

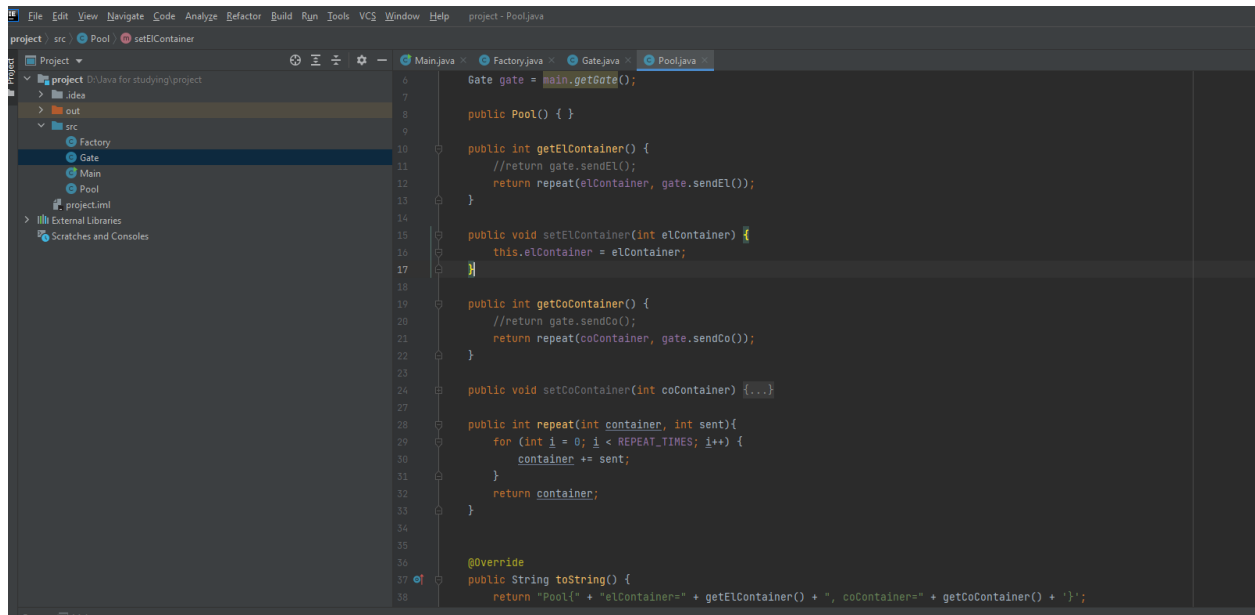
Spel projektets dokumentation

Denna gång också jag har skrivit alla kod själv. Jag har fått hjälp ut av webbsidor som förklarar saker om css, html och javascript (se referencer). Jag kan säga att jag har försökt mitt bäst att få allting rätt. Om jag började lite tidigare med projektet det kanske skulle hjälpa. Spelet heter Save The Earth. Jag hoppas att det som jag skrev är tillräckligt mycket för B.

Problem

1- utvecklingsprocessen var tröttsam. Jag hade tre planer...

Plan A var att koda spelets funktionalitet på JAVA och låta min HTML-fil läsa den. MEN!! det var ingen lätt uppgift alls. Jag har sökt på google, youtube och andra platser och det finns i princip inte mycket information som skulle hjälpa mig att länka java och html. Allt jag kunde hitta är något om "survet" och java.swing. förresten, spelet fungerar bra på java, men det finns ingen virtuell bara text (se bilden nedan).

A screenshot of an IDE (likely IntelliJ IDEA) showing a Java project. The project structure on the left includes 'project', 'src', 'out', and 'src' subfolders. The 'src' folder contains 'Factory', 'Gate', 'Main', and 'Pool' classes. The 'Pool.java' file is open in the editor, showing the following code:

```
1 Gate gate = main.getGate();
2
3 public Pool() {}
4
5 public int getElContainer() {
6     //return gate.sendEl();
7     return repeat(elContainer, gate.sendEl());
8 }
9
10 public void setElContainer(int elContainer) {
11     this.elContainer = elContainer;
12 }
13
14 public int getCoContainer() {
15     //return gate.sendCo();
16     return repeat(coContainer, gate.sendCo());
17 }
18
19 public void setCoContainer(int coContainer) {...}
20
21 public int repeat(int container, int sent){
22     for (int i = 0; i < REPEAT_TIMES; i++) {
23         container += sent;
24     }
25     return container;
26 }
27
28 @Override
29 public String toString() {
30     return "Pool{" + "elContainer=" + getElContainer() + ", coContainer=" + getCoContainer() + "}";
31 }
```

