گزارش کار:

پیدا کردن هدف در جهان جدولی بادی الگوریتم QLearning

مسئله:

یک عامل در یک جدول 10*7 از مکان شروع وارد می شود و باید به مکان هدف برسد. عامل می تواند برای حرکت کردن یکی از عملیات بالا، پایین، چپ یا راست را انتخاب کند. در هر ستون ممکن است باد با شدت معینی بوزد که حرکت عامل را منحرف کند اما عامل از آن باخبر نیست، لذا عامل پس از انتخاب عمل مورد نظر، عمل انتخاب شده و مکان فعلی خود را به تابع step می دهد و این تابع بر اساس عمل انتخابی و شدت باد در آن ستون مکان جدید عامل را به همراه پاداش یا جریمه بر می گرداند. عامل می بایست با حرکت در این جدول و دریافت پاداش و جریمه ها یاد بگیرد که با بالاترین پاداش از نقطه شروع به هدف برسد. پاداش برای هدف 50 و جریمه برای دیگر مکان ها 1- می باشد.

حل مسئله:

از آنجایی که برنامه با زبان C# نوشته شده است تابع step نیز از Python به این زبان برگردانده شده است:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace WindyGridWorld
{
    public class Windy
    {
        public static int WORLD_HEIGHT = 7;
        public static int WORLD_WIDTH = 10;

        public static int[] WIND = { 0, 0, 0, 1, 1, 1, 2, 2, 1, 0 };

        public const int ACTION_UP = 0;
        public const int ACTION_DOWN = 1;
        public const int ACTION_LEFT = 2;
        public const int ACTION_RIGHT = 3;

        static double REWARD = -1.0;

        public static int[,] START = { { 3, 0 } };
        public static int[,] GOAL = { { 3, 7 } };


        static int[] ACTIONS = { ACTION_UP, ACTION_DOWN, ACTION_LEFT, ACTION_RIGHT };

        public static StepResult step(int[,] state, int action)
        {
            int i = state[0, 0];
            int j = state[0, 1];
```

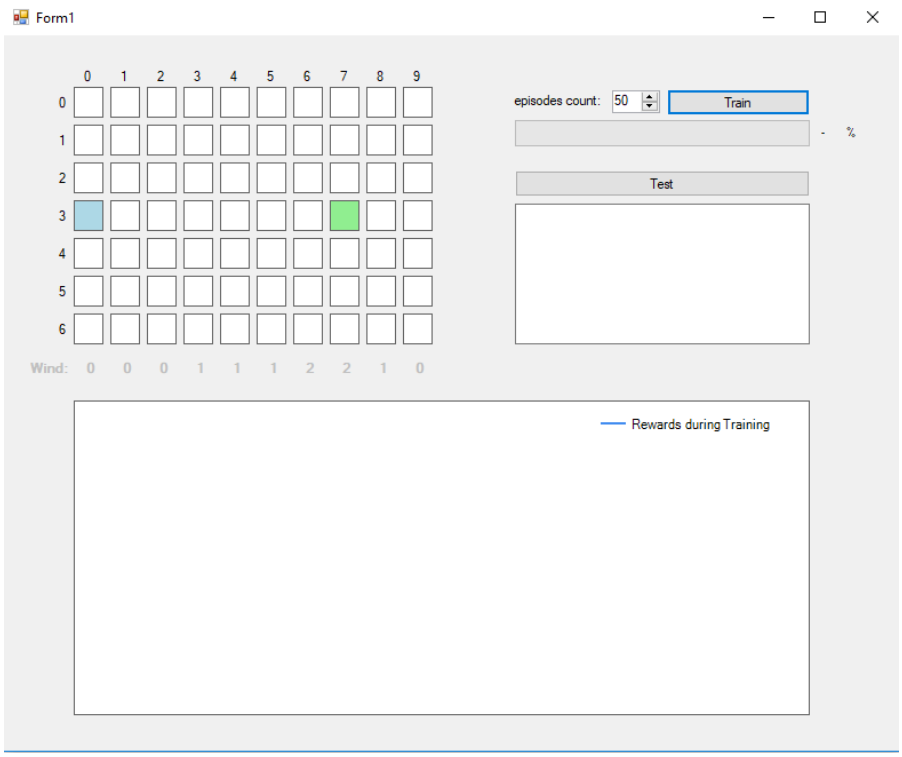
```
            if (action == ACTION_UP)
            {
                state = new int[,] { { Math.Max(i - 1 - WIND[j], 0), j } };
            }
            else if (action == ACTION_DOWN)
            {
                state = new int[,] { { Math.Max(Math.Min(i + 1 - WIND[j], WORLD_HEIGHT -
1), 0), j } };
            }
            else if (action == ACTION_LEFT)
            {
                state = new int[,] { { Math.Max(i - WIND[j], 0), Math.Max(j - 1, 0) } };
            }
            else if (action == ACTION_RIGHT)
            {
                state = new int[,] { { Math.Max(i - WIND[j], 0), Math.Min(j + 1,
WORLD_WIDTH - 1) } };
            }

            double reward = REWARD;

            if (state[0, 0] == GOAL[0, 0] && state[0, 1] == GOAL[0, 1])
            {
                reward = 50;
            }

