|  |
| --- |
| Close-up image showing the leaf-sides of two oversized books side-by-side on a bookshelf, with additional books in soft focus background |
| **Othello Guide**  ***"Expect the best, plan for the worst, and prepare to be surprised." - Denis Waitley*** |
| |  |  |  | | --- | --- | --- | | Keivan Ipchi Hagh | 4/6/20 | AP Project – Part 1 | |

Table of Contents

[مشخصات ابتدایی پروژه 2](#_Toc37147660)

[دستور العمل (نحوه تعامل با بازی) 2](#_Toc37147661)

[نوع و کاربرد کلاس های استفاده شده 2](#_Toc37147662)

[الگوریتم موتور بازی 3](#_Toc37147663)

[الگوریتم هوش مصنوعی بازی () 3](#_Toc37147664)

[شیء گرا بودن و استفاده از مفهوم Encapsulation, Encapsulation, Inheritance 3](#_Toc37147665)

[نمایش گرافیکی vs کنسول 4](#_Toc37147666)

[چند نکته مهم! 4](#_Toc37147667)

# مشخصات ابتدایی پروژه

|  |  |
| --- | --- |
| عنوان پروژه | بازی اوتلو |
| استاد درس | حسین زینلی |
| تاریخ تحویل | هجده فروردین 1399 |
| نویسنده | کیوان ایچپی حق – 9831073 |

# دستور العمل (نحوه تعامل با بازی)

* ابتدا، نحوه بازی را مشخص میکنید (singleplayer, multiplayer, ai\_vs\_ai). در ai\_vs\_ai دو الگروریتم بازی مقابل هم قرار میگیرند!!!!
* ابتدا بازیکن/دو بازیکن، اسامی خود را به بازی میدهند (بنابر نحوه بازی)
* اگر بازی تک نفره بود سختی AI مشخص میکنید (easy, medium, hard)
* سپس، نحوه نمایش بورد را انتحاب میکنند (اطلاعات بیشتر راجب IDE را در پایین حتما بخوانید!)
* console
* graphical
* ثبت اطلاعات تمام شده و بازی شروع میشود، هر بازیکن مختصات انتخابی خود را به صورت

<[1-8]> <[A-D]>وارد میکند. ورودی حساس به بزرگ/کوچک بودن نیست. عدد وارد شده شماره ردیف و حرف، شماره ستون را نشان میدهد. هر ورودی که به فرم بالا نباشد ارور برمیگرداند و دوباره پرسیده میشود.

# نوع و کاربرد کلاس های استفاده شده

پروژه از تعدادی *class* تشکیل شده که هرکدام مسئول انجام بخشی از عملیات مربوط به بازی است. بازی از یک بورد تشکیل شده که در *Board* به صورت ماتریس ذخیره شده و تعدادی تابع مسئول پرینت کردن این ماتریس به صورت کنسول یا گرافیکی (بنابر خواسته کاربر) است. هر عضو این ماتریس شیئی از نوع *“Cell”* است. هر یک از این عضوها، دارای مقدار، رنگ، نوع و مختصات است. مقدار یک *ASCCI CODE* برای نشان دادن دایره توپر، نوع در اصل *Enum بوده که نوع سلول را مشخص میکند. دو شی Player که هریک، اطلاعات هر بازیکن نظیر اسم و امتیاز و خانه های بورد مختص به او را شامل میشوند برای ذخیره اطلاعات کاربران وجود دارد. در کلاس Game الگوریتم اصلی بازی برای انجام تغییرات در بورد وجود دارد و کلاس استاتیک ConsoleColors کد تمام رنگ های قابل استفاده را نگهداری میکند.*

# الگوریتم موتور بازی

*موتور بازی یک الگوریتم ثابت داشته که برای دو کار 1. تغییر نوع خانه های تحت تاثیر قرار گرفته توسط بازیکن 2. نشان دادن انتخاب های موجود بازیکن، استفاده میشود. با استفاده از Brute Force تمام خانه های مجاور (شمال، شرق، شمال شرق و ...) چک میشود تا در صورت امکان عملیات موبوطه روی انجام شود.*

