Carrie Davis

Intro Programming 162

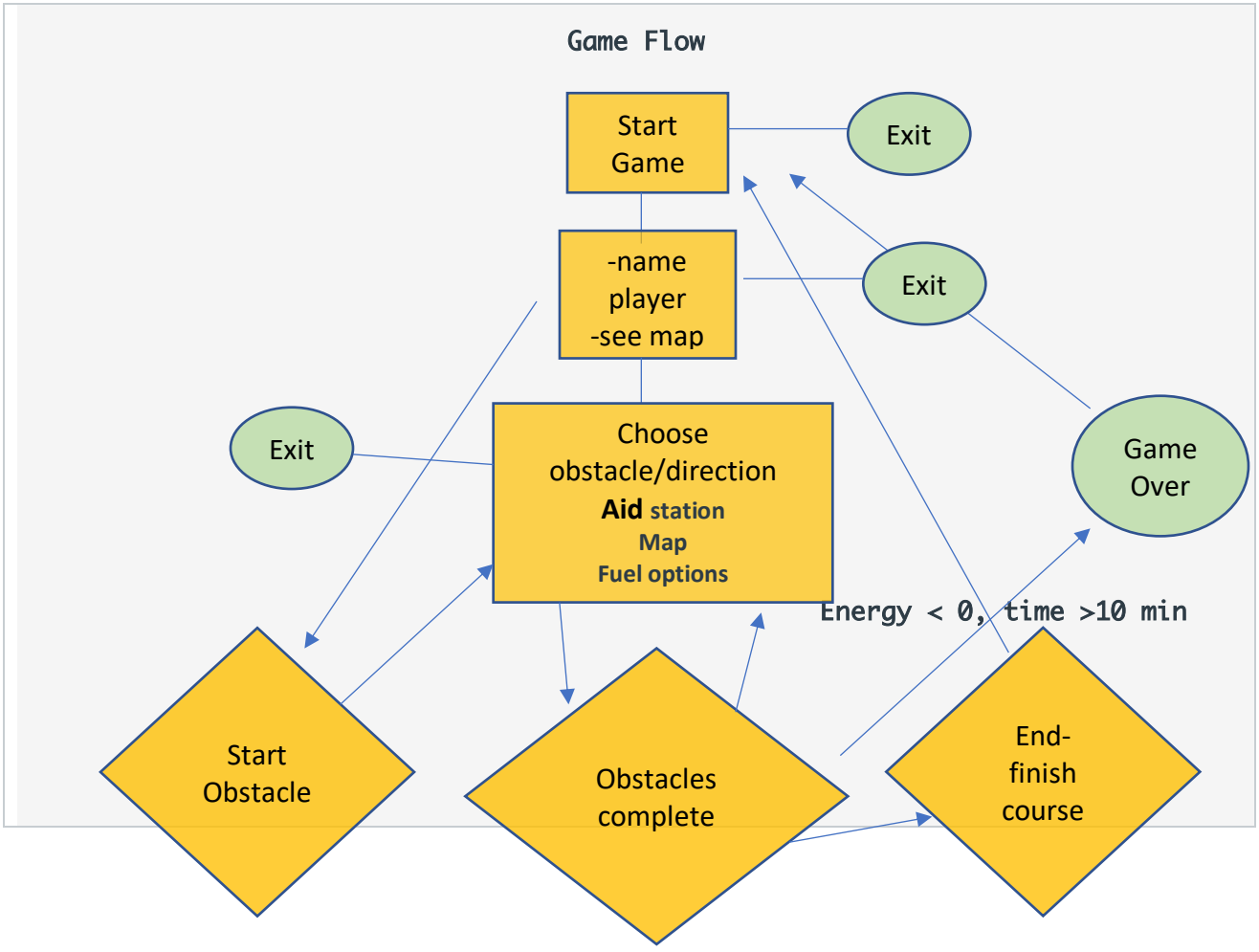
Final Project Plan

Obstacle course Game Criteria

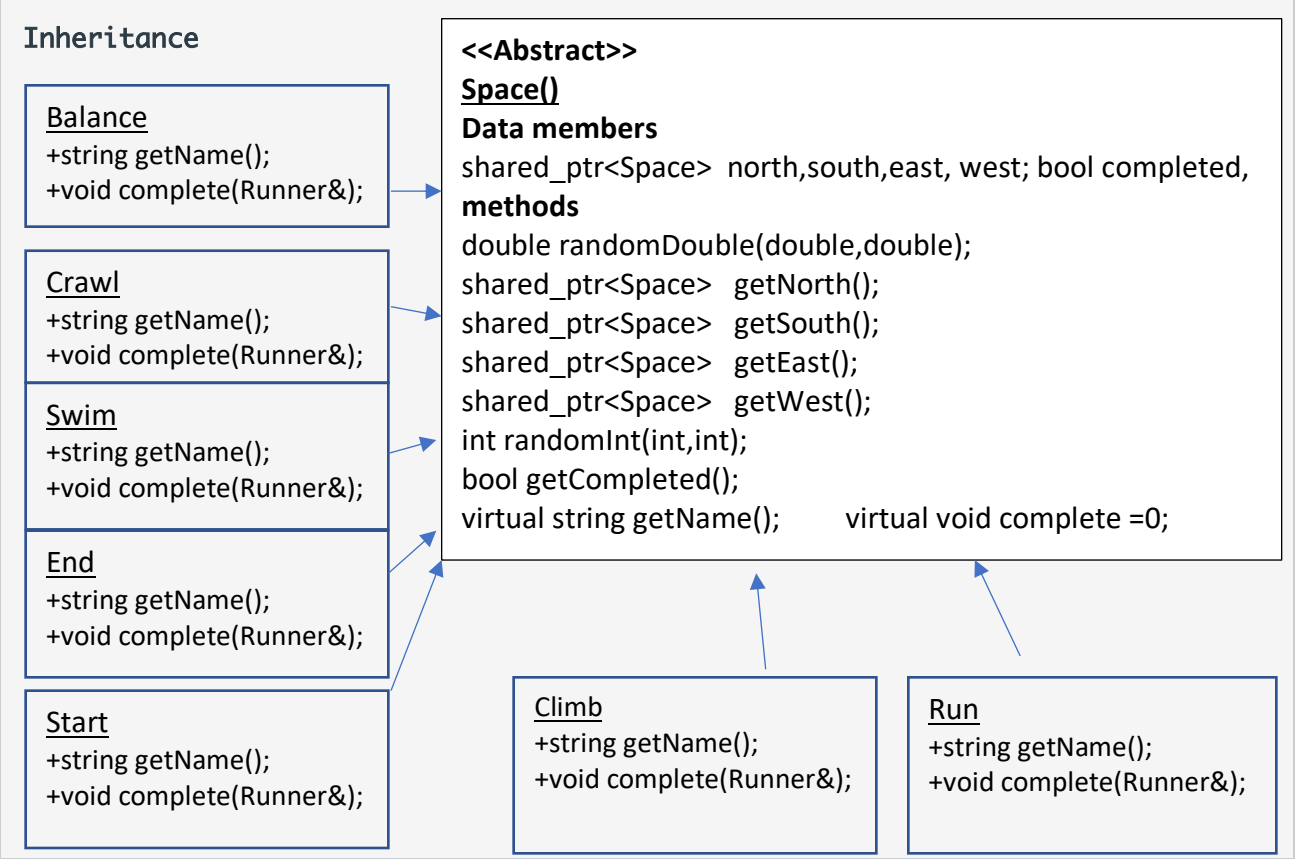
Obstacle course game is a one player game that requires the player to get from the Start location on a 2D board to the End location. Obstacles created by random must be completed to get to the end. The players time is limited to complete. Energy is also limited. Each obstacle takes a randomized amount of time. Time is dependent on players energy levels.

Criteria to Win.

1. Runner must move on map from Start to Finish.
2. Runner must finish in 10 minutes.
3. Runner cannot run out of energy.