Plan B var att försöka kopiera koden som jag har skrivit i java och använda den i Unity. Eftersom Unity använder C# som är väldigt likt java. MEN!! Det kräver också att man förstår hur objekt fungerar i Unity. Jag har glömt mycket av det så jag gick till plan C.

Plan C är att i princip använda javascript med html. Det kanske låter självklart till en början, men grejen är att i java kan jag använda klasser och därmed skapa en kod för flera fabriker i spelet. I javascript å andra sidan var jag tvungen att skriva om den mycket liknande koden för varje fabrik och det finns 6 fabriker i spelet.

2- Att balansera spelet var också en av de stora utmaningarna. Det finns flera faktorer som påverkar spelet. Jag har tänkt att göra värdena i spelet baserade på verkliga siffror. Problemet med det är att det inte finns något konstant. till exempel varierar värdet av att köpa och sälja en

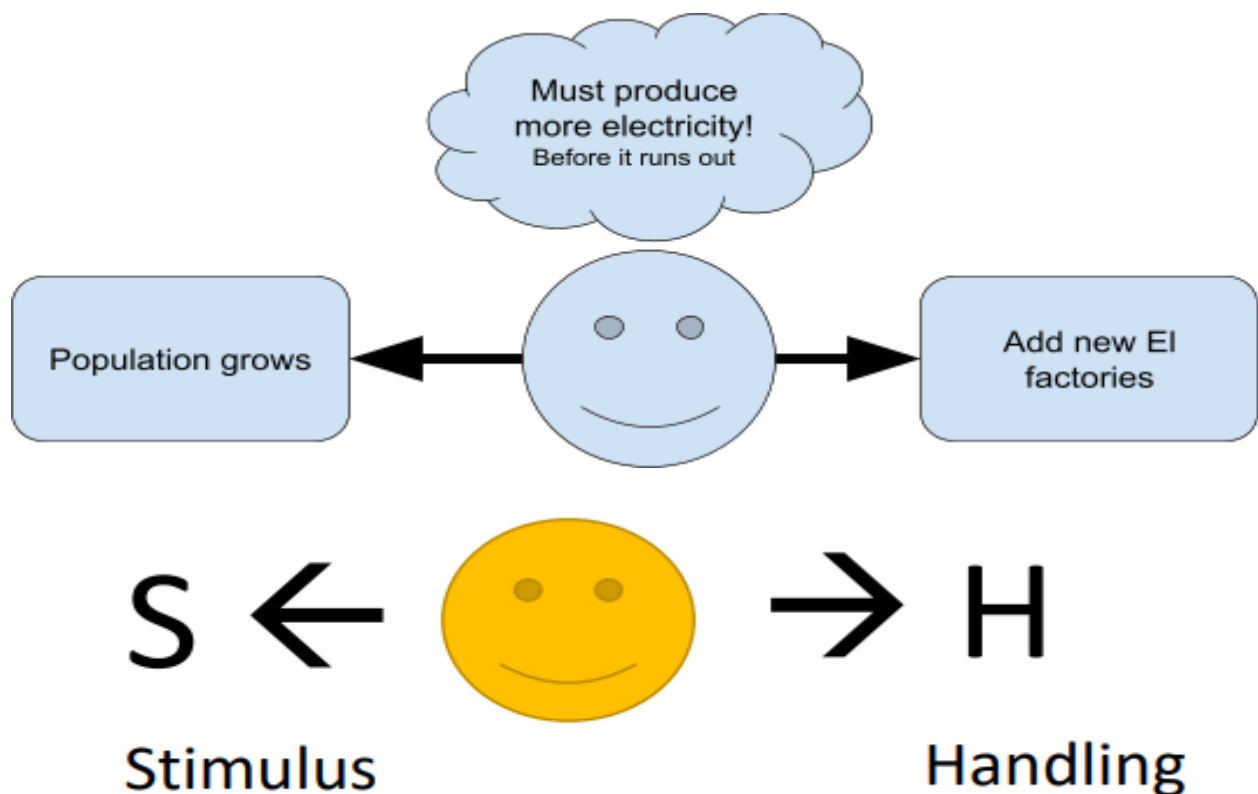
kolfabrik mycket från en plats till en annan. Jag tror att jag har lyckats få spelet i balans till slut(inte baserat på verkliga siffror). Men Det går att hävda att det inte är balanserat.

