Name: David Kleckner
Title: AstroExplorer: Voyage Home
Requirements/Documentation

Use Case ID:	UR-01		
Use Case Name	Player Sprite Movement		
Description	As the player, I want to be able to move through space with my jetpack.		
Actors:	Player		
Pre-condition	The user must already be in the game and able to move.		
Post-condition	The User moves the direction that they want to move.		
Frequency of Use	Almost constantly in-game		
Flow of Events:		Actor Action	System Response
	1	Player presses movement key	Fuel depletes and sprite moves specified direction.
	2	Player releases movement key	Stops directional momentum until player slows.
	3.	Player chooses another action	
Variations	In 1, if player collides with an in game object, health goes down.		

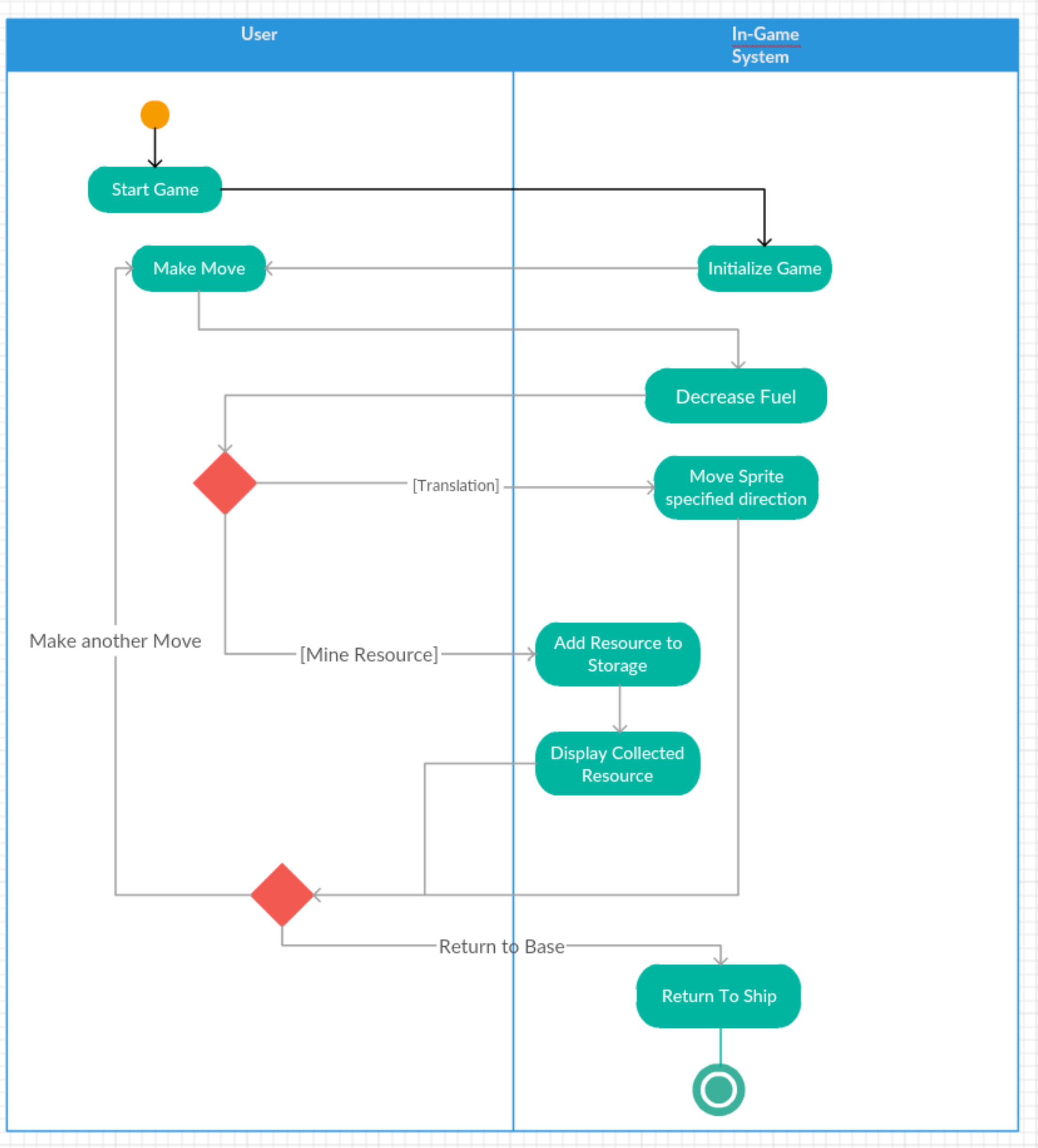
Use Case ID:	UR-02		
Use Case Name	Player Sprite Mining		
Description	As the player, I want to be able to mine resources as I move through asteroids.		
Actors:	Player		
Pre-condition	The user must already be in the game and on a mineable object		
Post-condition	The User gains material from mineable object.		
Frequency of Use	A necessary function in-game		
Flow of Events:		Actor Action	System Response
	1	Player presses mining key	Fuel depletes and sprite changes to mining state.
	2	Player holds key until resource collected.	After resource is mined system displayed collected resource, adds it to storage, and replaces resource object with regular asteroid.
	3.	Player chooses another action	
Variations	2. if storage is full, display pops up that the storage is full and resource could not be collected.		