Space Specification



Start Game: main()

Prompt user if start program or quit

input user choice

do while choice == 1 loop

1 → start program //creates instance of ObstacleCourse and calls run();

2 → Do nothing and quit

ObstacleCourse::run method called

setUpCourse method //creates 2D vector of start/end & random Space objects

10x4

Rumner object created

Input runner name

Explain course goals/rules

Do {

   Print take on challenge

      Print 1. Yes

      Print 2. See Map

      Print 3. Exit Game

Input choice

1 → sets rows and column for runner [0][0] at start

 - completes Start Space and outputs info on intial energy/time for Runner

ObstacleCourse::navigateCourse method called

2 → prints 2D Vector of Start and End location “*” are for spaces not completed yet.

3-> exits program

- **NavigateCourse(Runner&)**

Do {

Displays the Spaces that are north, south, east, west based on runner’s location

Print 1. North “getNorth Space Name”

Print 2. South “getSouth Space Name”

Print 3. East “getEast Space Name”

Print 4. West “getWest Space Name”

Print 5. See Map. //2D Vector with Names of Spaces completed & start/end

Print 6. See Stats //runner methods to get status of energy/time

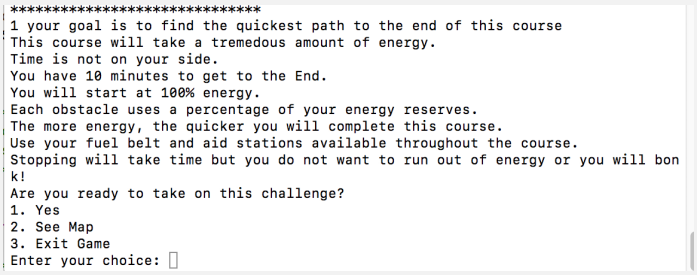
Print 7. Use Fuel Belt // menu and then select fuels available in Fuel vector

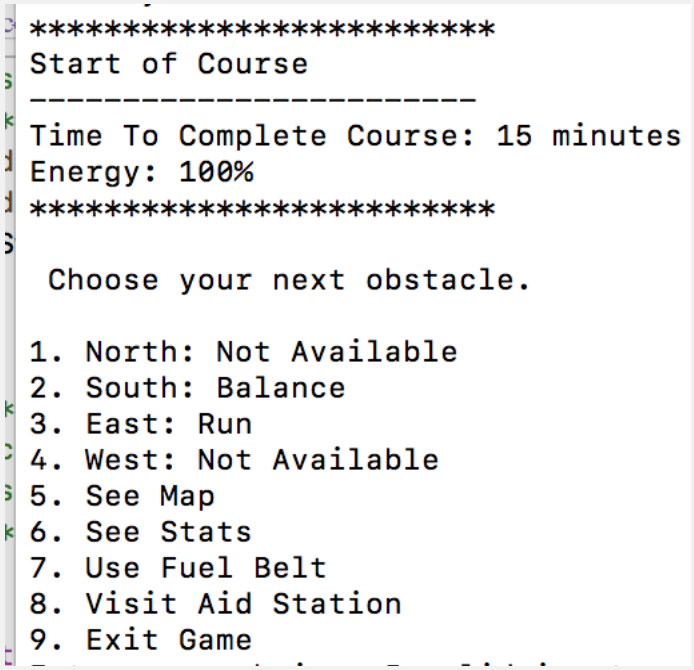
Print 8. Visit Aid Station //menu and select Gummies, water or Gel to add to belt (only carry 3)

Print 9. Exit Game

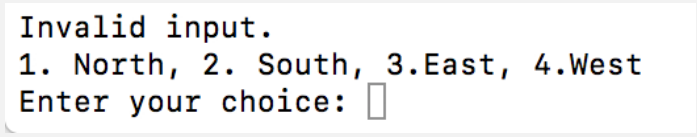
}While Runner energy >0 && End != completed && runner time > 0

