Team: David Baird
Aaron Holt
Kyle Olsen

Title: Asteroids 3D

Project Summary:
A game inspired by the 2D game asteroids brought into 3 dimensions.

Project Requirements:

We have no business requirements.

User Requirements:		
ID	**Requirement**	**Priority**
U-1	Movement	
U-1.1	I can change the pitch angle	Critical
U-1.2	I can change the yaw angle	Critical
U-1.3	I can thrust the ship forward	Critical
U-1.4	I can do all movement actions at the same time	Critical
U-2	Weapons	
U-2.1	I can fire a weapon	Critical
U-2.2	I can switch to an alternate weapon (when available)	Low
U-2.3	Weapon actions may be done at the same time as movement actions.	Critical
U-3	Settings	
U-3.1	I can turn turn all sound on and off	Medium
U-3.2	I can turn the music on and off	Low
U-3.3	I can turn the sound effects on and off	Low

U-4	In-game actions	
U-4.1	I can pause the game	High
U-4.2	I can save the game	Low
U-4.3	I can exit the game	High
U-5	Profiles	
U-5.1	I can select a profile from existing profiles	High
U-5.2	I can create a profile	High
U-5.3	I can delete a profile	Medium
U-6	Statistics	
U-6.1	I can view global statistics	High
U-6.2	I can view profile statistics	Medium

Functional Requirements:		
ID	**Requirement**	**Priority**
F-1	Player Profiles	
F-1.1	Player profiles are stored	High
F-2	Player Data	
F-2.1	High scores are stored	High
F-2.2	Player's highest level is stored	Medium
F-2.3	Asteroids destroyed is stored	Medium
F-2.4	Player deaths is stored	Medium

F-3	Object Interactions	
F-3.1	When the ship encounters one of the boundary walls it is transported to the come out of the wall immediately opposite it	Medium
F-3.2	Asteroids react to appropriate weapon hits	Critical
F-3.3	Asteroid warp upon encountering boundary walls	Medium
F-3.4	A projectile (a specific weapon type) will disappear upon collision with a wall	Low
F-3.5	There is a percentage chance a weapon/power-up will be spawned after an asteroid is destroyed	Low
F-3.6	Upon collision between ship and an asteroid, the ship takes damage	High
F-3.7	Upon collision with the ship, an asteroid reacts as if hit by a weapon	Low
F-3.8	After taking damage the ship is temporarily invulnerable	Low
F-3.9	When two asteroids collide they bounce off each other	High
F-3.10	When a larger asteroid is hit by a weapon or the ship it spawns smaller asteroids with random velocities from around the original asteroids position	Low
F-3.11	User is warned about objects in proximity to them	Low
F-3.12	Upon collision between a ship and a projectile, the ship takes damage	Low
F-3.13	Upon collision with anything, a projectile is destroyed	High
F-3.14	Upon collision with another ship, both ships take damage and bounce off each other	Low
F-4	Sound	
F-4.1	Sound effects will be associated with selected in game actions and interactions	Low
F-4.2	There will be in game music	Low

F-5	Movement	
F-5.1	There is a maximum linear velocity	Low
F-5.2	There is a maximum angular velocity	Low
F-6	Gameplay	
F-6.1	When all asteroids are completely destroyed the level is completed	Low
F-6.2	Upon enough damage to the ship a life is lost	High
F-6.3	After losing ‘Available Lives’, the level is failed	High
F-7	Viewing Angle	
F-7.1	The user’s view is from behind the ship	Critical

Non-Functional Requirements:		
ID	**Requirement**	**Priority**
N-1	Graphics	
N-1.1	FPS >= 30 on computer with Intel® Haswell integrated graphics	High

Data Storage:

Type: MongoDB

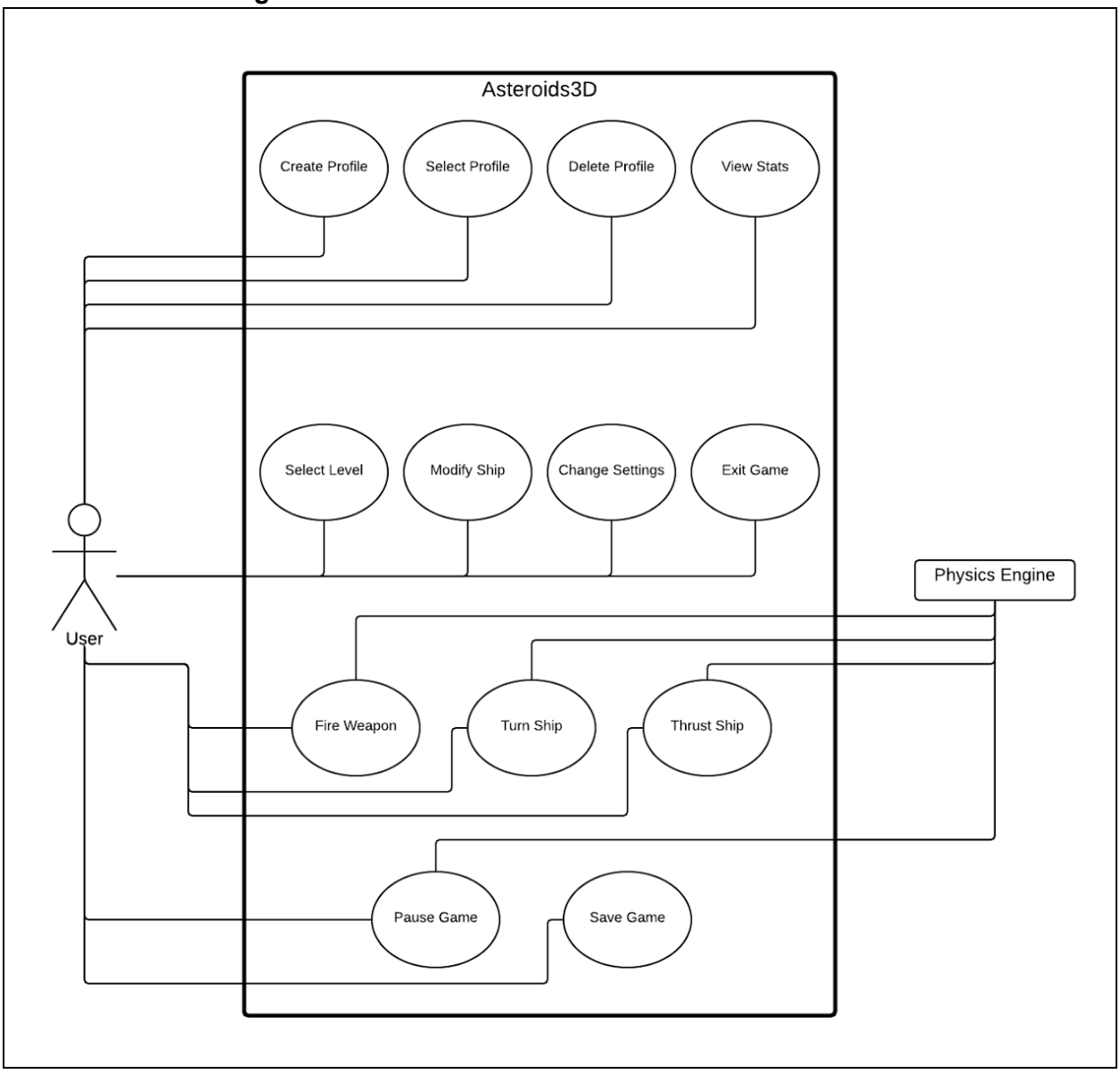
Usage: Player Profiles, Saved Levels (both user and system)

Classes:

- DatabaseInterface (An interface)
- MongoDBInterface(Concrete Class)

Use Cases:

Use Case Diagram:



Use Case Documents:

Use Case Id:	UC-1
Use Case Name:	Thrust Ship
Description:	The user can cause a force on the ship in the direction the ship is facing. The ship will keep a linear velocity between 0 and a maximum.

Actors:	Player		
Pre-Conditions:	In game play (Not in a menu)		
Post-Conditions:	The linear velocity of the ship is changed		
Frequency of Use:	Very often, whenever the player wants to move or change their movement		
Flow of Events:		Actor Action	System Response
	1	Press thrust control	Apply force and adjust for maximum linear velocity
	2	Hold thrust control	Continue applying force and adjust for maximum linear velocity
	3	Release thrust control	
Variations:			
Issues and Notes:			
Developer Notes:			

Use Case Id:	UC-2
Use Case Name:	Create profile
Description:	User can create a profile.

Actors:	Player		
Pre-Conditions:	At profile select screen		
Post-Conditions:	Profile is stored. Profile is selectable by user.		
Frequency of Use:			
Flow of Events:		Actor Action	System Response
	1	Selects ‘Change Profile’	Show profile creation form
	2	Enters profile information	
	3	Submits form	Profile created and stored
Variations:			
Issues and Notes:			
Developer Notes:			

Use Case Id:	UC-3
Use Case Name:	View Stats
Description:	User can view statistics

Actors:	Player		
Pre-Conditions:	At welcome screen		
Post-Conditions:	At welcome screen		
Frequency of Use:	Infrequently		
Flow of Events:		Actor Action	System Response
	1	Click ‘Stats’	Fetch and display statistics. This includes global and profile statistics
	2	Click different tab	Display appropriate statistics
	3	Click ‘Return to Menu’	Return to the menu
Variations:			
Issues and Notes:			
Developer Notes:			

Use Case Id:	UC-4
Use Case Name:	Turn Ship
Description:	User can adjust the ship's orientation.

