



Faculty of Natural Science

Department of Computer Science

Kristianstad University

Course : DA120C

Program : Bachelor in Software Development

Authors : Wael Natafji , Samuel Joseph McMurray , Oussama Safi , Lukas Olsson

Text Adventure Game



Table of Contents

1. Introduction.....	4
1.1 Restrictions.....	4
2. Requirements.....	5
3. Design and Implementation.....	8
3.1 Model Package.....	8
3.1.1 Character Class Model.....	8
3.1.2 Hero Class Model.....	9
3.1.3 Monster Class Model.....	9
3.1.4 Item Class Model.....	10
3.1.5 Key Class Model.....	10
3.1.6 Heart Class Model.....	11
3.1.7 HealthPotion Class Model.....	11
3.1.8 Treasure Class Model.....	12
3.1.9 Coffee Class Model.....	12
3.1.10 Room Class Model.....	13
3.1.11 Door Class Model.....	13
3.1.11 FlavorText Class Model.....	14
3.2 View Package.....	14
3.2.1 HeroView Class.....	14
3.2.2 MonsterView Class.....	15
3.2.3 ItemView Class.....	15
3.2.4 RoomView Class.....	16
3.2.5 DoorView Class.....	16
3.2.6 KeyView Class.....	17
3.3 Controller Package.....	17
3.3.1 HeroController Class.....	17
3.3.2 MonsterController Class.....	18
3.3.3 RoomController Class.....	18
3.3.4 ItemController Class.....	19



3.3.5 DoorController Class.....	19
3.3.6 KeyController Class.....	20
3.3.6 FlavorTextController Class.....	20
4. Test Results.....	21
5. Summary and Conclusion.....	22
5.1 Weekly Progress.....	22
5.1.1 Week 1.....	22
5.1.2 Week 2.....	22
5.1.3 Week 3.....	22
5.1.4 Week 4.....	23
5.1.5 Week 5.....	23
5.1.6 Week 6.....	23
5.1.7 Week 7.....	23
5.2 Difficulties and challenges.....	23
5.2.1 Getting Started with GIT Difficulty 1.....	23
5.2.2 Hero Navigation Between Rooms Difficulty 2.....	23
5.2.3 Learning MVC architecture 3.....	23
5.3 Correctness of time estimates.....	23
5.4 Priority decisions.....	24
5.5 Conclusion.....	24
6 References.....	25



1. Introduction

We were asked to design and implement an application as a project, so we chose to create a text adventure game which is written in java programming language. The user is tasked with escaping from a labyrinth by finding keys and collecting items. While the goal is to escape, points are earned by collecting “treasures” throughout the game that will increase their score.. The user has two choices either fight or flee both options can have deadly consequences! If the user chooses to fight the monster, they may face a grizzly end. Escape is possible if your fast enough, but you won’t discover as much treasure. The user can find items to help along their journey, the items can rejuvenate health and add additional lives. One of the items being keys to open the locked doors and another being hidden treasure which increases the score. The user will be able to save and load game, the user will also be able to save the high score.

1.1 Restrictions

The game has some rules that the user must follow. In order to successfully get to the end the player must not let their health or lives be depleted.

The user is not always going to have the advantage when playing the game, as every new player has to choose between three characters each of which have different statistics that can affect gameplay dramatically. The items will be dropped randomly based on a percentage chance and will be based on rarity with items like treasure being the least rare and usable items being more rare.

Our program also uses no Java FX or any graphics for its interface. While this is to expect as the program is a simple word game, it might not satisfy most modern users.



2. Requirements

Table 1 below lists all the requirements for this project. The time estimated (EMH) and time spent on a task (AMH) is measured in man hours. The prio column lists the priorities where Prio 1 is highest priority and Prio 10 is lowest priority.

Table 1 - Requirements

Req. No	Req. Name	Req. Description	EM H	AMH	Prio
1	MVC Architecture	Build MVC Architecture within java project	2	3 DONE	1
2	Start Menu	Create start menu with for beginning the game and choosing character	2	1.5 DONE	2
3	Normal game menu	Create game menu for use while within the labyrinth (not encounters)	3	4 DONE	3
4	Implement character class	Add character variables and attributes to character class	1	1.5 DONE	4
5	Implement hero subclass	Add hero variables and attributes to hero subclass	1.5	2 DONE	5
6	Implement monster subclass	Add monster variables and attributes to monster subclass	0.5	0.5 DONE	6