3-Det fanns några buggar. Jag har fixat de tror jag.

Teori

I föreläsning 4 berättar Thomas Westin om SR och SH-modulen. i SR-modellen accepterar och memorerar den studerande medan i SH-modellen ifrågasätter eleven aktivt. Save The Earth lutar mer åt SH-modellen eftersom spelaren för att vinna måste aktivt fortsätta övervaka status för befolkning, ekonomi/pengar, mängden el och CO2. Det är för att ge rätt handling på de ständigt föränderliga värdena.

Utvecklingspsykologiska inriktningar visar att det som följer SH-modellen i princip också följer konstruktivism som t.ex Piagets intelligensteori. I hög grad verkar det stämma i spelet. Spelet Save The Earth går hand i hand med Piagets intelligensteori om självreglering och assimilering som säger att man anpassar miljön efter sig själv. det är som i spelet att växande befolkning ökar elförbrukningen vilket spelaren som anpassar miljön genom att öka antalet elfabriker då har spelaren möjlighet att vinna(se bilden nedan). det finns fler exempel i spelet men jag ska inte nämna alla.



Mechanics-Dynamics-Aesthetics(MDA)

Mekanik: De särskilda komponenterna i spelet på nivån för datarepresentation och algoritmer. Save The Earth har data representation och algoritmer för spelets faktorerna. Spelaren kan bara interagera med spelet genom att addera eller subtrahera el fabriker. Detta saker kan spelaren vet om genom instruktionerna i huvudmeny, plus genom att testa och experimentera.

Dynamik: Körningsbeteendet hos mekanikerna som verkar på spelaringångar och varandras utgångar övertid. Spelaringångar kan påverka spelets ingående. Spelet Save The Earth gör fokus på begränsningar, val och dess konsekvenser.

Estetik: De önskvärda känslomässiga reaktionerna som framkallats i spelaren när han/hon interagerar med spelsystemet. I spelet utmanas spelaren att kunna konvertera till förnybar el fabriker. För att klara utmaningen måste spelaren tänka kreativt på den bästa strategin att implementera.

Referencer

Föreläsning 1

Föreläsning 4

https://www.w3schools.com/css/tryit.asp?filename=trycss_grid_display_inline-grid

<http://jsfiddle.net/u5jGK/1/>

<https://stackoverflow.com/questions/21585279/need-help-updating-some-text-repeatedly-in-java-script>

https://www.w3schools.com/css/css_grid_container.asp

<https://www.youtube.com/watch?v=q3zc1ph5fvq>

<https://www.chartjs.org/docs/latest/getting-started/>

<https://flaviocopes.com/how-to-disable-button-javascript/>

https://www.w3schools.com/html/tryit.asp?filename=tryhtml_images_girl

https://www.w3schools.com/html/html_images.asp

https://www.w3schools.com/jsref/prop_style_backgroundcolor.asp

https://www.w3schools.com/w3css/tryit.asp?filename=tryw3css_progressbar_round

https://www.w3schools.com/w3css/tryit.asp?filename=tryw3css_progressbar_labels_js

https://www.w3schools.com/w3css/w3css_progressbar.asp

<https://developpaper.com/css-horizontal-progress-bar-and-vertical-progress-bar-implementation-code/>