Actors:	Player		
Pre-Conditions:	User is in game (not paused or in a menu).		
Post-Conditions:	The ship is rotated		
Frequency of Use:			
Flow of Events:		Actor Action	System Response
	1	Press positive yaw control	Ship rotates one tick with respect to current orientation
	2	Hold positive yaw control	Ship turns one tick per event cycle with respect to current orientation
	3	Release positive yaw control	
Variations:	1. Press negative yaw control 1. Press positive pitch control 1. Press negative pitch control 2. Hold positive pitch control 2. Hold negative pitch control 2. Hold negative yaw control		
Issues and Notes:			
Developer Notes:			

Use Case Id:	UC-5
Use Case Name:	Select profile
Description:	User can select a profile

Actors:	Player		
Pre-Conditions:	At profile select screen.		
Post-Conditions:	Profile information is loaded.		
Frequency of Use:			
Flow of Events:		Actor Action	System Response
	1	Selects ‘Change Profile’	Show profile select screen with available profiles
	2	Clicks available profile	
	3	Select profile	Any currently loaded profile is unloaded. Submitted profile loaded
Variations:			
Issues and Notes:			
Developer Notes:			

Use Case Id:	UC-6
Use Case Name:	Delete profile
Description:	User can delete a profile.

Actors:	Player		
Pre-Conditions:	At profile select screen		
Post-Conditions:	Profile is deleted. Statistics pertaining to deleted profile are deleted.		
Frequency of Use:			
Flow of Events:		Actor Action	System Response
	1	Selects ‘Change Profile’	List available profiles for deletion
	2	Click desired profile for deletion	
	3	Select ‘Delete’	Profile and profile’s statistics are deleted
Variations:			
Issues and Notes:			
Developer Notes:			

Use Case Id:	UC-7
Use Case Name:	Pause game
Description:	User can pause the game

Actors:	Player		
Pre-Conditions:	In game, not in a menu.		
Post-Conditions:	Game pauses, game state is stored, the pause screen is displayed.		
Frequency of Use:			
Flow of Events:		Actor Action	System Response
	1	Presses ‘pause’ control	Game pauses, game state is stored, the pause screen is displayed.
	2		
	3		
Variations:			
Issues and Notes:			
Developer Notes:			

Use Case Id:	UC-8
Use Case Name:	Save game
Description:	Game state for current level is stored

Actors:	Player		
Pre-Conditions:	In game at the pause menu.		
Post-Conditions:	Game state is stored.		
Frequency of Use:			
Flow of Events:		Actor Action	System Response
	1	Click ‘Main Menu’	Game state for current level is stored, overwriting a previous saved game for that profile if it exists
	2	Click ‘Yes’	Updates and returns to Welcome Screen
	3		
Variations:	1. Click ‘Exit Game’ 2. Click ‘No’		
Issues and Notes:			
Developer Notes:			

Use Case Id:	UC-9
Use Case Name:	Exit game
Description:	Game state for selected level is loaded

Actors:	Player		
Pre-Conditions:	At welcome screen, pause screen, or profile screen		
Post-Conditions:	Application closes		
Frequency of Use:			
Flow of Events:		Actor Action	System Response
	1	Press ‘Exit Game’ control	Display a message asking if the user really wants to exit
	2	Press ‘Yes’ control	Application closes
	3		
Variations:			
Issues and Notes:			
Developer Notes:			

Use Case Id:	UC-10
Use Case Name:	Select level
Description:	The level is selected

Actors:	Player		
Pre-Conditions:	At profile screen		
Post-Conditions:	Playing selected level		
Frequency of Use:			
Flow of Events:		Actor Action	System Response
	1	Press ‘level select’ control	Level select screen with available levels to play is displayed
	2	Select level	Screen asking to play from start or load game is displayed
	3	Press ‘load level’ control	The specified game state is loaded and the game starts
Variations:	3. Press ‘start’ control (starts level from beginning)		
Issues and Notes:			
Developer Notes:			

Use Case Id:	UC-11
Use Case Name:	Change settings
Description:	The selected settings are modified

Actors:	Player		
Pre-Conditions:	At the welcome screen		
Post-Conditions:	The selected settings are modified		
Frequency of Use:			
Flow of Events:		Actor Action	System Response
	1	Click ‘Settings’	The settings screen is displayed
	2	Change setting	
	3	Click ‘Return to Menu’	Show dialog with a ‘Save’ and ‘Cancel’ option.
	4	Click ‘Yes’	Updates and returns to Welcome Screen
Variations:	4. Click ‘No’		
Issues and Notes:			
Developer Notes:			

Use Case Id:	UC-12
Use Case Name:	Fire weapon
Description:	The weapon is fired

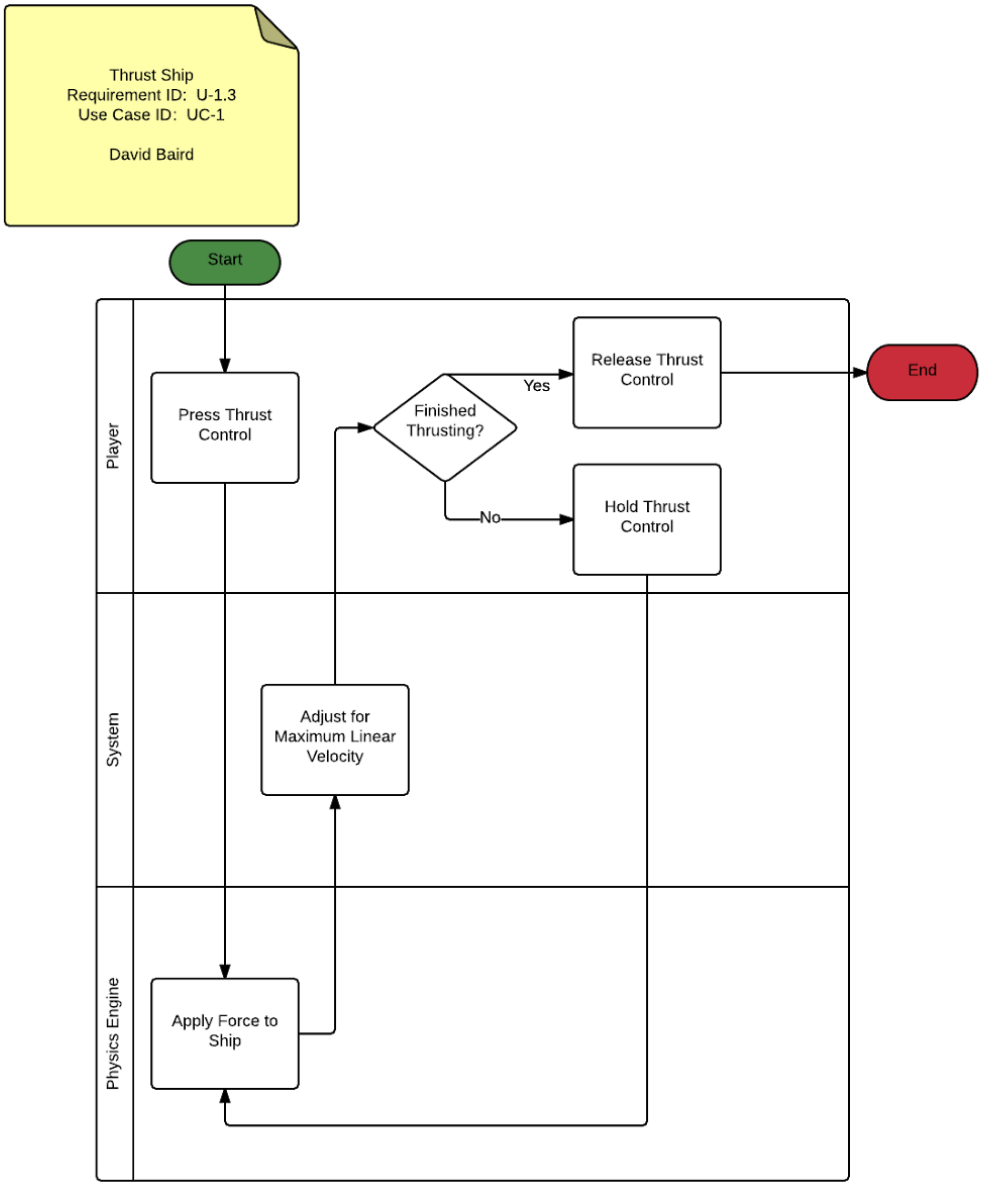
Actors:	Player		
Pre-Conditions:	In game, not in a menu.		
Post-Conditions:	The current weapon fires. A projectile is launched from the ship in the direction it is facing.		
Frequency of Use:			
Flow of Events:		Actor Action	System Response
	1	Press weapon control	A projectile is launched from the ship in the direction the ship is facing, accounting for the maximum projectile limit.
Variations:			
Issues and Notes:			
Developer Notes:			

Use Case Id:	UC-13
Use Case Name:	Modify Ship
Description:	Change ship properties based on item pickups and available weapons.

