

```

1 -For the game
2     -No class
3         -Everything in MainSimulator.java
4             -UI (Ben)
5                 -Figure out the math (Aaron)
6                     -Find images (Adam)
7 -Extra features
8     -SQL (Andrew)
9         -Server creation (Ben)
10        -Special events (Aaron and Adam)
11        -Chat window (Andrew)
12        -Games (Stratego)
13        -Sliders (Ben)
14
15 Code architecture
16     -MainSimulator extends PApplet
17         -MainSimulator()
18             -Build frame?
19                 -Adds additional non-processing GUI Elements?
20         -main()
21             -Calls processing main
22             -Launch communication server thread
23             -Launch SQL Thread
24         -settings()
25             -specify window size
26         -setup()
27             -creates window
28             -sets initial values
29             -adds static UI elements (text fields, buttons,
etc)
30             -first face
31         -draw()
32             -updates the face
33             -Show command line SQL
34         -button action methods
35             -update values when buttons are pressed
36             -Send SQL Queries
37     Note: We are not making functions to create buttons,
sliders, etc. Instead we are using the controlP5 processing
38         library to create editable text fields and buttons.
39
40 Data
41
42 final float BASE_HAPPINESS = 50 // Base happiness, run
weekly to populate happyWeekly
43 final float BASE_WEALTH = 0 // Base wealth, run weekly to
populate wealthWeekly
44 final float BASE_GRADE_POTENTIAL = 90 // Base grade
potential, run weekly to populate gradeWeekly
45 final float HOURS_MAX = 160 // Hours maximum, total of all
numbers cannot exceed this
46
47 float wealthWeekly // wealth earned weekly, stored after
week in wealthTotal
48 float wealthTotal // total wealth accrued

```

```

49
50 float happyWeekly // happiness modifier weekly, stored in
51   happySum after week
52 float happySum // stores sum of all happiness modifiers
53 float happyAverage // takes average of happySum and
54   produces a number
55
56 float gradeWeekly // grade potential modifier, stored in
57   gradeSum after week
58 float gradeSum // stores values of weekly grade potentials
59 float gradeAverage // averages gradeSum and produces a
60   number
61
62 int creditHours //
63   Amount of credits taken by user, run once
64   Range: 12 – 24
65   Will decrease happyWeekly by 5/4% per credit over 12
66   Will decrease gradeWeekly by 1.5% over after 12 credits
67   Over twelve credits gives flat $1000 to wealthTotal
68
69 int workHours //
70   Weekly selection of work hours, selected by user
71   Range: 0 – 40
72   Up to 20 hours decreases happyWeekly by .5% per hour
73   Over 20 hours decreases happyWeekly amount by 1% per
74     hour
75   gradeWeekly decreases over 20 hours by .5% per hour
76   Gives $5 to wealthWeekly per hour
77   If player achieves $55 in a week, gives flat happyWeekly
78     bonus of 5%
79
80 int classTime //
81   Weekly selection of active class time, selected by user
82   Range: 0 – credits
83   Decreases happyWeekly by 1% per hour
84   Increases gradeWeekly by .5% per hour
85
86 int studyTime //
87   Weekly selection of amount of time spent studying,
88   selected by user
89   Range: 0 – 2x credits
90   Increases gradeWeekly by .5% per hour
91   Decreases happyWeekly by .5% per hour
92
93 int academicVisit //
94   Weekly selection of use of academic centers, selected by
95   user
96   Range: 0 – .5x studyTime
97   Increases gradeWeekly by .5% per hour
98   Decreases happyWeekly by .5% per hour
99
100 int partyTime //
101   Weekly selection of leisure time, selected by user
102   Range: 0 – (HOURS_MAX – (credits – jobHours – classTime
103     – studyTime – academicVisit))

```

```
95 Increases happyWeekly by 3% per hour
96 Decreases gradeWeekly by 3% per hour
97 Decreases wealthWeekly by $7 per hour, if wealthWeekly
   would be negative, pulls from wealthTotal, if wealthTotal
   is empty, happyWeekly = .5%
98 If no leisure time is taken, gradeWeekly reduced
   severely (50%)
99
100
101
102
103
104
105
106
107
```