an administrator and 20 points for a trustee. At the end of the game, your score will be recorded and if you have a top 10 score you will be prompted to enter your name in the leaderboards.

2.3 Game Environment



Figure 1.

The layout of the game will be similar to Figure 1 where the player and the amount of lives it has is shown at the bottom of the screen. The player will be able to move only left or right. Protecting the player will be the barriers above. The enemies will start at the top region of the screen and as they hit the edges they come closer to the player.

2.4 Characters

The player unit (student or faculty) is able to move by using the arrow keys and shoot with space bar.

The enemy unit (trustee or administrator) comes in waves of rows and columns and is able to shoot at the player.

There are four (4) barriers that exist to block enemy fire from the player.

3. Use Cases

A detailed description of each use case is given the remainder of this section.

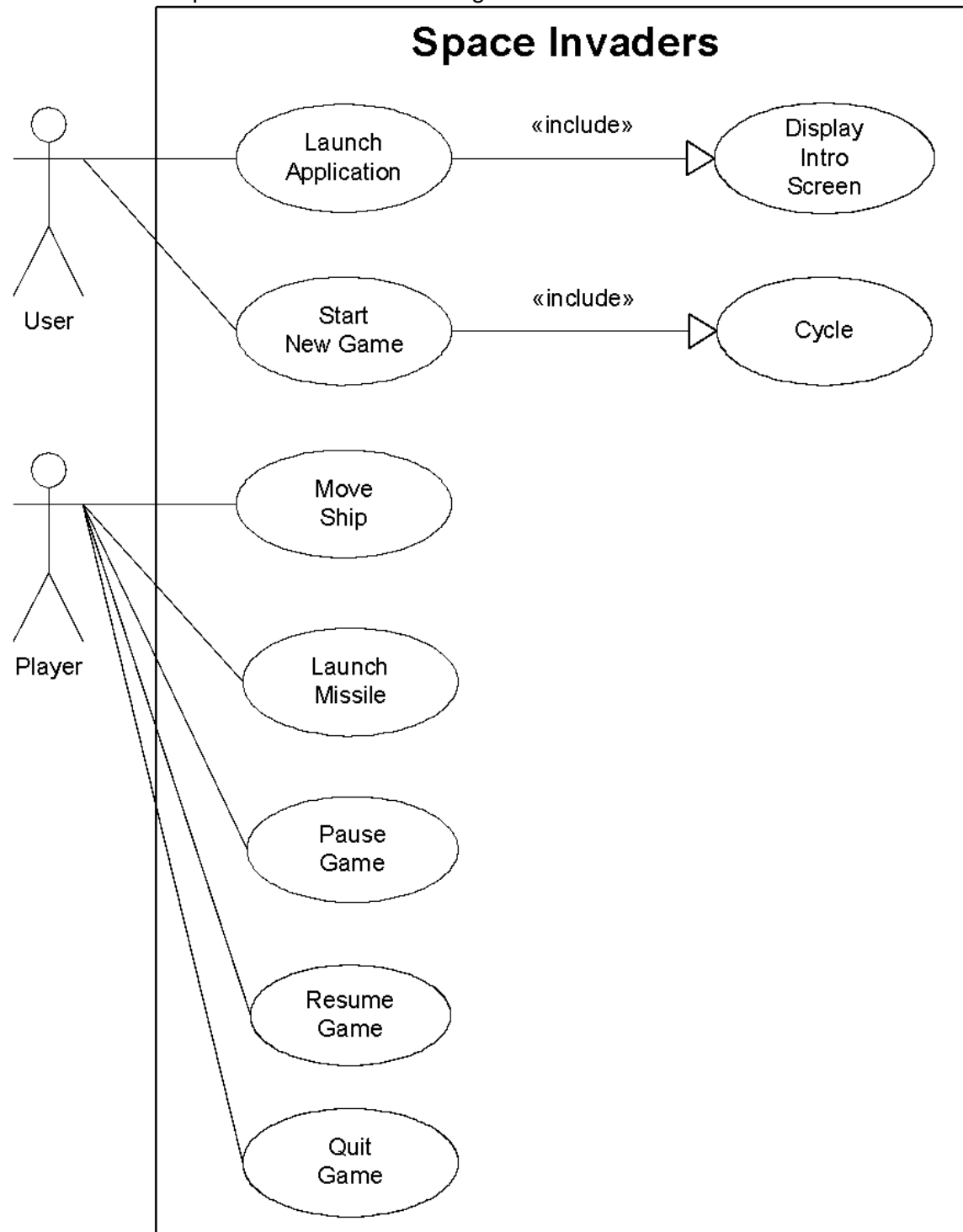


Figure 3.1
Player Use Case Diagram

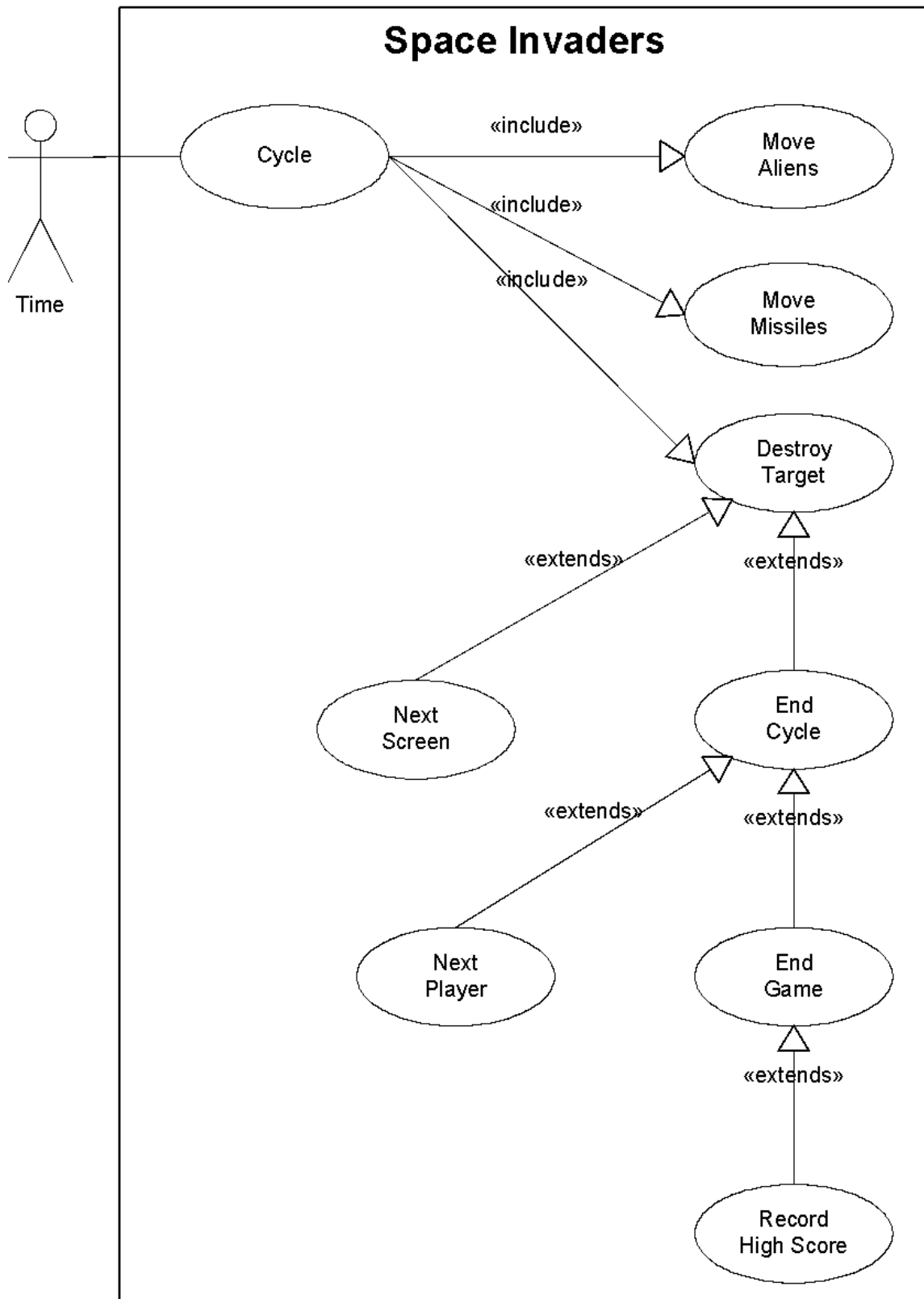


Figure 3.2
Time Use Case Diagram

2.1.1 Launch Application

Launch Application
Id: UC1
Actors: User
Preconditions: Application has been installed on the player's computer.
Flow of events: 1 Player double clicks on the Space Invaders program icon. 2 The system initializes the application by creating the various user interface screens that can be displayed to the user. 3 «include» Display Intro Screen use case. 4 Use case ends.
Post Conditions: The introduction screen is displaying, which allows the user to start a new 1 or 2 player game.