            return new StepResult() { State = state, Reward = reward };
        }

    }
}
```

* تمامی فاکتورهای مسئله مثل نقطه شروع و پایان، شدت باد و اندازه جدول و پاداش و جریمه ها قابل مقداردهی و تغییر هستند.

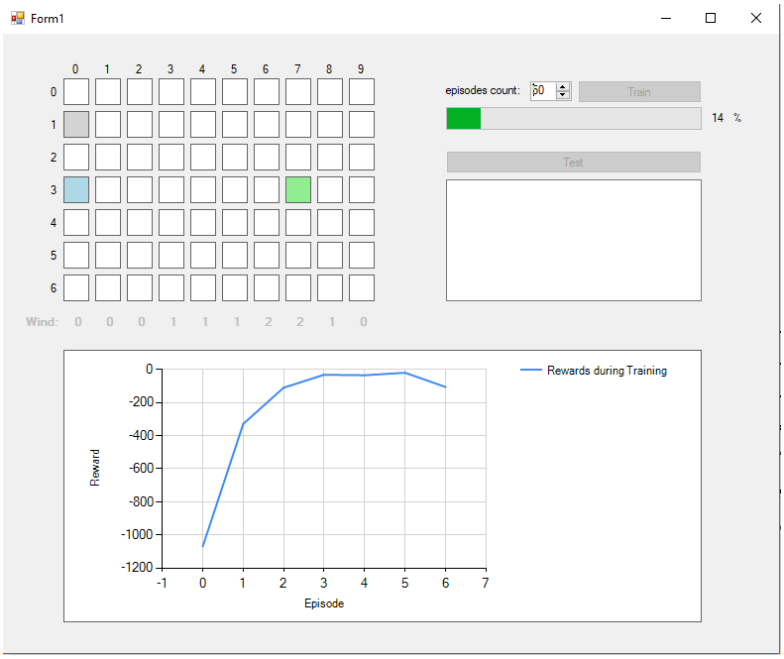
آموزش:

از آنجایی که روش حل مسئله QLearning می باشد می بایست یک ماتریس Q داشته باشیم. به این صورت که یک ماتریس سه بعدی برای طول، عرض و عملیات، برای کیفیت هر عمل در هر مکان در نظر گرفته می شود که در این مسئله اندازه ماتریس برابر 4*10*7 است.

در زیر هر ستون شدت باد نمایش داده شده است.



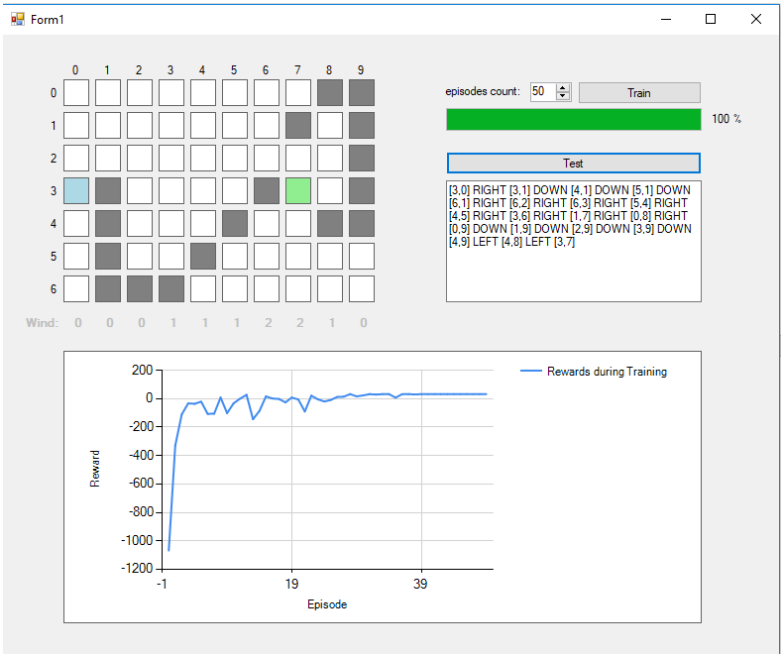
در برنامه می توان تعداد دفعات مورد نظر برای یادگیری را اتخاب نمود و سپس دکمه Train را کلیک کرد عامل از مکان شروع وارد می شود. مکان شروع با رنگ آبی و مقصد با رنگ سبز مشخص شده است. عامل در هر مرحله بر اساس الگوریتم QLearning عملیات انتخابی و مکان فعلی خود را به تابع step می دهد و پاداش یا جریمه و مکان بعدی خود را از آن می گیرد و بر اساس پاداش داده شده ماتریس Q را بروز رسانی می کند. مکان عامل با رنگ خاکستری نمایش داده می شود. این عملیات آنقدر تکرار می شود تا عامل به مقصد برسد در نتیجه یک دور یادگیری به پایان می رسد. این عملیات تا به تعداد دفعات یادگیری تکرار می گردد. در هر دفعه از مراحل یادگیری با رسیدن عامل به مقصد جمع پاداش های داده شده در نموداری نمایش داده می شود تا کیفیت یادیگیری عامل در هر مرحله قابل مقایسه باشد. در برنامه مکان عامل در جدول در هر لحظه نمایش داده می شود.



تست:

پس از اتمام دفعات یادگیری همانطور که گفته شد ماتریس Q یا کیفیت عملیات در هر مکان آماده شده است. پس عامل در هر مکان بر اساس انتخاب بالاترین مقدار عملیات برای آن مکان عمل مورد نظر خود را انتخاب می کند و این کار را تا رسیدن به مقصد تکرار می کند.

مسیر انتخابی عامل تست در جدول علامتگزاری می شود و همچنین مکان و عملیات انتخاب شده در مسیریابی نوشته می شود.



فیلم کامل مراحل یادگیری و تست در فولدر برنامه موجود می باشد. همچنین می توانید برنامه را از مسیر WindyGridWorld/bin/Debug/WindyGridWorld.exe اجرا نمایید.