7	Design Map Outside Java	Create an outline for the map using pen and paper. Place doors and objects within etc.	0.5	1 DONE	7
8	Fill Room Class	Fill room class with appropriate variables	0.5	0.5 DONE	8
9	Implement Designed Map	Within java framework implement the map in the	2	3+ ongoing	9
10	Create initial story elements	Create a story element including goal and necessary info	0.5	1 DONE	10
11	Add Monsters	Create the various possible monsters to encounter	2		11
12	Map navigation	Navigate the map by taking the commands in the normal game menu	2		12
13	Encounter Menu	Add encounter menu with inputs and reactions from players	3		13
14	Printable map screen	Map screen that shows location and where you've been	2		14
15	Death Menu	To give options for what to do upon	1.5		15
16	Treasure Score	Track treasure pickup and list score at end of game	2		16



17	High Score	Ability to check highest score	4		17
18	Additional Flavor Text	Add more flavor text for entering rooms, attacks and finish other placeholders	4		18
19	Save game function	Add ability to save and quit	3		19
20	Implement Item Functions	Add functionality to items	4		20
21	Key and Doors	Implement doors and keys to function within game	4		21
22	Level System	Use exp value from monsters to increase level providing health benefits etc.	2		22
23	Boss Fight	Add final boss for end game finish	2		23
24	End Screen Menu Win	Add a screen for when you win	1		24
25	Difficulty System	Modify existing enemies and player health etc to create easier/more difficult fights	4		25



3. Design and Implementation

3.1 Model Package

The model package holds the data of the objects, the objects are instantiated in the main and then connected to the view and then to the controller.

3.1.1 Character Class Model

The character class is the super class of the hero class and the monster class.

A screenshot of the IntelliJ IDEA IDE showing the source code for the Character.java file. The file is located in the package project1E7.Model. The code defines a public class Character with protected attributes health, strength, speed, description, and name. It includes a constructor and several getter and setter methods. The IDE interface shows the project structure on the left, the code editor in the center, and the status bar at the bottom.

```
1 package project1E7.Model;
2
3 public class Character {
4     protected int health;
5     protected int strength;
6     protected int speed;
7     protected String description;
8     protected String name;
9
10    public Character(int health, int strength, int speed, String description, String name) {
11        this.health = health;
12        this.strength = strength;
13        this.speed = speed;
14        this.description = description;
15        this.name = name;
16    }
17
18    public int getHealth() { return health; }
19
20    public void setHealth(int health) { this.health = health; }
21
22    public int getStrength() { return strength; }
23
24    public void setStrength(int strength) { this.strength = strength; }
25
26    public int getSpeed() { return speed; }
27
28    public void setSpeed(int speed) { this.speed = speed; }
29
30 }
```




3.1.2 Hero Class Model

The hero class inherits from the character class has association with main class with aggregation from both item class and key class.

A screenshot of the IntelliJ IDEA IDE showing the Hero.java file. The code defines a Hero class that extends Character. It includes private attributes for endurance, keyRing, backPack, lives, currentTreasure, and alive. The constructor initializes these attributes. There are getter and setter methods for keyRing, backPack, endurance, lives, and alive.

```
1 public class Hero extends Character {
2     private int endurance;
3     private Key[] keyRing;
4     private ArrayList<Item> backPack;
5     private int lives;
6     private int currentTreasure;
7     private boolean alive;
8     public Hero(int health, int strength, int speed, String description, String name, int endurance) {
9         super(health, strength, speed, description, name);
10        this.endurance = endurance;
11        lives = 5;
12        currentTreasure = 0;
13        alive = true;
14    }
15
16    public Key[] getKeyRing() { return keyRing; }
17
18    public void setKeyRing(Key[] keyRing) { this.keyRing = keyRing; }
19
20    public ArrayList<Item> getBackPack() { return backPack; }
21
22    public void setBackPack(ArrayList<Item> backPack) { this.backPack = backPack; }
23
24    public int getEndurance() { return endurance; }
25
26    public int getLives() { return lives; }
27
28    public void setLives(int lives) { this.lives = lives; }
29 }
```

3.1.3 Monster Class Model

The monster class inherits from the character class with aggregation from item class and association with room class.



```
1 package project1E7.Model;
2
3 public class Monster extends Character {
4     private Item item;
5     private boolean alive;
6     private int xp;
7
8     public Monster(int health, int strength, int speed, String description, String name, Item item, boolean alive, int xp) {
9         super(health, strength, speed, description, name);
10        this.item = item;
11        this.alive = alive;
12        this.xp = xp;
13    }
14
15    public Item getItem() { return item; }
16
17    public void setItem(Item item) { this.item = item; }
18
19    public boolean isAlive() { return alive; }
20
21    public void setAlive(boolean alive) { this.alive = alive; }
22
23    public int getXp() { return xp; }
24 }
25
```

