USE CASE SPECIFICATIONS

COM4050-N Rapid Application Development

2016/17
RICHARD LUCAS N3088581

Contents

User

Input

Use case: Input	
Overview: Takes an input from the User's controller and returns which AbsPlayer use case to activate.	
Actors (primary and secondary): User, System	
Pre-conditions: The game has started and the user has progressed into the gameplay	
Post conditions:	
Basic Flow of actions:	

	Event	Response
1	Button Pressed	Depending on Button Pressed activate AbsPlayer use cases.

Alternative Flows:	
If left Analogue stick moved	Activate the ship's Move use case and pass it a vector which is the direction the analogue stick was pressed in.
If Right Analogue stick moved	Activate the ship's Shoot Bullet use case and pass it a vector which is the direction the analogue stick was pressed in.
If the A button is pressed	Activate the ship's Shoot Missile use case, if it is not on cool-down.
If the X button is pressed	Activate the ship's Boost use case.
If Left Trigger or Right Trigger is pressed	Activate the ship's Roll Dodge use case.
If either Left Bumper or Right Bumper is pressed	Activate the ship's Flip Dodge use case.
If the Back button is pressed	Activate the AbsPlayer's Upgrade use case.

Extensions		
1	If the Start button is pressed	Bring up the pause menu and stop all gameplay functions.
2	If the game is paused and the Start button is pressed	Resume gameplay functions.

AbsPlayer

Move

Use case: Move	
Overview: Move the AbsPlayer's ship in a given direction.	
Actors (primary and secondary): System, AbsPlayer	
Pre-conditions: The ship is instantiated and the use case has been given a vector from the User's Input use case.	
Post conditions:	
Basic Flow of actions:	
Event	**Response**
1 When given a Vector input	Move the ship in the direction of that vector.
Extensions	
1 If the player touches a pickup	Apply that pickup's properties to the ship.

Use case: Boost	
Overview: Moves the player forwards at an increased speed with invulnerability and any ship that is touched takes a large amount of damage.	
Actors (primary and secondary): System, AbsPlayer	
Pre-conditions: The ship is instantiated, the player's Boost remaining is greater than 0 and the use case has been given 'X Button pressed' from the User's Input use case.	
Post conditions:	
Basic Flow of actions:	

	Event	**Response**
1	When X button is received from the Input	Instantiate a particle effect around the player's ship, lock the Players rotation and move the player in the direction they are facing.
2	Upon moving the player	Reduce the players boost remaining by a predefined amount per second.
Extensions		
1	If the players ship collides with an enemy ship while they are boosting	Activate the hit ship's take damage use case.

Upgrade

Use case: Upgrade	
Overview: Opens the upgrade menu, which pauses the game, and allows the User to select an upgrade.	
Actors (primary and secondary): System, HuPlayer	
Pre-conditions: The HuPlayer is instatiated, the HuPlayer has reached a predefined threshold of Scrap and the use case has been given 'back Button pressed' from the User's Input use case.	
Post conditions:	
Basic Flow of actions:	

	Event	Response
1	When Back button is received from Input	Pause all gameplay functions, open the Upgrade tree screen and allow a selection of one upgrade from the upgrade tree.
2	Upon the user selecting an accessible upgrade	The upgrade is permanently applied to the HuPlayer's Ship and the Ship's attributes and level are upped.
3	Upon the upgrade being applied	The Upgrade Tree screen is closed and gameplay functions are resumed.

Flip Dodge

Use case: Flip Dodge	
Overview: Makes the AbsPlayer's ship invulnerable and rotates it about its width in a backflip motion.	
Actors (primary and secondary): System, AbsPlayer	
Pre-conditions: The ship is instantiated, the player's Flip Dodge function is not on cool-down, and the use case has been given either 'Left Bumper pressed' or 'Right Bumper Pressed' from the User's Input use case.	
Post conditions:	
Basic Flow of actions:	

	Event	Response
1	When either Left Bumper pressed or Right Bumper Pressed is received from Input	Rotate the ship 360 degrees about the ships width and the ship is made invulnerable. Additionally set the Function to being on cool-down.
2	Upon completion of the rotation	The ship is made vulnerable.

Roll Dodge

Use case: Roll Dodge	
Overview: Makes the AbsPlayer's ship invulnerable and moves it to either the left or the right, relative to the ship's rotation.	
Actors (primary and secondary): System, AbsPlayer	
Pre-conditions: The ship is instantiated, the player's Roll Dodge function is not on cool-down, and the use case has been given either 'Left Trigger pressed' or 'Right Trigger Pressed' from the User's Input use case.	
Post conditions:	
Basic Flow of actions:	

	Event	Response
1	When Left Trigger is received from Input	Rotate the ship 360 degrees anti clockwise about its length while the ship is moved to its port(left) side by a predefined amount and the ship is made invulnerable. Additionally set the Function to being on cool-down.
2	Upon completion of the rotation and movement to either side	The ship is made vulnerable.