2.1.2 Display Introduction Screen

Display Intro Screen
Id: UC2
Actors: User
Preconditions: Application has been launched by the user.
Flow of events: 1 System displays the introduction screen. 2 System animates the introduction screen. 3 Use case ends.
Post Conditions: The user is allowed to start a new game.

2.1.3 Start New Game

Start New Game
Id: UC3
Actors: User
Preconditions: Introduction screen is being displayed.
Flow of events: 1 User presses either the new 1 player or 2 player game button. 2 The system initializes a new 1 or 2 player game. 3 The system displays the game screen 4 «include» Cycle use case. 5 Use case ends.
Post Conditions: The user is playing a new game

2.1.4 Move Ship

Move Ship
Id: UC4
Actors: Player

Preconditions: Game is executing in the playing/cycling state.
Flow of events: 1 User press the right or left arrow key (or moves the mouse). 2 The system moves the ship according to the gesture in step 1 and displays the ship in its new position. 3 Use case ends.
Post Conditions:

2.1.5 Launch Missile

Launch Missile
Id: UC5
Actors: Player
Preconditions: Game is executing in the playing/cycling state.
Flow of events: 1 User presses either the space bar or the left mouse button. 2 The system creates a new upward-moving missile at the ship's location. 3 Use case ends.
Post Conditions: The user is playing a new game

2.1.6 Pause Game

Pause Game
Id: UC6
Actors: Player
Preconditions: Game is executing in the playing/cycling state.
Flow of events: 1 User presses the 'P' key. 2 The system suspends the cycle use case and enters the Paused state. 3 System waits for the 'Q' or 'R' key. 4 Use case ends.
Post Conditions: The game enters (is in) the paused state.

2.1.7 Resume Game

Resume Game
Id: UC7
Actors: Player
Preconditions: Executing system is in the paused state.
Flow of events: 1 User presses the 'R' key.

2	The system resumes the game by entering the playing/cycling state and resumes the suspended cycle use case.
3	Use case ends.
Post Conditions: Game is executing in the playing/cycling state.	

2.1.8 Quit Game

Quit Game	
Id: UC8	
Actors: Player	
Preconditions: Launch Application use case has complete.	
Flow of events:	
1	User presses the 'Q' key.
2	The system ends execution of the application.
3	Use case ends.
Post Conditions: The application is no longer executing.	

2.1.9 Cycle

Cycle	
Id: UC9	
Actors: Time	
Preconditions: Execution of the New Game use case has completed.	
Flow of events:	
1	while ship is present (e.g., not destroyed) do
1.1	«include» Move Aliens Use Case.
1.2	«include» Move Missiles Use Case.
1.3	«include» Destroy Target Use Case.
1.4	if all aliens are destroyed
1.4.1	«extend» to include Next Screen use case
2	«include» End Cycle use case
3	Use case ends.
Post Conditions:	

2.1.10 End Cycle

End Cycle	
Id: UC10	
Actors: Time	
Preconditions: A cycle use case is completing.	
Flow of events:	
1	If there is a next player, which includes another ship for a single player
1.1	«include» Next Player Use Case.
2	else if there is no next player and no next ship
2.1	«include» Game Use Case.
3	Use case ends.

Post Conditions:

2.1.11 Next Screen

Next Screen
Id: UC11
Actors: Time
Preconditions: A game cycle is executing.
Flow of events: 1 System will pause the cycle by not allowing any game input 2 System will clear the screen 3 System will create a new screen with lower positioned aliens 4 System will resume the cycle by allowing game input 5 Use case ends.
Post Conditions: User will be playing a new game cycle

2.1.12 Next Player

Next Player
Id: UC12
Actors: Time
Preconditions: A cycle has ended. There is a player with existing ships.
Flow of events: 1 System will display the screen associated with the next player 2 Use case ends.
Post Conditions: System will begin a new cycle

2.1.13 End Game

End Game
Id: UC13
Actors: Time
Preconditions:
Flow of events: 1 If the score for the current game is > minimal score in the history list 1.1 «include» Record High Score use case... 2 Use case ends.
Post Conditions:

2.1.14 Record High Score

Record High Score
Id: UC14

Actors: Time
Preconditions: End Game was executed
Flow of events: 1 System will add the current score and player to the high score history list. 2 Use case ends.
Post Conditions: Player and user is now on the high score list.