3.1.4 Item Class Model

The item class is the super class for key class, treasure class, heart class, healthpotion class, and coffee class. It has an association with room class aggregation with both monsters and players.

```
1 package project1E7.Model;
2
3 public class Item {
4     private String name;
5
6     public Item(String name) { this.name = name; }
7
8     public String getName() {
9         return name;
10    }
11 }
12
```

3.1.5 Key Class Model

The key class inherits from the item class and has an aggregation with the hero class.



```
1 package project1E7.Model;
2
3 public class Key extends Item {
4     public Key(String name) { super(name); }
5 }
6
7
8
```

3.1.6 Heart Class Model

The heart class inherits from the item class.

```
1 package project1E7.Model;
2
3 public class Heart extends Item {
4     private int extraLife;
5
6     public Heart(String name) {
7         super(name);
8         extraLife = 1;
9     }
10
11     public int getExtraLife() { return extraLife; }
12
13 }
14
15
16
```



3.1.7 HealthPotion Class Model

The healthpotion class inherits from the item class.

A screenshot of an IDE window showing the code for HealthPotion.java. The code is as follows:

```
1 package project1E7.Model;
2
3 public class HealthPotion extends Item {
4     private Integer addHealth;
5
6     public HealthPotion(String name) {
7         super(name);
8         addHealth = 60;
9     }
10
11     public Integer getAddHealth() { return addHealth; }
12
13 }
14
15
```

The IDE interface includes a menu bar at the top, a toolbar with icons for running, debugging, and other actions, and a sidebar on the left with tabs for Project, Z-Structure, and Favorites. The bottom status bar shows the time as 3:14, CRLF line endings, UTF-8 encoding, 4 spaces for indentation, and the Git branch as development. The system tray at the bottom right shows the date and time as 8:25 PM on 12/4/2019.

3.1.8 Treasure Class Model

The treasure class inherits from the item class.



```
1 package project1E7.Model;
2
3 public class Treasure extends Item {
4     private int amount;
5
6     public Treasure(String name,int amount) {
7         super(name);
8         this.amount = amount;
9     }
10
11     public int getAmount() { return amount; }
12
13     public void setAmount(int amount) { this.amount = amount; }
14 }
15
16
17
18
19
```

3.1.9 Coffee Class Model

The coffee class inherits from the item class.

```
1 package project1E7.Model;
2
3 public class Coffee extends Item {
4     private Integer speed;
5
6     public Coffee(String name) {
7         super(name);
8         speed = 3;
9     }
10
11     public Integer getSpeed() { return speed; }
12
13 }
14
15
```



3.1.10 Room Class Model

The room class has an association with both item class and monster class, it also has an aggregation with door class and composition with main class.

```
1 public class Room {
2     private String description;
3     private Item item;
4     private boolean hasMonster;
5     private boolean found;
6     private Monster monster;
7     private boolean hasItem = false;
8
9     public Room(String description, Item item, boolean hasMonster, boolean found, Monster monster, boolean hasItem) {
10         this.description = description;
11         this.item = item;
12         this.hasMonster = hasMonster;
13         this.found = found;
14         this.monster = monster;
15         this.hasItem = hasItem;
16     }
17
18     public String getDescription() { return description; }
19
20     public Item getItem() { return item; }
21
22     public void setDescription(String description) { this.description = description; }
23
24     public void setItem(Item item) { this.item = item; }
25
26     public boolean isMonster() { return hasMonster; }
27
28     public void setMonster(boolean monster) { this.hasMonster = monster; }
29 }
```

3.1.11 Door Class Model

The door class has an aggregation with the room class.

```
1 package project1E7.Model;
2
3 public class Door {
4     private boolean lock;
5     private String nameOfKey;
6     private String description;
7
8     public Door(boolean lock, String nameOfKey, String description) {
9         this.lock = lock;
10        this.nameOfKey = nameOfKey;
11        this.description = description;
12    }
13
14    public boolean isUnlocked() { return lock; }
15
16    public String getNameOfKey() { return nameOfKey; }
17
18    public String getDescription() { return description; }
19 }
```



3.1.11 FlavorText Class Model

The FlavorText is a temporary class anything done in here will be moved once we figure out where everything will go.