Alternative Flows:	
When Right Trigger is received from Input	Rotate the ship 360 degrees clockwise about its length while the ship is moved to its Starboard(right) side by a predefined amount and the ship is made invulnerable. Additionally set the Function to being on cool-down.

Use case: Shoot Bullet	
Overview: Rotates the AbsPlayer's ship and fires a bullet in that direction.	
Actors (primary and secondary): System, AbsPlayer	
Pre-conditions: The ship is instantiated and the use case has been given a vector from the User's Input use case.	
Post conditions:	
Basic Flow of actions:	

	Event	Response
1	When given a vector input	Rotate the ship to face that direction vector.
2	Ship is rotated	Instantiate the number of bullets defined by the player and give them the direction vector for them to move along.
Extensions		
1	If the bullet collides with an enemy ship	Activate that ship's Take Damage use case.

Use case: Shoot Missile	
Overview: Instantiates a predefined number of missiles which choose a target and then moves towards it until it hits the target or are dodged.	
Actors (primary and secondary): System, Missile	
Pre-conditions: The player's Shoot Missile function is not on cool-down, the ship is instantiated and the use case has been given 'A Button pressed' from the User's Input use case.	
Post conditions:	
Basic Flow of actions:	

	Event	Response
1	When A button is received from Input	Instantiate the number of Missiles defined by the Player and activate those missiles' Choose Target use case.
2	A number of Missiles was instantiated	Set the players Shoot Missile function to being on cool down.

Use case: Take Damage

Overview: If this AbsPlayer collides with another AbsPlayer or is hit by a Bullet or Missile, this AbsPlayer's ship takes an amount of damage based on the object its hit by.

Actors (primary and secondary): AbsPlayer, Bullet, Missile

Pre-conditions: A HuPlayer has been instantiated, at least one AiPlayer has been instantiated and two objects have collided.

Post conditions:

Basic Flow of actions:

	Event	Response
1	When this ship collides with another object	Apply damage based on what type of object this ship was hit by

Alternative Flows:

If hit by a Bullet	Reduce this ships health by the Bullet's damage stat.
If hit by a Missile	Reduce this ships health by the Missile's damage stat.
If hit by another Ship	Reduce this ships health by a small amount.
If hit by another Ship that is currently boosting	Reduce this ships health by a very large amount.

Extensions

1	If this ship is NOT owned by the HuPlayer	Add a an amount of scrap to the HuPlayers Scrap that is relative to the Aiplayer's ship's level.
2	If this ship IS owned by the HuPlayer	The game over screen is displayed and the User is taken back into the menu system.

AiPlayer

Change Playstyle

Use case: AI Change Playstyle	
Overview: allows the AI ships to change its set of available actions either based on time or when the ship is spawned	
Actors (primary and secondary): AiPlayer	
Pre-conditions: The HuPlayer and an AiPlayer has been instantiated	
Post conditions:	
Basic Flow of actions:	

	Event	Response
1	When instantiated	Choose a random playstyle from the available list of playstyles
Extensions		
1	If the current playstyle is aggressive and the Ship falls below half health	Choose a defensive playstyle.
2	If the current playstyle is defensive and the HuPlayer's ship is at a great distance	Choose an aggressive playstyle.

Missile

Choose Target

Use case: Missile Choose Target	
Overview: Choses a target in from of the Missile and move towards it.	
Actors (primary and secondary): System, Missile	
Pre-conditions: The Missile has been instantiated.	
Post conditions:	
Basic Flow of actions:	

	Event	Response
1	On Activation	Find what targets are in a predefined range.
2	Upon knowing what targets are in range	Remove those that are 'behind' the Missile from the list.
3	Upon removal of those that are 'behind' the missile	Check the distance to remaining targets and select the one which is closest.
4	Selection of the closest possible target	Set as target.
5	Target selected	Send the Target to the Missile's Move use case.

Alternative Flows:	
If there are no possible targets	The missile will move in a straight line in the direction the player is facing.

Use case: Missile Move

Overview: Move the missile towards the given target.

Actors (primary and secondary): System, Missile

Pre-conditions: The Missile is instantiated and a target is chosen

Post conditions:

Basic Flow of actions:

	Event	Response
1	System tick	Find the Targets location.
2	Target location found	Rotate the Missile to face the target.
3	Missile rotated	Move in the direction of the Target.

Alternative Flows:

If the Target Dodges when the Missile is within a certain range of the Target	The missile loses the target and moves in a straight line.

Extensions

1	If the Missile collides with an enemy ship	Activate that ship's Take Damage use case.