Obstacle Course Test

Tests	*Action performed*	*Expected output*
Test1:	runnerName Entered 1	
	choice Entered @#1	Invalid Entry Are you ready to take on this challenge? 1. Yes 2. See Map 3. Exit Game Enter your choice:
	choice Entered 2.5	**choice:** 2 //removes decimal values
		Enter your choice: 2.5 ------Map------- Start * End Are you ready to take on this challenge? 1. Yes 2. See Map 3. Exit Game Enter your choice:
	choice Entered 100	Invalid Entry Are you ready to take on this challenge? 1. Yes 2. See Map 3. Exit Game Enter your choice:
	choice Entered 1.4$	**choice:** 1 //removes decimal values and char

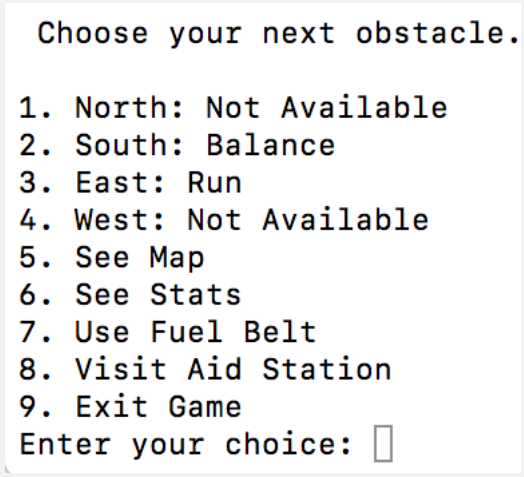


dirChoice Entered $3



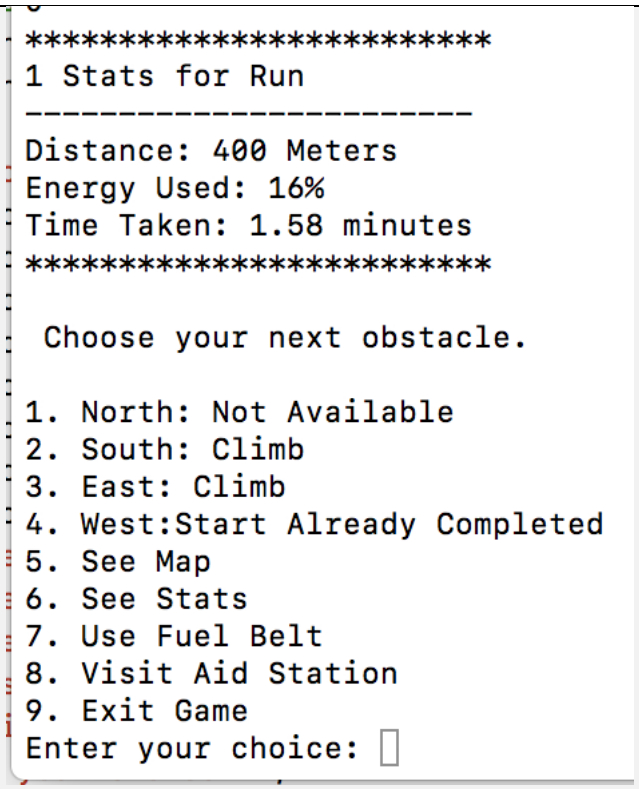
dirChoice Entered 1$%##@

dirChoice 1 //nothing will happen



dirChoice Entered 3

dirChoice: 3 EAST



dirChoice Entered 3

dirChoice: 3 EAST

```
*************************
1 Stats for Climb
-----------------------
Distance: 50 Meters
Energy Used: 18%
Time Taken: 1.15 minutes
*************************

 Choose your next obstacle.

1. North: Not Available
2. South: Swim
3. East: Swim
4. West:Run Already Completed
5. See Map
6. See Stats
7. Use Fuel Belt
8. Visit Aid Station
9. Exit Game
Enter your choice:
```

dirChoice: 3 EAST

dirChoice 3 Entered

```
*************************
1 Stats for Swim
-----------------------
Distance: 50 Meters
Energy Used: 8%
Time Taken: 0.86 minutes
*************************

 Choose your next obstacle.

1. North: Not Available
2. South: Run
3. East: Not Available
4. West:Climb Already Completed
5. See Map
6. See Stats
7. Use Fuel Belt
8. Visit Aid Station
9. Exit Game
Enter your choice:
```