A screenshot of an IDE window showing the FlavorText.java file. The code is as follows:

```
1 package project1E7.Model;
2
3 public class FlavorText {
4
5     private String flavorText;
6
7     public String getFlavorText() { return flavorText; }
8
9     public void setFlavorText(String flavorText) { this.flavorText = flavorText; }
10
11 }
12
13
14
```

The IDE interface includes a menu bar at the top, a toolbar with icons for running, debugging, and other actions, and a sidebar on the left showing the project structure. The bottom status bar indicates the current file is FlavorText, and the bottom right corner shows the time as 3:14 and the date as 12/4/2019.

3.2 View Package

The view package is what will eventually hold everything that will be printed out to the screen for the user to see sent from the location related to what will be printed takes the model as a parameter.



3.2.1 HeroView Class

```
theTextAdventure [C:\Users\De\IdeaProjects\theTextAdventure] - ...src\project1E7\View\HeroView.java
src project1E7 View HeroView
HeroView.java
13 public class HeroView {
14     Hero model;
15     Scanner input = new Scanner(System.in);
16
17     public HeroView(Hero model) { this.model = model; }
18
19     public void printStats() {
20         int[] stats = new int[4];
21         stats[0] = model.getHealth() / 10;
22         stats[1] = model.getSpeed() / 10;
23         stats[2] = model.getStrength() / 10;
24         stats[3] = model.getEndurance() / 10;
25
26         String[] tableDescrip = new String[4];
27         tableDescrip[0] = "Health: ";
28         tableDescrip[1] = "Speed: ";
29         tableDescrip[2] = "Strength: ";
30         tableDescrip[3] = "Endurance: ";
31
32         System.out.println("Name: " + model.getName());
33         System.out.println("Character Description: " + model.getDescription());
34
35         for (int i = 0; i < stats.length; i++) {
36             System.out.print(tableDescrip[i]);
37             for (int j = 0; j < stats[i]; j++) {
38                 System.out.print(" ");
39             }
40             System.out.println();
41         }
42     }
43 }
```

3.2.2 MonsterView Class

```
theTextAdventure [C:\Users\De\IdeaProjects\theTextAdventure] - ...src\project1E7\View\MonsterView.java
src project1E7 View MonsterView
MonsterView.java
1 package project1E7.View;
2
3 import project1E7.Model.Monster;
4
5 public class MonsterView {
6     Monster model;
7
8     public MonsterView(Monster model) { this.model = model; }
9
10    public void printStatus(Monster monster) {
11        System.out.println("Enemy Type: " + model.getDescription());
12        if (model.isAlive() == true) {
13            System.out.println("Health: " + model.getHealth());
14        }
15        else {
16            System.out.println("The " + monster + "has been defeated at your hands...");
17        }
18    }
19
20    public void encounterMenu() {
21        System.out.printf("\nYou have encountered a %s\n", model.getDescription() +
22            "\nWhat is your next move?\n" +
23            "To fight enter '1'\n" +
24            "To flee enter '2'\n" +
25            "To use one of your item enter '3'\n");
26    }
27 }
28
29 }
```




3.2.3 ItemView Class

A screenshot of the IntelliJ IDEA IDE showing the source code for the ItemView class. The interface includes a menu bar at the top, a toolbar, and a sidebar on the left with tabs for Project, Structure, and Favorites. The main editor area displays the Java code for ItemView.java. The code defines a public class ItemView with private attributes for Coffee, HealthPotion, Heart, and Treasure. It includes a constructor and a viewItem method that uses a switch statement to print flavor text for each item type. The bottom status bar shows the current file is ItemView, and the system tray at the very bottom displays the time and date.

```
5 public class ItemView {
6     Item model;
7     private Coffee coffee;
8     private HealthPotion healthPotion;
9     private Heart heart;
10    private Treasure treasure;
11
12    public ItemView(Item model) { this.model = model; }
13
14
15
16    public Item viewItem(Item item) {
17        if (item == coffee) {
18            System.out.println("flavor text for coffee");
19            return coffee;
20        } else if (item == healthPotion) {
21            System.out.println("flavor text health potion");
22            return healthPotion;
23        } else if (item == heart) {
24            System.out.println("flavor text for heart");
25            return heart;
26        } else if (item == treasure) {
27            System.out.println("flavor text for treasure");
28            return treasure;
29        }
30        return null;
31    }
32 }
33
```