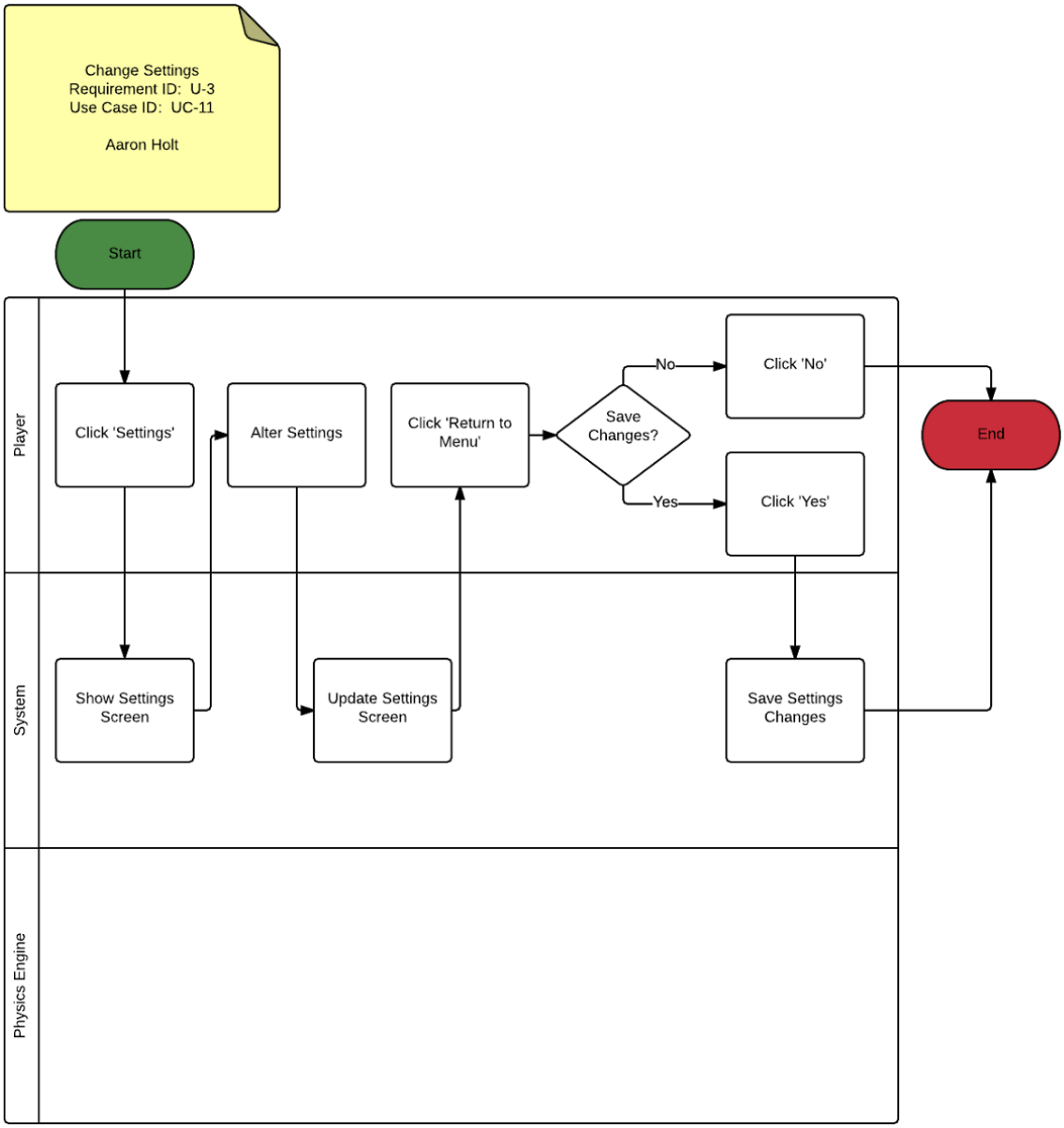
Actors:	Player		
Pre-Conditions:	At level select screen		
Post-Conditions:	The ship properties are modified		
Frequency of Use:			
Flow of Events:		Actor Action	System Response
	1	Click ‘Change Loadout’	The modify ship screen is displayed.
	2	Change setting	
	3	Click ‘Return to Menu’	
	4	Save settings	The changes ship settings are saved.
Variations:			
Issues and Notes:			
Developer Notes:			

Activity Diagram:

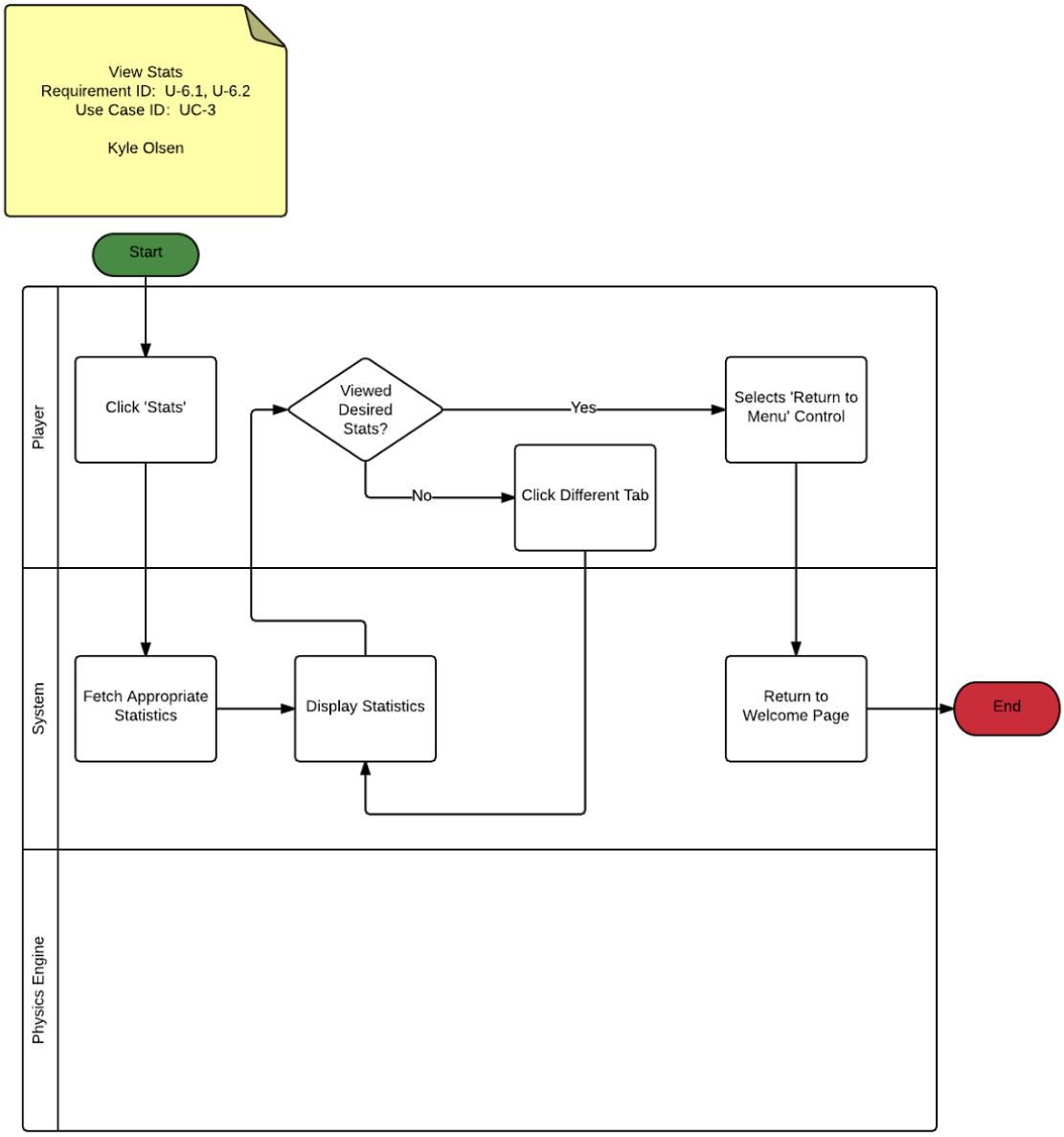
David Baird:



Aaron Holt:

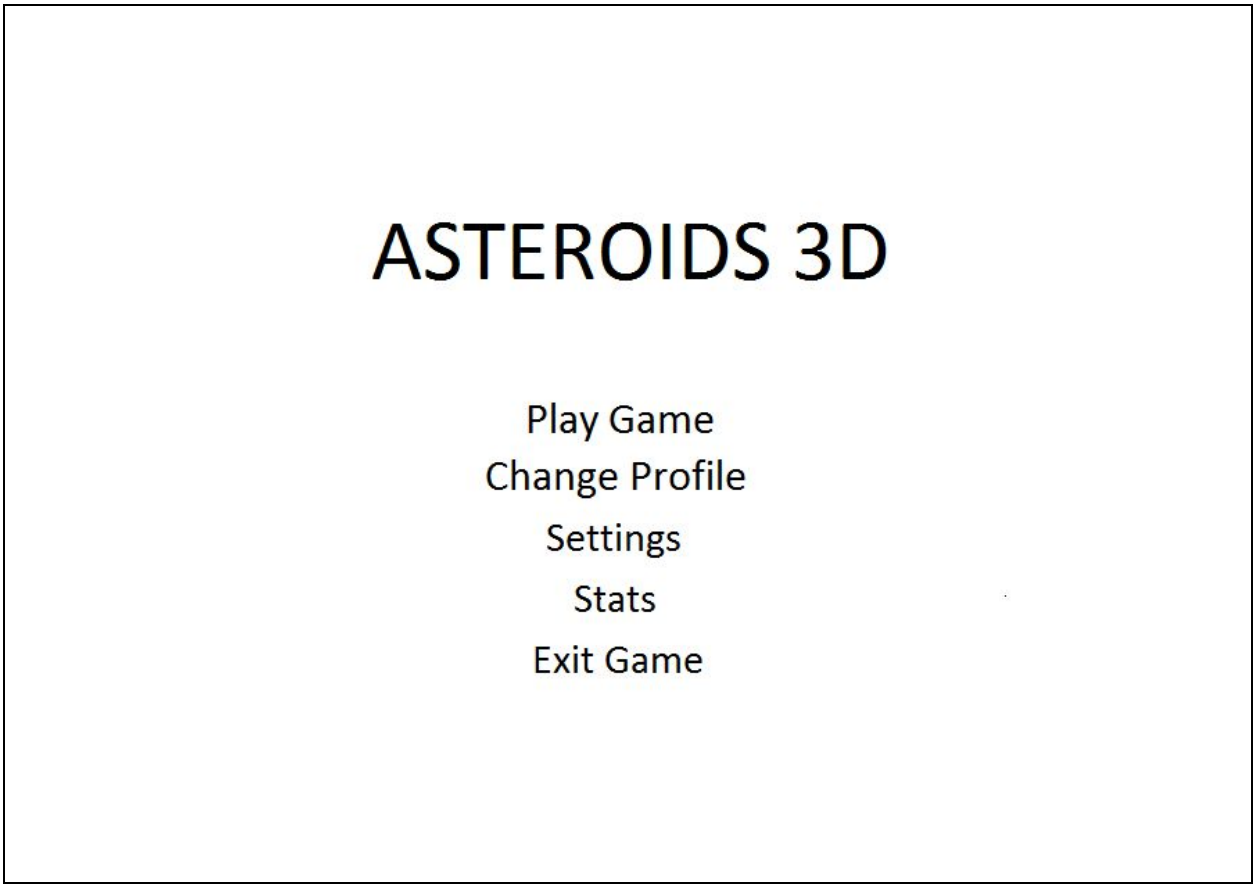

Change Settings
Requirement ID: U-3
Use Case ID: UC-11
Aaron Holt
Start
Player
System
Physics Engine
Click 'Settings'
Alter Settings
Click 'Return to Menu'
Save Changes?
No
Yes
Click 'No'
Click 'Yes'
End
Show Settings Screen
Update Settings Screen
Save Settings Changes

Kyle Olsen:

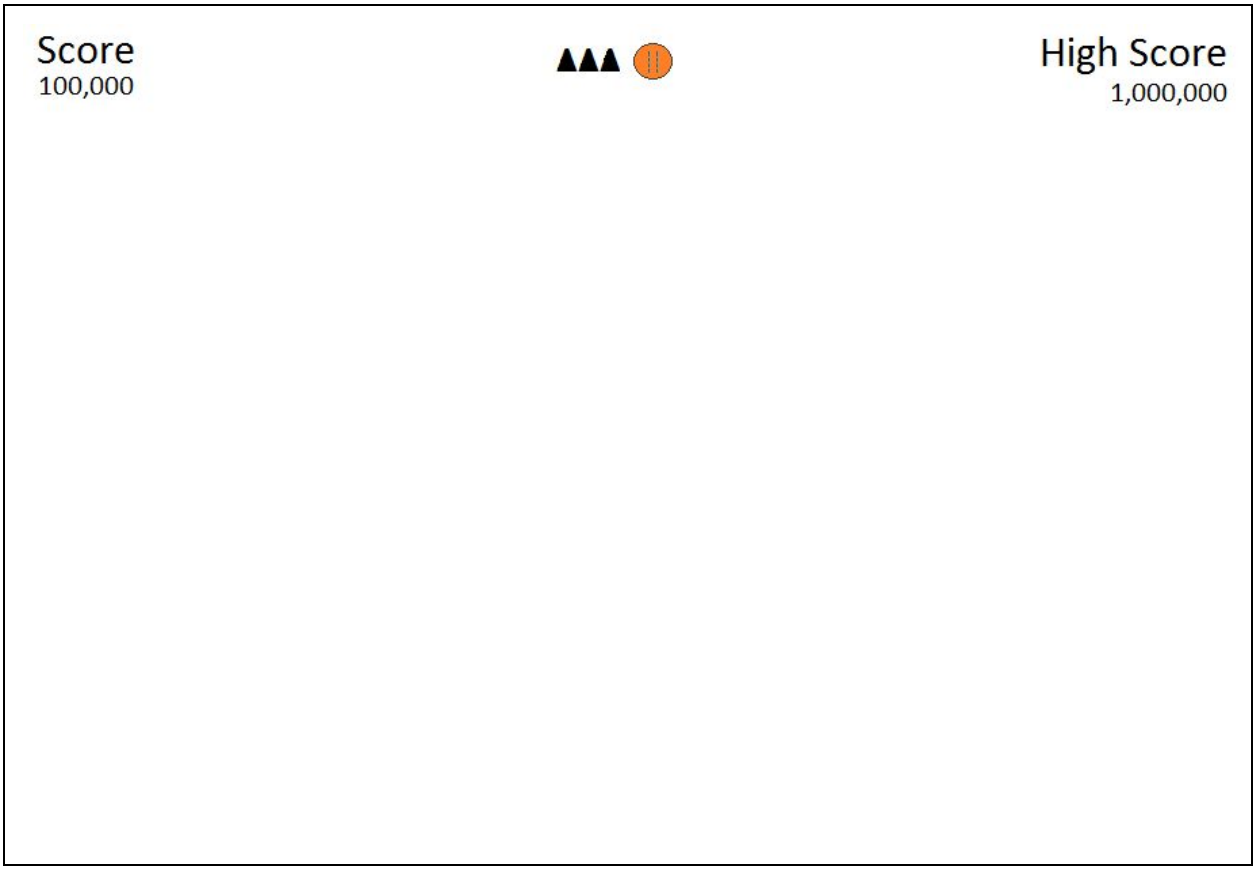

View Stats
Requirement ID: U-6.1, U-6.2
Use Case ID: UC-3
Kyle Olsen
Start
Player
Click 'Stats'
Viewed Desired Stats?
Yes
Selects 'Return to Menu' Control
No
Click Different Tab
System
Fetch Appropriate Statistics
Display Statistics
Return to Welcome Page
End
Physics Engine

UI Mockups:

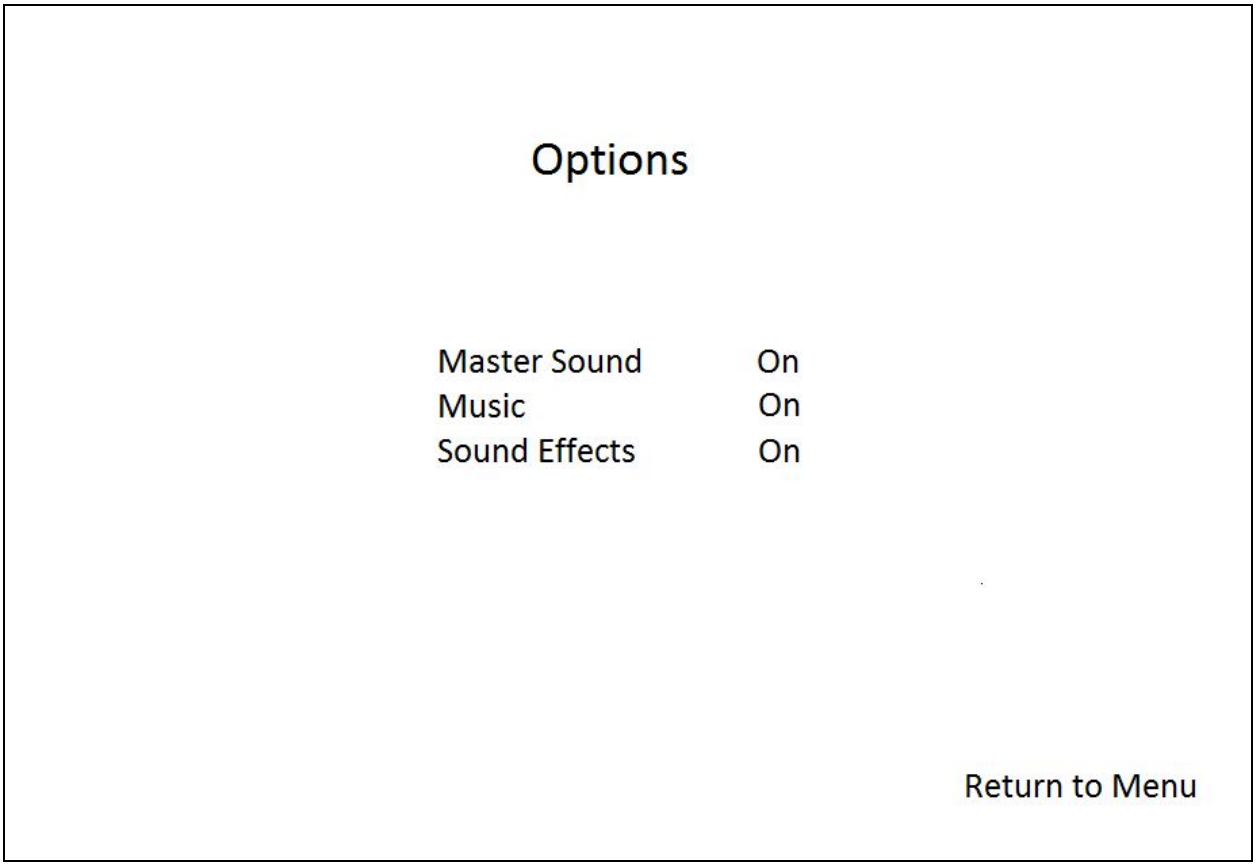
Welcome Screen: Screen that you see when you start the application. The main screen.