# الگوریتم هوش مصنوعی بازی

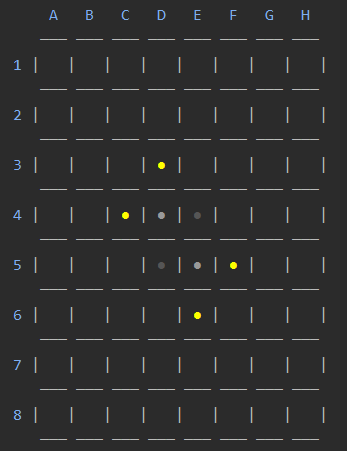
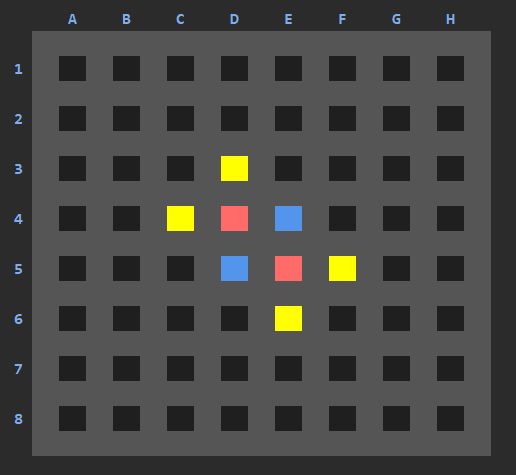
کامپیوتر سه حالت برای بازی دارد:

* حالت ساده (easy): در این حالت کامپیوتر یکی از حرکت های ممکن را به صورت تصادفی انتخاب و اجرا میکند.
* حالت متوسط (medium): این الگوریتم نسخه ساده شدهAlpha Beta Search Algorithm (MinMax) است که تمام حرکت های خود و حریف را چک کرده. بهترین حرکت (حرکتی که در آن خودمون بیشتری امتیاز و حریف کمترین امتیاز رو بگیره) انتخاب و اجرا میشه.
* حالت سخت (hard): با استفاده از چند تکنیک و الگوریتم هوش مصنوعی پیاده سازی شده که از حالت متوسط هوشمندتر است. (Perceptron, Activation Functions, Layers, …)

# شیء گرا بودن و استفاده از مفهوم **Encapsulation, Encapsulation, Inheritance**

کلاس های *Player, Cell, Board, Game, CellType اشیاء موجود در بازی هستند و باهم تعامل دارند، که نشانگر استفاده از مفهوم شیء گرایی است. همچنین اینکه اکثر توابع و متغیر های مربوط به یک شی درون آن به صورت یک پک نوشته شده و هیچ کلاسی اطلاعی از محتویات کلاس های(اشیا) دیگر ندارد. کلاس AI توابعی برای پیش بینی حرکات بعدی خود و حرف دارد و برای این کار از برخی توابع موتور بازی استفاده میکند. برای کمتر شدن کد از مفهمو ارث بری استفاده شده.*

# نمایش گرافیکی vs کنسول

**

# چند نکته مهم!

* بازی دارای دو الگوریتم مختلف برای نمایش بورد است:

1. الگوریتم نمایش بورد به صورت گرافیکی: Visual Code قادر به نمایش گرافیکی نیست!!! و حتما حتما در IntelliJ تست کنید!!! (اگر در این IDE از نمایش گرافیکی استفاده کردید و چرت و پرد نشان داده شد مشکل بازی نیست 😉)
2. الگوریتم نمایش بورد به صورت ساده (کنسول): Visual Code نمایش کنسول ساده بدون رنگبندی های پیچیده تر را نشان میدهد (Visual Code SAFE!)

* پروژه در GitLab بارگزاری شده! [لینک رو از کلیک کنید](http://git.ceit.aut.ac.ir/9831073/othello.git)
* بازی نمایش گرافیکی (رنگی) دارد (نمره امتیازی🤙)
* بازی هوش مصنوعی دارد (تقریبا) و سه سطح دارد برای باهوشی کامپیوتر (نمره امتیازی✌)

با تشکر – کیوان ایپچی حق - 9831073