3.2.4 RoomView Class

```
1 package project1E7.View;
2
3
4 import ...
5
6
7
8 public class RoomView {
9     Room model;
10
11     public RoomView(Room model) { this.model = model; }
12
13
14
15     public boolean roomHasItem(Room room) {
16         if (model.isHasItem() == true) {
17             return true;
18         } else {
19             return false;
20         }
21     }
22 }
```

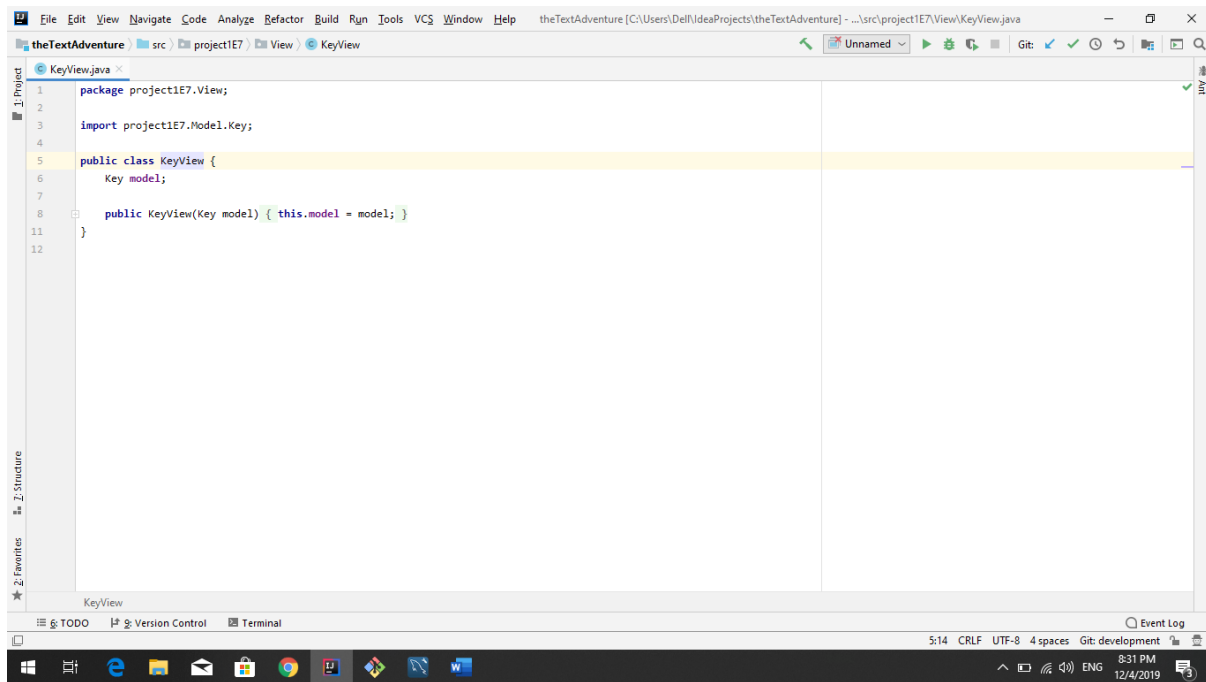
3.2.5 DoorView Class

```
1 package project1E7.View;
2
3 import project1E7.Model.Door;
4
5 public class DoorView {
6     Door model;
7
8     public DoorView(Door model) { this.model = model; }
9
10
11
12 }
```



3.2.6 KeyView Class

This class may be leaving.