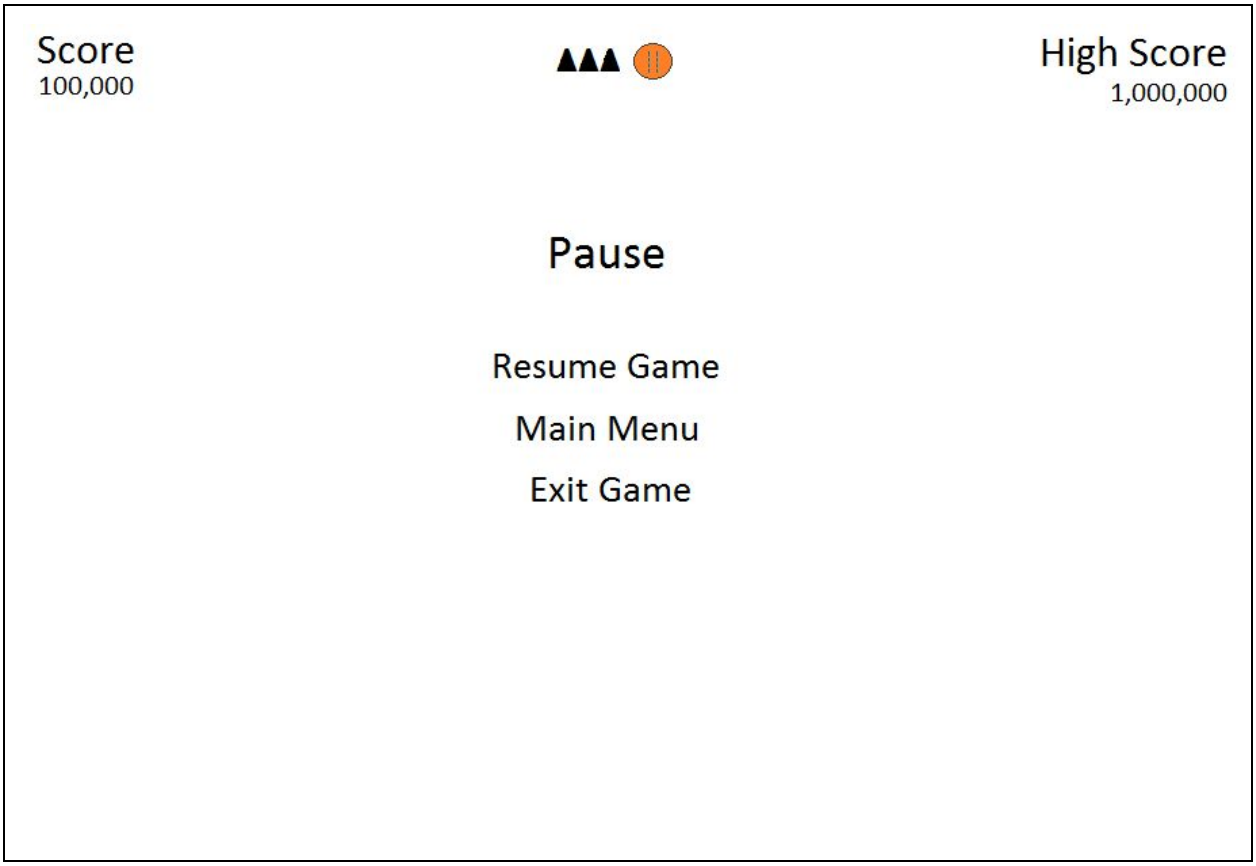
In Game Screen: Screen you see when you're playing the game.



Options Screen: Screen to change options at.

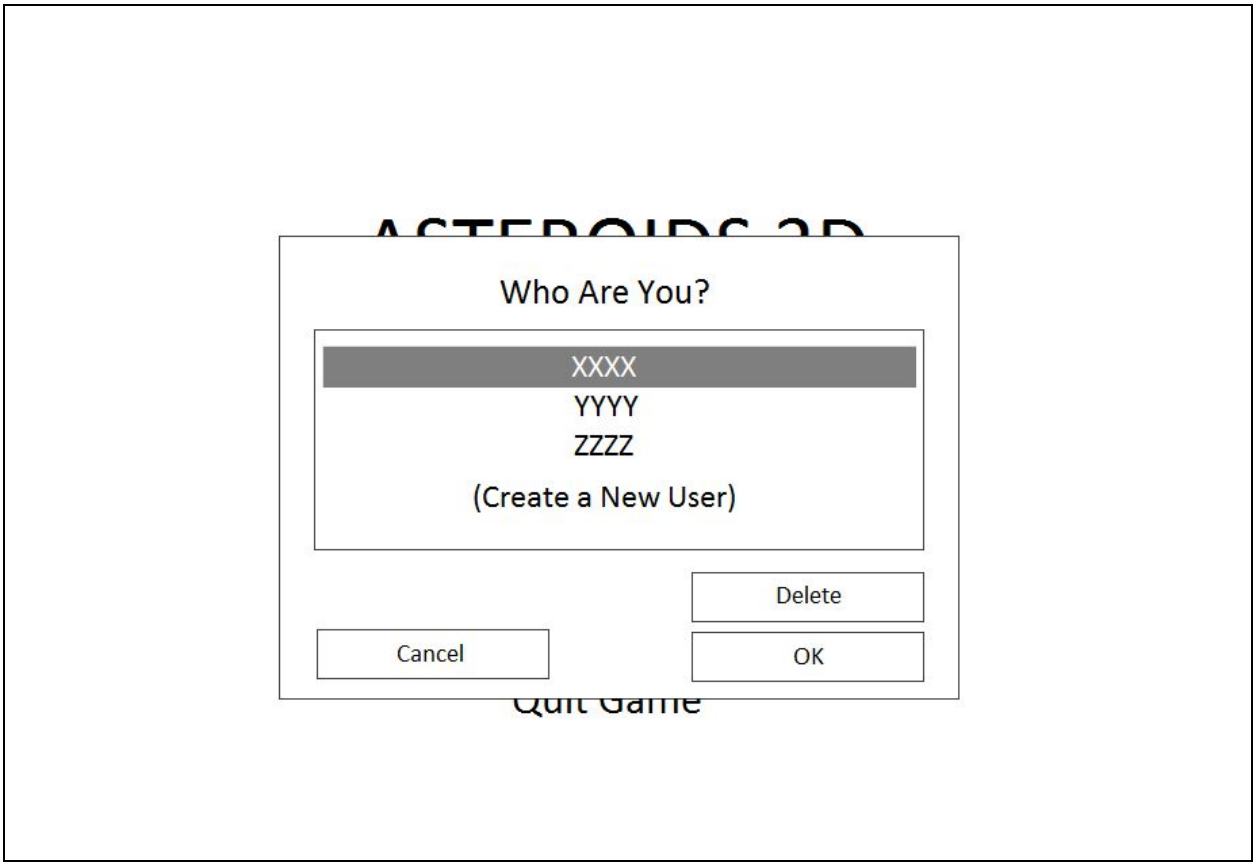


Pause Screen: Screen shown when gameplay is paused.

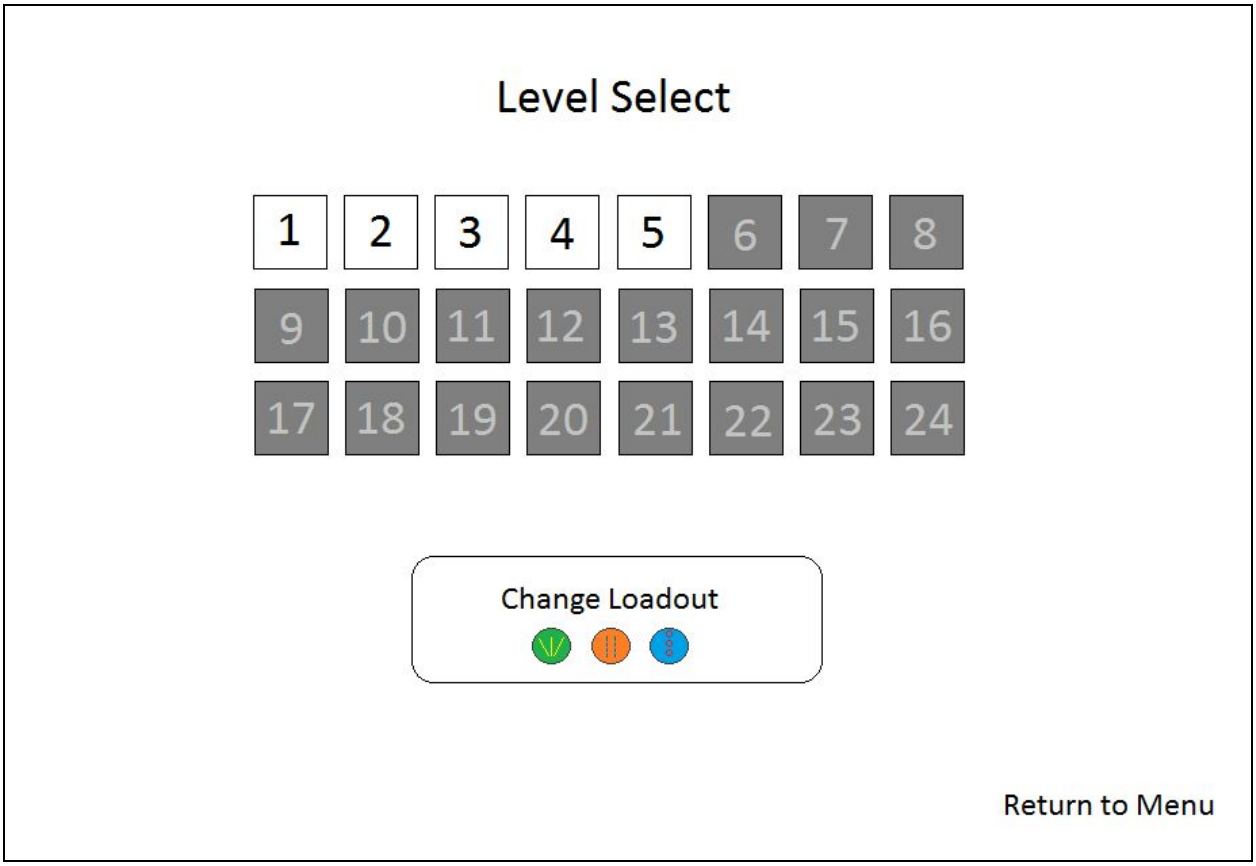


Stats Screen: Screen that shows statistics.

