

Projectsamenvatting IPASS

1010 0 10 0 0 1 1 111 0 0 0 1 1 111 0001

1 1 1101 0 1010 100100 0110 1 0 1 00110 101001001 0111010 1 10 0

0001011011 0 000000 0101001 110100101 0 1 00 001011 010000 0010 1 0

0 0 01 010 0010001 00 111 000 100 01 0100011 100 00110 00 1100 1010 1 1

11110 00011 00 110 10 010100 110 01000101 0111 0110 00 10 011010 100 1011

10110 11 0000 11100 1110 011110 111001 00 1 1011 00 1 000 00011 11 0

1000 1 01 110 10 00111101000 1 001 111001 11 110 0 10 1 100 11000 110 0

0 1 000 100 1 1 011 1000 0111 0111 000 011110 0000010 010 100 110 0 10000 0

0110 11100 01 1000 01101 1 01 110001 1110 11 011 11 11110 111 00 10 101 01100 001

0 1 1 0 10000 111000 1 01 010 10 110 10 01011 110 000 1 000 01 100000 0 10 0110

11 10 0010 11 1000 10 10 101 0 01010 0 11010 1100011 1 0111 111111 11000 1 11100 0

1000 10 0 00 01100 010001 0110 010001 0110 011 0001 0111 00110 101 00110 0010111

11000 0100 001 000 0 01011101 00 01011 011101000 1 0000 010 0 0 1100 0 1101

0000 1 11 0010110 0001 0 11 10111 100001 011010111 101 00001001 010100 0100000

0001000 10 010 110 0001 1101 101 0 001 0000110 1 011 0011 0000001000 10001010

1 10 01101 0111000 10 00 0 00 01 101100 01 0001001110100 0001001110100 1101 0

01 11 11 01 1010 1111 00001 00110 110000001 01 00011110 1101 1110 00 111011

00 00 0 1110 0100 0001011000 01 001011000 0 0100001 11 11000

0 1 0 11010 0 1101 0 1001 0 0

0 0 1 0 10010 1001 0

1 0

Door Justin van vliet

Datum: 28-06-2023

Inhoud:

1. probleem beschrijving
2. beschrijving eisen
3. beschrijving algoritme
4. voor- en nadelen van het gekozen algoritme
5. uiteindelijke algoritme
6. de game
7. Overwegingen
8. licenties

1. Probleem beschrijving:

De opdrachtgever krijg de laatste tijd kritiek opdat de AI van de vijanden in de games van zijn bedrijf te dom zijn en te makkelijk te lezen zijn, dus hij gaaf de opdracht om een nieuwe AI te maken.

2. Beschrijving eisen:

Hij wil deze AI ook in actie zien in een game, dus moet er ook een low-effort game eromheen gemaakt worden of toegevoegd worden op een al bestaande game. Voor als je zelf een game wilt maken moet er wel een beetje graphics in zitten. Ook moet de AI tenminste twee keer voorkomen in de game.

3. beschrijving algoritme:

Conitzer & Sandhol (2006, Springer Science + Business Media, LLC 2006)

<https://www.cs.cmu.edu/~conitzer/awesomeML06.pdf>

Deze bron heb ik gebruikt voor het project. De bron gaat over een complexe manier om een repeated prisoner's dilemma te coderen door middel geavanceerde wiskunde.

4. voor- en nadelen van het gekozen algoritme

Voordelen: er staat al code in het papier dus het is makkelijker te bergrijpen in tegen stelling tot een papier dat alleen maar tekst is.

Nadelen: de wiskunde is soms moeilijk om zomaar te begrijpen.

5. uiteindelijke algoritme

Het is mijn jammer genoeg niet gelukt om de code van mijn gekozen bron volledig na te maken, omdat het niet paste bij wat ik wou maken en omdat de code die in de tekst stond voor mij deel te moeilijk was om na te maken. Dus uiteindelijk heb ik een algoritme dat ongeveer hetzelfde doet maar niet hetzelfde is gecodeerd. Mijn algoritme kijkt naar welke zeten de speler al heeft gedaan en als de speler steeds dezelfde zeten doet dan zal het daar op een aanpassing maken zodat hij kan winnen. Als de speler dan niet steeds hetzelfde doet zal het algoritme random een zet doen uit een lijst met zetten.