dirChoice 3 Entered

dirChoice: 3 //Nothing will happen and screen will display again bc out of bounds

```
Enter your choice: 3

 Choose your next obstacle.

1. North: Not Available
2. South: Run
3. East: Not Available
4. West:Climb Already Completed
5. See Map
6. See Stats
7. Use Fuel Belt
8. Visit Aid Station
9. Exit Game
Enter your choice:
```

dirChoice 5$

dirChoice 5 Map

```
Uncompleted Obstactles show up *
----------Map Course ----------
  Start   Run   Climb   Swim
  *     *     *     *
  *     *     *     *
  *     *     *     *
  *     *     *     *
  *     *     *     *
  *     *     *     *
  *     *     *     *
  *     *     *     *
  *     *     *     End
---------------------------------

 Choose your next obstacle.
```

dirChoice 6

dirChoice 6 Stats

```
Stats for 1
-----------------------
Time Available to Complete Course: 11.41 minutes
Energy Available: 58/100
```

	dirChoice 7*	**dirChoice** 7 Fuel
		`Enter your choice: 7*`
		`You are out of snacks!`

	dirChoice 8	**dirChoice** 8 Aid Station
		`Welcome to the Aid Station`
		`Pick Energy Sources for your Fuel Belt`
		`1. Gummies`
		`2. Water Bottle`
		`3. Energy Gel`
		`4. Leave Aid Station`
		`Enter your choice:`

	energyChoice 5	`Invalid input.`
		`1. Gummies`
		`2. Water Bottle`
		`3. Energy Gel`
		`4. Leave Aid Station`
		`Enter your choice:`

	energyChoice 4	**energyChoice** 4 //exit menu
		`Choose your next obstacle.`
		`1. North: Not Available`
		`2. South: Run`
		`3. East: Not Available`
		`4. West:Climb Already Completed`
		`5. See Map`
		`6. See Stats`
		`7. Use Fuel Belt`
		`8. Visit Aid Station`
		`9. Exit Game`
		`Enter your choice:`

	dirChoice 2$	**dirChoice** 2 //south

```
**************************
1 Stats for Run
-----------------------
Distance: 400 Meters
Energy Used: 23%
Time Taken: 1.63 minutes
**************************

 Choose your next obstacle.

1. North:Swim (Already Completed)
2. South: Balance
3. East: Not Available
4. West: Swim
5. See Map
6. See Stats
7. Use Fuel Belt
8. Visit Aid Station
9. Exit Game
```

dirChoice 6^ Entered

dirChoice 6

```
Stats for 1
-----------------------
Time Available to Complete Course: 9.77 minutes
Energy Available: 35/100
```

dirChoice Entered 8

dirChoice 8 // Aid Station

energyChoice Entered 2%

energyChoice 2 //Water

```
Enter your choice: 2%
Water is an essental choice.
You picked: Water
Energy Available: 10%

 Choose your next obstacle.
```

dirChoice Entered 7

dirChoice 7 //fuel belt

```
1: Water
2: Empty
3: Empty
4. Select Nothing
```

fuelChoice Entered 2

```
You picked: Water
Energy added: 10

 Choose your next obstacle.

1. North: Not Available
2. South: Balance
3. East: Not Available
4. West:Climb Already Completed
5. See Map
6. See Stats
7. Use Fuel Belt
8. Visit Aid Station
9. Exit Game
Enter your choice:
```

dirChoice Entered 6

```
Stats for 1
-----------------------
Time Available to Complete Course: 11.72 minutes
Energy Available: 91/100
```

dirChoice Entered 2

dirChoice 2 //south

dirChoice Entered 2

dirChoice 2 //south

dirChoice Entered 5

```
  Start    Balance    Climb    Run
  *        *          *        Balance
  *        *          *        Climb
  *        *          *        *
  *        *          *        *
  *        *          *        *
  *        *          *        *
  *        *          *        *
  *        *          *        *
  *        *          *        End
-----------------------------------
```