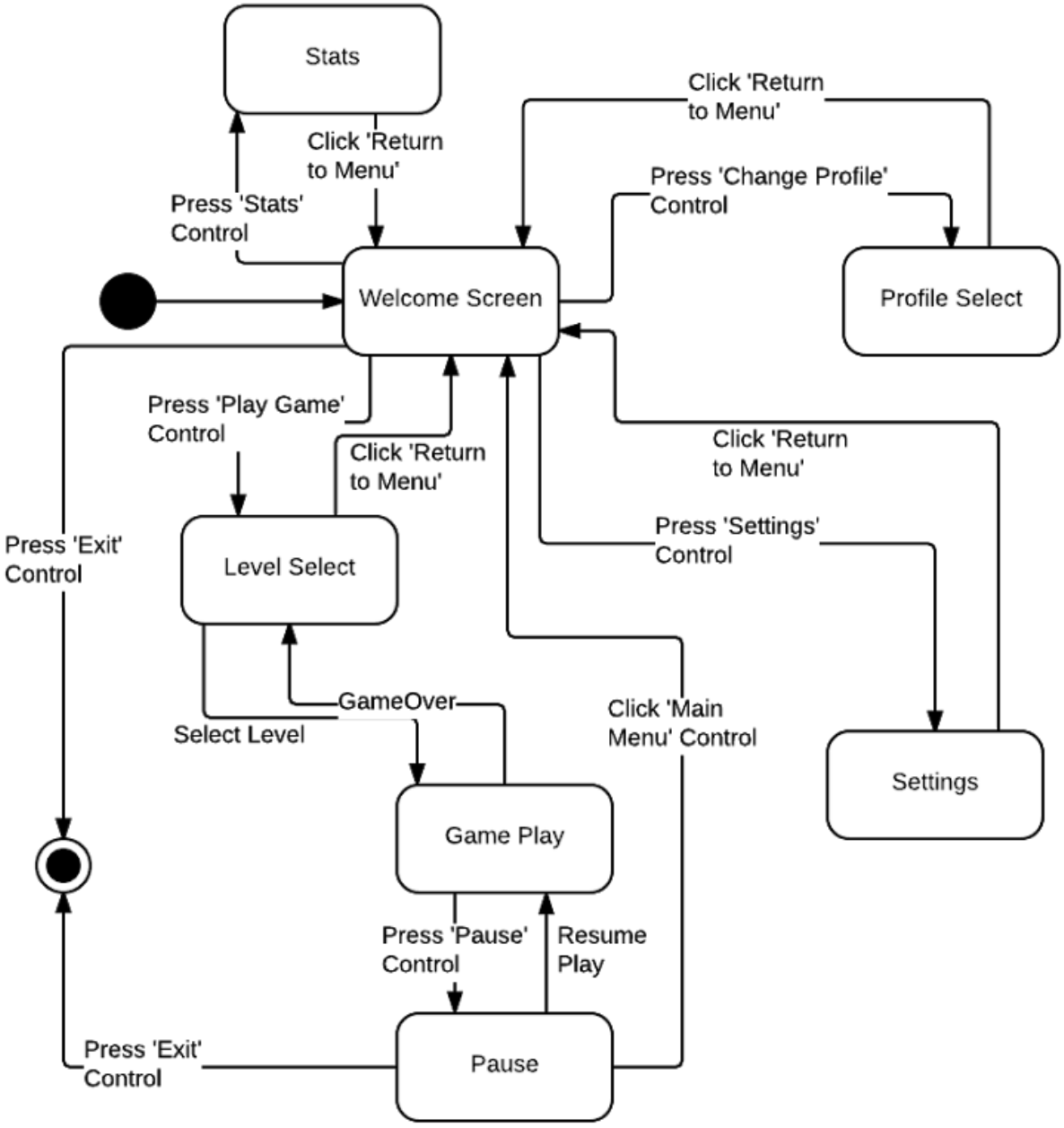
Profile Screen: Screen where you can create, delete, and select a profile.



Level Select Screen: Screen where the current secondary weapons can be selected by the player. Additional the player can select the next level to play.

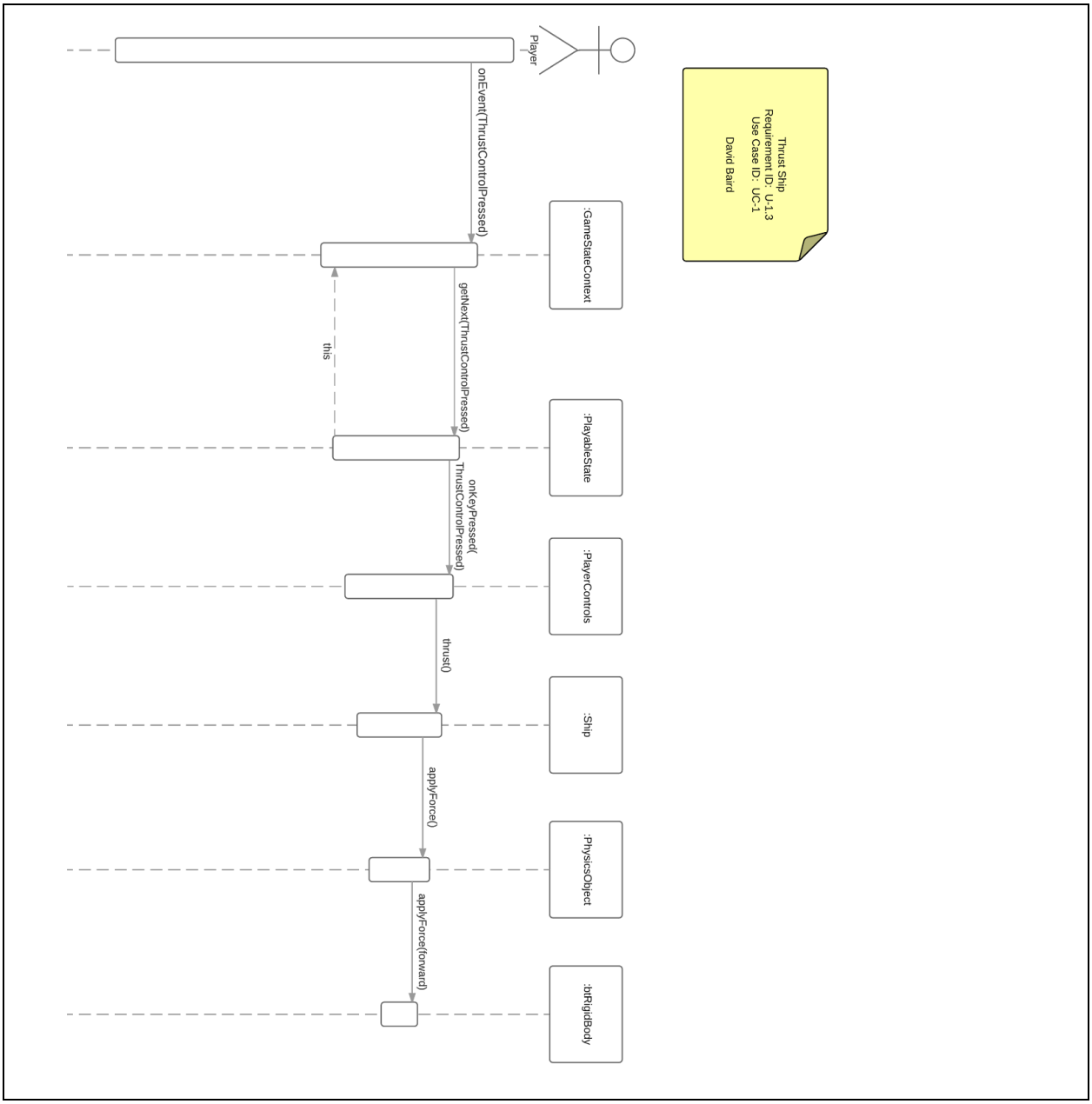


State Diagram:

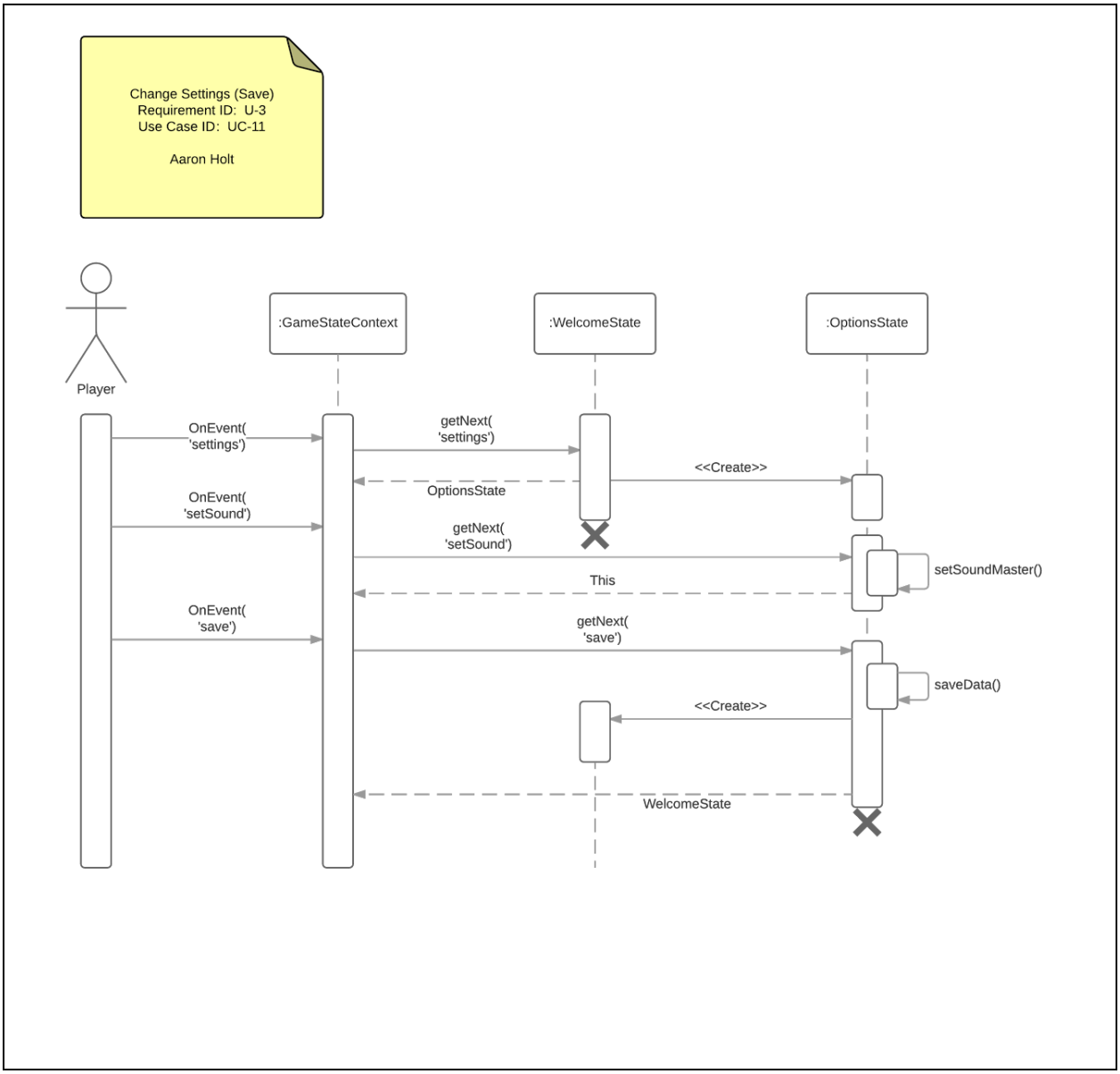


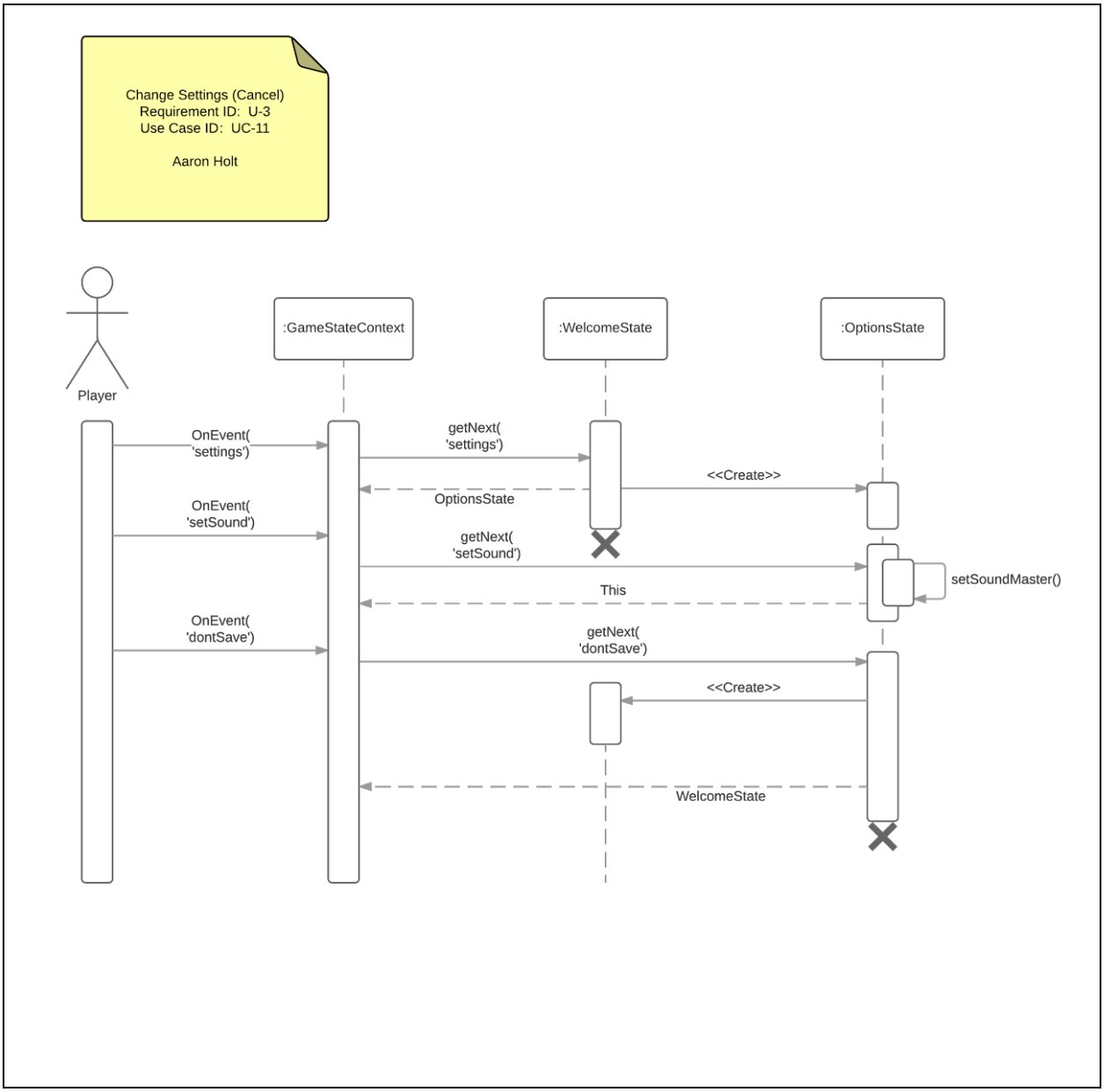
User Interactions:

David Baird:

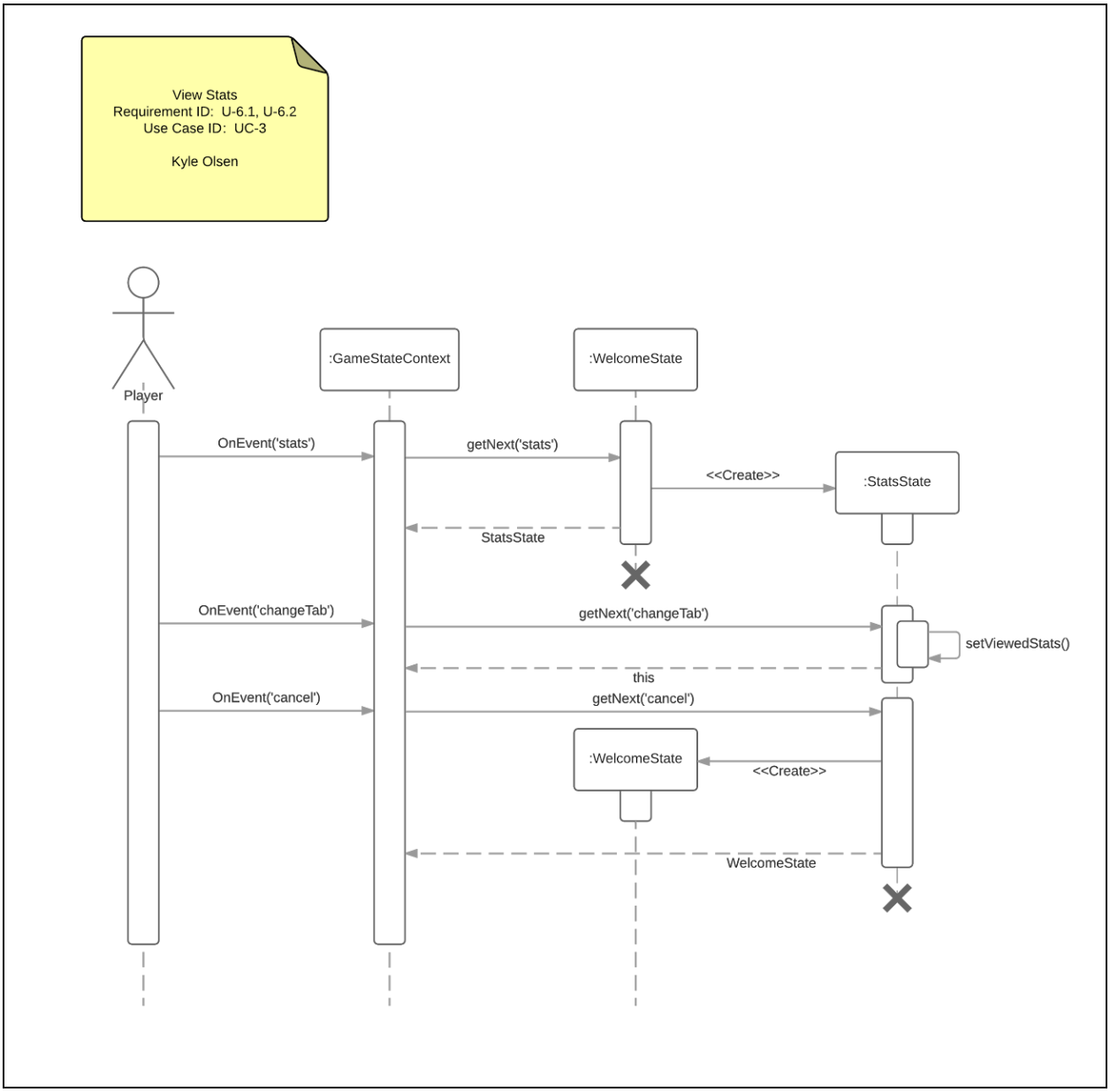


Aaron Holt:

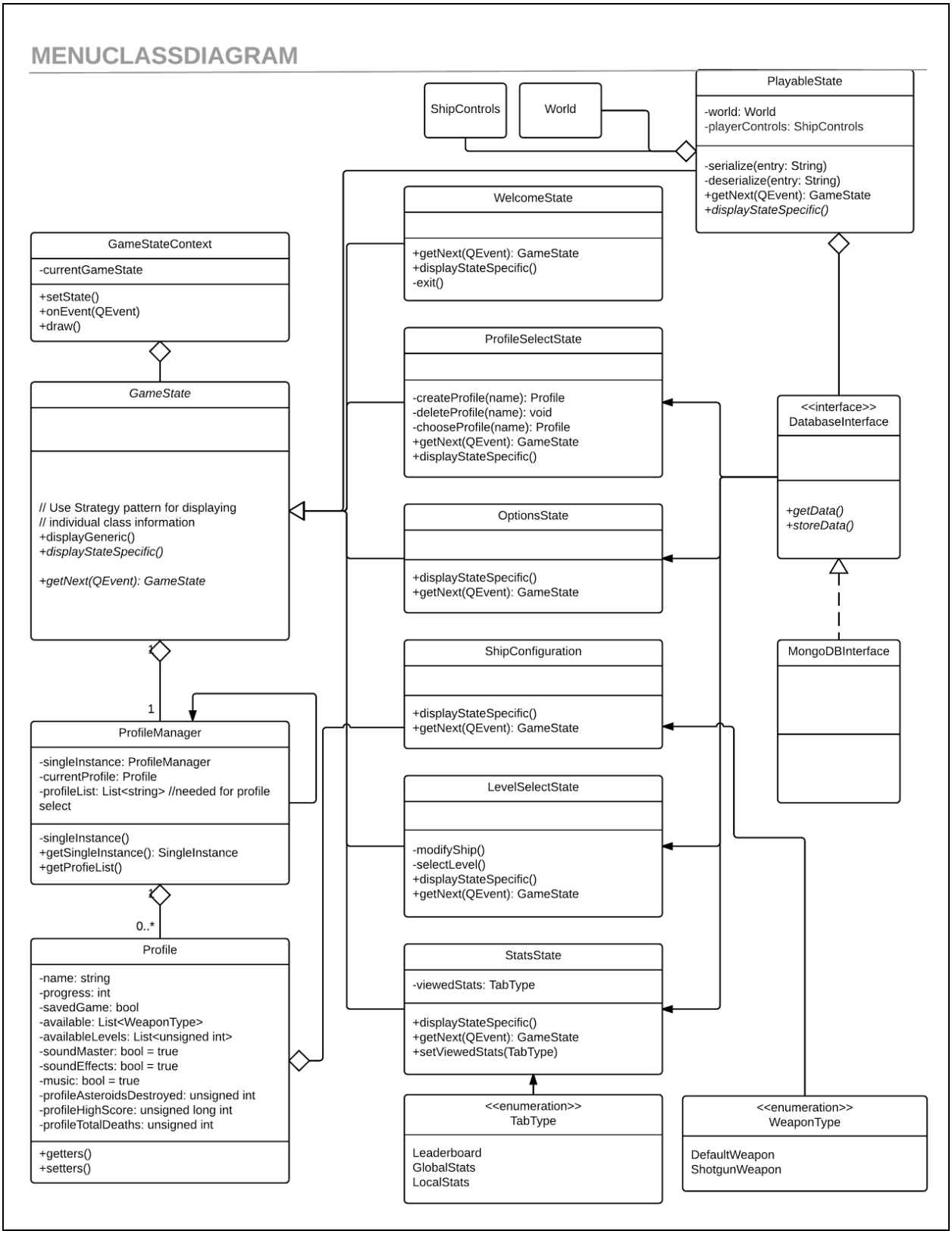
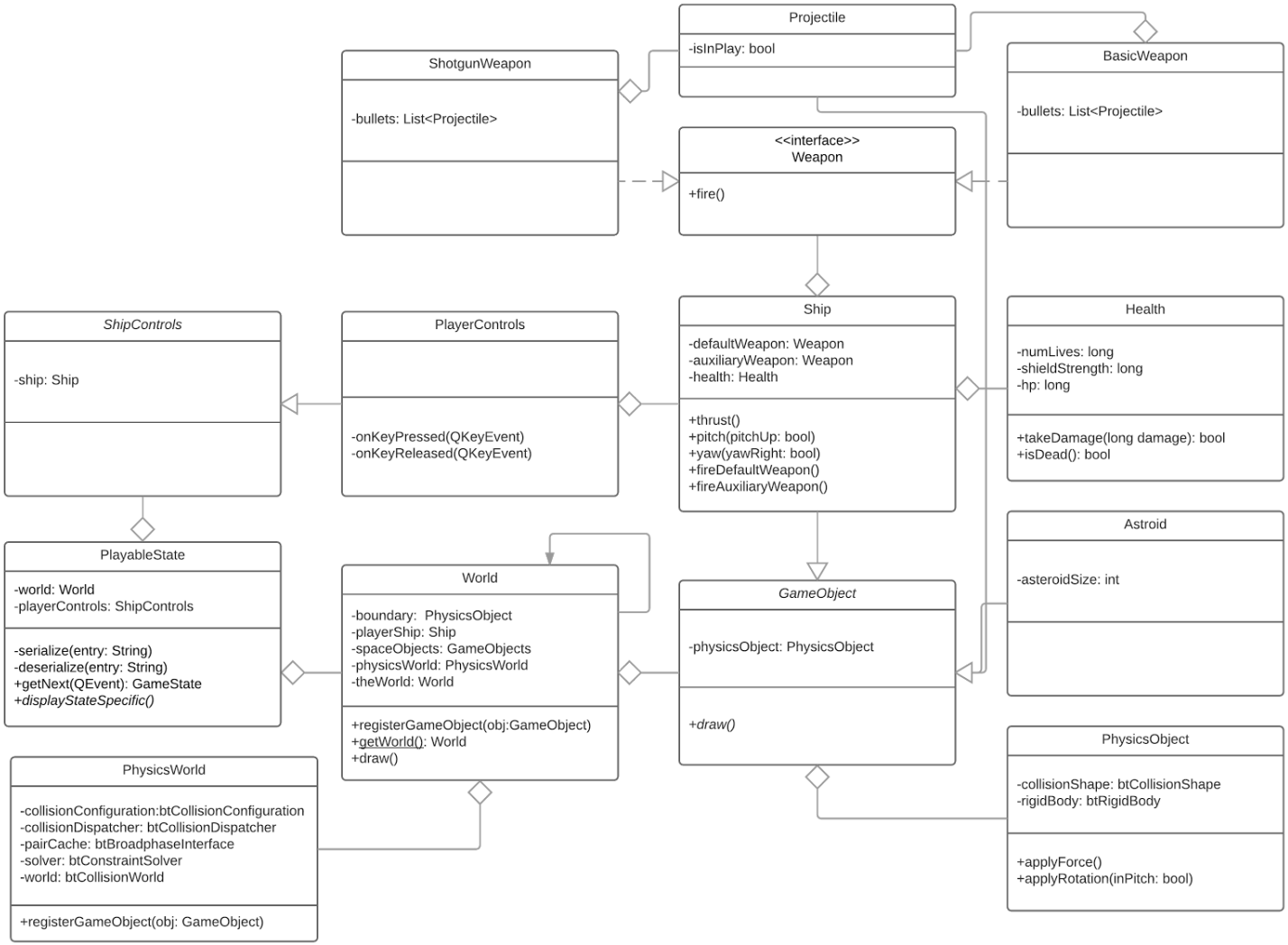

Change Settings (Save)
Requirement ID: U-3
Use Case ID: UC-11
Aaron Holt
Player
:GameStateContext
:WelcomeState
:OptionsState
OnEvent('settings')
getNext('settings')
<<Create>>
OptionsState
OnEvent('setSound')
getNext('setSound')
setSoundMaster()
This
OnEvent('save')
getNext('save')
saveData()
<<Create>>
WelcomeState


Change Settings (Cancel)
Requirement ID: U-3
Use Case ID: UC-11
Aaron Holt
Player
:GameStateContext
:WelcomeState
:OptionsState
OnEvent('settings')
getNext('settings')
<<Create>>
OptionsState
OnEvent('setSound')
getNext('setSound')
setSoundMaster()
This
OnEvent('dontSave')
getNext('dontSave')
<<Create>>
WelcomeState

Kyle Olsen:


View Stats
Requirement ID: U-6.1, U-6.2
Use Case ID: UC-3
Kyle Olsen
Player
:GameStateContext
:WelcomeState
:StatsState
OnEvent('stats')
getNext('stats')
<<Create>>
StatsState
OnEvent('changeTab')
getNext('changeTab')
setViewedStats()
this
OnEvent('cancel')
getNext('cancel')
:WelcomeState
<<Create>>
WelcomeState

Class Diagrams:



PLAYABLESTATECLASSDIAGRAM