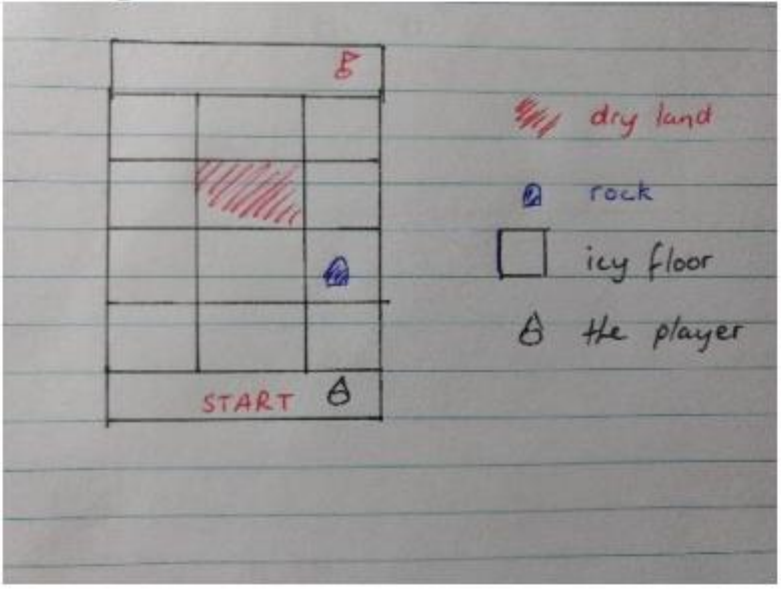
Design Overview for <<Slide>>

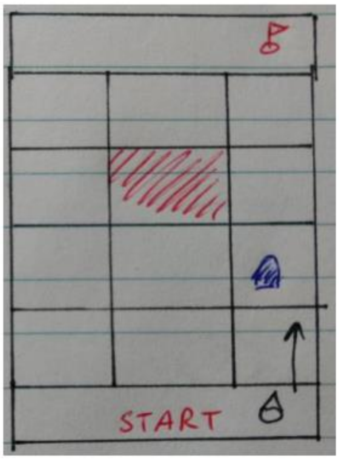
Name: Klim Huynh
Student ID: 101634015

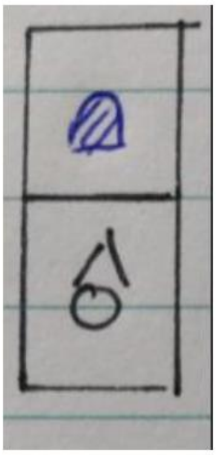
Summary of Program

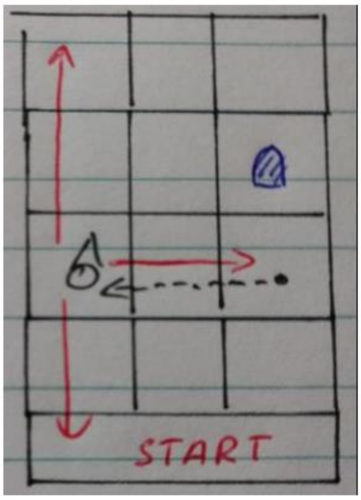
The purpose of the game is to help the player get from point A to point B. A major of the surface is ice, which is slippery and will not allow the player to stop moving until they have either hit the edge, an object or somewhere dry.

This is a rough overview of what the game will look like.