Activity Diagram

UR-01: Player Sprite Movement

As the player, I want to be able to move through space with my jetpack.

Use Case: In Game Controls: David Kleckner



Sequence Diagram:

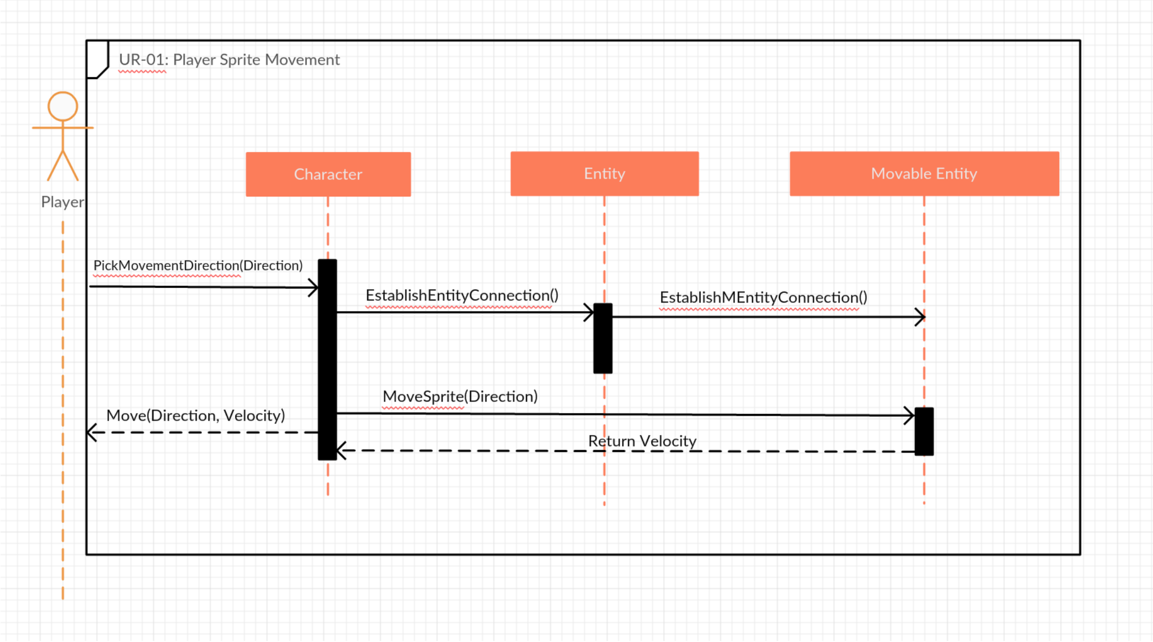
UR-01: Player Sprite Movement

As the player, I want to be able to move through space with my jetpack.

Use Case: In Game Controls: allows the player basic movement control over their sprite.

1. Player Enters preferred direction of movement, which is input into Character class.
2. Character then establishes connection with entity and movable Entity Class, allowing for Character to access MoveSprite().

3. After calling MoveSprite(), it returns the velocity of the sprite, which is then implemented in-game.


UR-01: Player Sprite Movement
Player
Character
Entity
Movable Entity
PickMovementDirection(Direction)
EstablishEntityConnection()
EstablishMEntityConnection()
MoveSprite(Direction)
Move(Direction, Velocity)
Return Velocity