dirChoice Entered 2

dirChoice 2 //south

dirChoice Entered 2

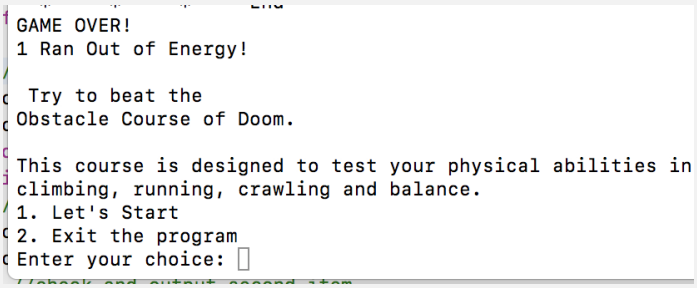
dirChoice 2 //south

dirChoice Entered 2

dirChoice 2 //south

dirChoice Entered 2

dirChoice 2 //south

	dirChoice Entered 2	**dirChoice** 2 //south
	dirChoice Entered 2	**dirChoice** 2 //south
	dirChoice Entered 2	
	choice Entered 2	*Note: Randomization in the game results will not be the same every time. **Choice:** 2 // game exits

Example of possible Obstacle options
Set up and completion of Obstacles
- Random obstacles will be displayed and will impact time/energy
- < energy > time takes to complete

Menu below only see options that are north,south,east, west and will list if completed or unavailable for bounds.

1: North : Climb the barrier
2: South : Swim 50 yards
3: East : Run 500 yards
4: West: Slide down Hill/embankment

You have picked obstacle Hill Slide.
This will be quick and dirty.
Do you want to take a supplement?
1: Yes
2: No
You survived the Hill Slide.
Time used: xxx
Points earned: xxx

Reflection

I decided with this project to use only containers and non-raw pointers. I used the following code found on https://stackoverflow.com/questions/13237490/how-to-use-a-2d-vector-of-pointers to implement and display my map of the obstacle course .

```
// C++ code to demonstrate 2D vector
#include <iostream>
#include <vector> // for 2D vector
using namespace std;

int main()
{
    // Initializing 2D vector "vect" with
    // values
    vector<vector<int> > vect{ { 1, 2, 3 },
                               { 4, 5, 6 },
                               { 7, 8, 9 } };

    // Displaying the 2D vector
    for (int i = 0; i < vect.size(); i++) {
        for (int j = 0; j < vect[i].size(); j++)
            cout << vect[i][j] << " ";
        cout << endl;
    }
```

I initially tested my menu with Space as an object, not just abstract base before creating the derivatives. I wanted to use pointers and not use “new” in order to not have to deal with garbage collection. A big issue on this project with Space data members north,south,east and west pointers by trying to use unique pointers.

```
  ObstactleCourse line 131
- Original
        shared_ptr<Space> outPtr = unique_ptr<Swim>();
     return outPtr;
    }
- Final
     shared_ptr<Space> outPtr = make_shared<Swim>();
     return outPtr;
    }
```

I found help using StackOverFlow search.
https://stackoverflow.com/questions/41060167/strange-error-use-of-deleted-function-stdunique-ptr-tp-dpunique-ptr-wh

It was challenging at first to learn the STL with vectors. I spent some time working with doing out of bounds checks. I originally thought that the function was built in and the out of bounds values would automatically come up null or 0. I was able to use Try/Catch to implement bounds checking in a much better manner than done in previous labs and projects. Try Catch exceptions worked really well as a solution and I hope to use this more in future projects.

```
Space line 44
    try {
        north = inVector.at(inRow - 1).at(inCol);
        if(north->getCompleted() == false)
        {
            cout << "1. North: " << north->getName() <<endl;
            return north;
        }
        if(north->getCompleted() == true)
        {
```

```
            cout << "1. North:" << north->getName() <<" (Already Completed)"
<<endl;
            return north;
        }
    } catch (const out_of_range& oor) {
        cout << "1. North: Not Available" <<endl;
        return NULL;
    }
```

Overall, I am happy with my design, but because of time constraints I am not able to add the extras that I would like and include the extras that might make the game actually fun. This project was a great learning experience in design and the use of containers. I was able to think of different solutions than in the past.