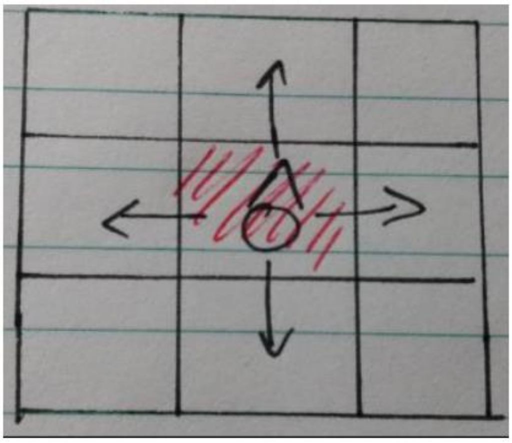






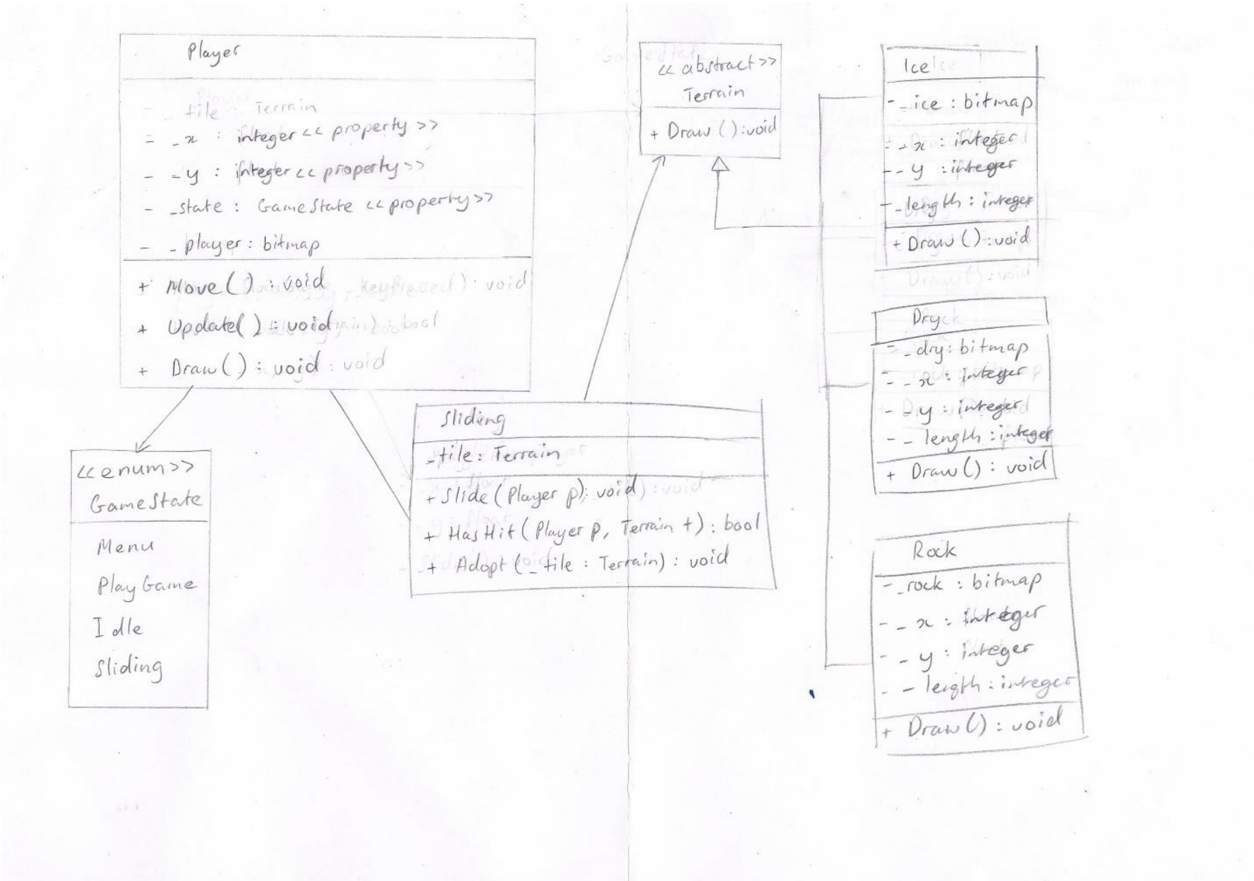


The player will continue to slide through the direction the chose until they hit something.