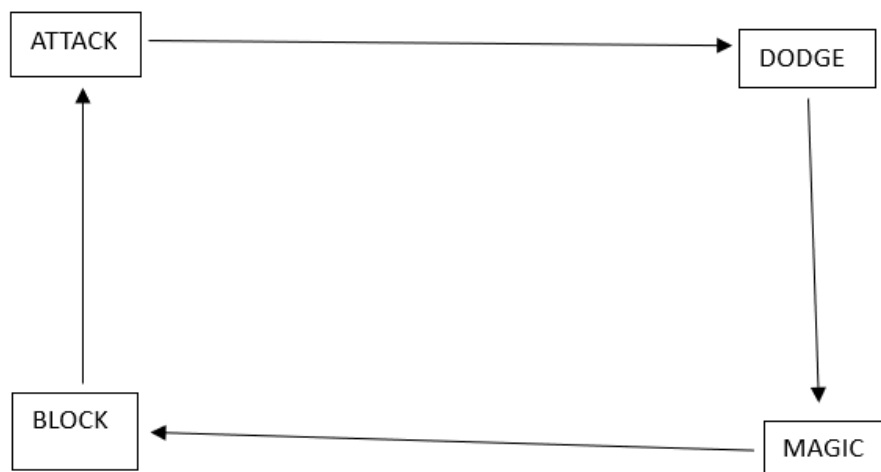
6. De game:

De game die ik dan om het algoritme heb gemaakt gaat als volgt. De speler moet de wereld redden door vijanden te verslaan om zo de grote baas te komen om die dan te verslaan. Dit hoe je die vijanden is door 1 van de 4 zeten te kiezen de vijand doet hetzelfde en dan is het afwachten welke zet wint, het is eigenlijk gewoon een grotere versie van steen papier schaar.

P = player

E = enemy

P\E	attack	<u>dodge</u>	block	<u>magic</u>
attack	draw	attack	block	draw
<u>dodge</u>	attack	draw	draw	<u>dodge</u>
block	block	draw	draw	<u>magic</u>
<u>magic</u>	draw	<u>dodge</u>	<u>magic</u>	draw



Dit is dan de game. Het is niet zoiets groots zoals undertale of andere grote games, maar in de opdracht stond dat het low-budget moest zijn.

7. Overwegingen:

Effectiviteit: de code werkt goed en effectief vanaf het begin dat ik het probeerde te maken. Dit was dan wel mij eigen versie van de code dus er waren weinig overwegingen over een andere algoritme te gebruiken en als ze er waren kwam dat, omdat ik steeds aan het proberen was de code uit de bron compleet na te maken.

Efficiëntie: over efficiëntie heb ik geen overwegingen gedaan. De code liep altijd snel en is redelijk goed opgesplitst om het efficiënter te maken en zodat ik niet een bestand het met 4000 regels code.

Privacy: volgens mij is er in mij project nergens iets dat de privacy van andere mensen die het gebruiken schendt dus er waren zowat geen overwegingen.

Morele keuzes: er zijn geen morele keuzes in de game dus en over de code zelf dus dit is voor mijn denk ik niet van toepassing

8. licenties(?):

de licentie heb ik in het Engels gedaan. niet omdat ik het gecopy + paste heb, mar omdat Engels een universele taal die zowat iedereen kan of moet begrijpen en zo is het makkelijker voor grotere groepen mensen ver de hele wereld te begrijpen.

Copyright (c) 2023 Justin van Vliet

Permission is hereby granted, free of charge, to any person
Obtaining a copy of this software and associated documentation
Files (the "Software"), to deal in the Software without
Restriction, including without limitation the rights to use,
Copy, modify, merge, publish, distribute, sublicense, and/or sell
Copies of the Software, and to permit persons to whom the
Software is furnished to do so, subject to the following
Conditions:

The above copyright notice and this permission notice shall be
Included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
OTHER DEALINGS IN THE SOFTWARE.