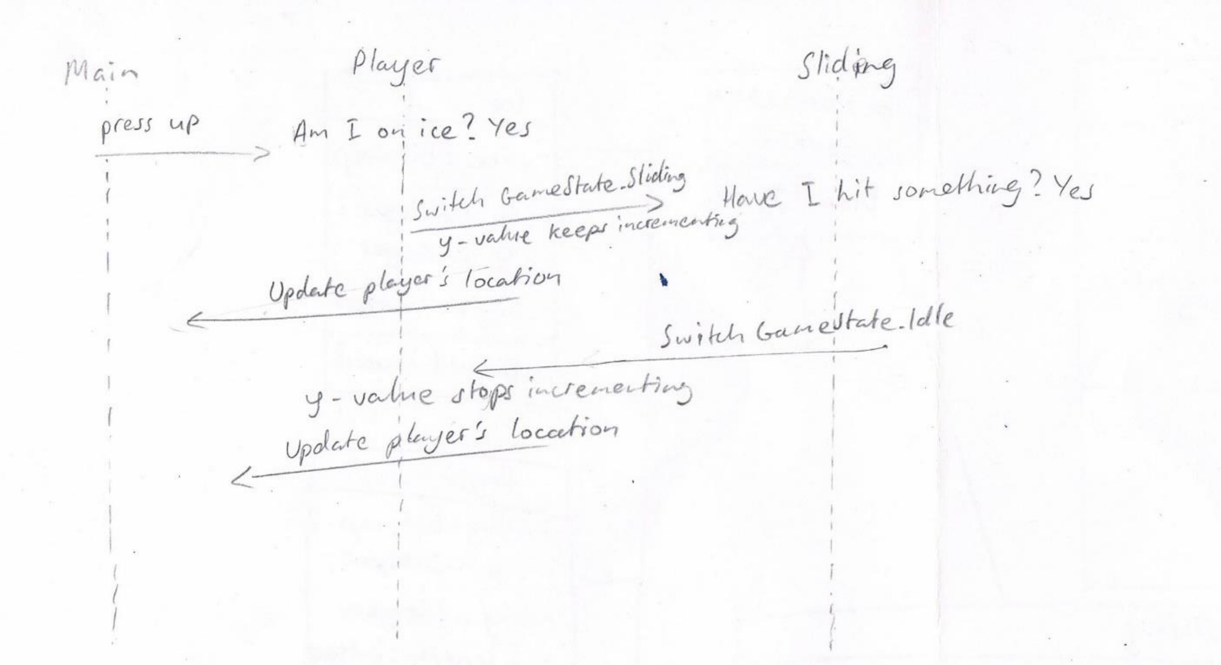


Hitting dry land will allow the player to stop and give them the option of sliding in a different direction.

UML Diagram



Sequence Diagram



Required Roles

Describe each of the classes and interfaces you will create using the following table (one per record).

Table 1: <<Player>> details

Responsibility	Type Details	Notes
	Field type, parameter and return types	
Player	Parameter: integer_x, _y GameState _state = GameState.Idle Bitmap _player; Terrain _tile	This is the constructor.

X	Get {return _x;}
	Set {_x = value;}
		Read and write property

Y	Get {return _y;}
	Set {_y = value;}
		Read and write property

State	Get {return _state;}
	Set {_state = value;}	

Move		Switches the game state to sliding. Passes player to sliding which moves the player.
Update		Update the player on screen, so you can see the player moving.
Draw	Parameter: integerx, y	Draw the player

Table 2: <<Sliding>> details

Responsibility	**Type Details**	**Notes**
	Field type, parameter and return types	
Adopt	Parameter: Terrain t	Adds the different terrains and their x and y values into a list.
Slide	Parameter: Player p	Increments either the x or y value depending on which directional button is pressed
HasHit	Parameter: Player p, Terrain t	Checks to see if the player has hit any dry terrain or rocks. If it has hit either dry terrain or rocks, switch the game state to idle, stopping the player from moving any further.

Table 3: <<Terrain (abstract class)>> details

Responsibility	**Type Details**	**Notes**
	Field type, parameter and return types	
Terrain	Integer _x, _y	This is the constructor.
Draw	Integer _x, _y	Abstract method

Table 4: <<Ice>> details

Responsibility	**Type Details**	**Notes**
	Field type, parameter and return types	
Ice	Base: (integer _x, _y) Bitmap _ice	This is the constructor. It inherits from terrain class.
Draw	Integer _x, _y	Draws the bitmap of ice at the coordinate

Table 5: <<Dry>> details

Responsibility	**Type Details**	**Notes**
	Field type, parameter and return types	
Dry	Base: (integer _x, _y) Bitmap _dry	This is the constructor. It inherits from terrain class.
Draw	Integer _x, _y	Draws the bitmap of dry land at the coordinate

Table 6: <<Rock>> details

Responsibility	Type Details	Notes
	Field type, parameter and return types	
Rock	Base: (Integer x, y) Bitmap _rock	This is the constructor. It inherits from terrain class.
Draw	Integer _x, _y	Draws the bitmap of a rock at the coordinate

Table 7: << GameState enum>> details

Value	Notes
Menu	Draw menu
PlayGame	Clear the screen and draw the game
Idle	Stops the player from moving. Update the bitmap
Sliding	Depending on which direction only either the x-coordinate or y-coordinate changes. Update the bitmap