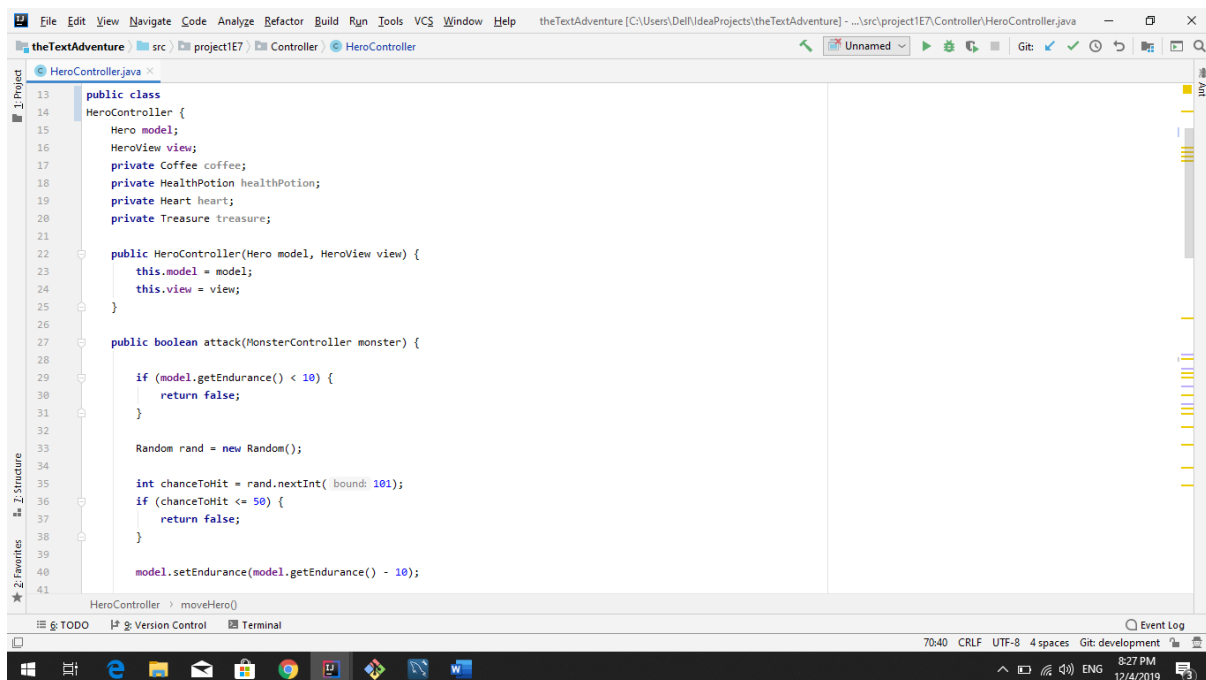


```
1 package project1E7.View;
2
3 import project1E7.Model.Key;
4
5 public class KeyView {
6     Key model;
7
8     public KeyView(Key model) { this.model = model; }
9
10 }
11
12
```

3.3 Controller Package

The controller package holds the logic operations of the game to the respective classes the controller takes the model and the view as a parameter when created.

3.3.1 HeroController Class



```
13 public class
14 HeroController {
15     Hero model;
16     HeroView view;
17     private Coffee coffee;
18     private HealthPotion healthPotion;
19     private Heart heart;
20     private Treasure treasure;
21
22     public HeroController(Hero model, HeroView view) {
23         this.model = model;
24         this.view = view;
25     }
26
27     public boolean attack(MonsterController monster) {
28
29         if (model.getEndurance() < 10) {
30             return false;
31         }
32
33         Random rand = new Random();
34
35         int chanceToHit = rand.nextInt( bound: 101);
36         if (chanceToHit <= 50) {
37             return false;
38         }
39
40         model.setEndurance(model.getEndurance() - 10);
41
42     }
43
44     moveHero()
45 }
```



3.3.2 MonsterController Class

```
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help theTextAdventure [C:\Users\Deif\IdeaProjects\theTextAdventure] - ...project1E7\Controllor\MonsterController.java
theTextAdventure src project1E7 Controllor MonsterController
MonsterController.java
1 package project1E7.Controllor;
2
3 import ...
4
5
6
7
8 public class MonsterController {
9     Monster model;
10    MonsterView view;
11
12    public MonsterController(Monster model, MonsterView view) {
13        this.model = model;
14        this.view = view;
15    }
16
17    public void takeDamage(int incDamage) {
18
19        if (incDamage > model.getHealth()) {
20            model.setAlive(false);
21        } else
22            model.setHealth(model.getHealth() - incDamage);
23    }
24
25    public boolean attack(HeroController hero) {
26
27        int damageDone;
28        Random rand = new Random();
29
30        int chanceToHit = rand.nextInt( bound: 100);
31
32    }
33
34 }
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635

```



3.3.4 ItemController Class

```
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help theTextAdventure [C:\Users\Deff\IdeaProjects\theTextAdventure] - ...src\project1E7\Controller\ItemController.java
theTextAdventure src project1E7 Controller ItemController
ItemController.java
1 package project1E7.Controller;
2
3 import ...
4
5
6
7
8 public class ItemController {
9     Item model;
10    ItemView view;
11    private Treasure treasure;
12    private Coffee coffee;
13    private HealthPotion healthPotion;
14    private Heart heart;
15
16    public ItemController(Item model, ItemView view) {
17        this.model = model;
18        this.view = view;
19    }
20
21 }
```

3.3.5 DoorController Class

```
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help theTextAdventure [C:\Users\Deff\IdeaProjects\theTextAdventure] - ...src\project1E7\Controller\DoorController.java
theTextAdventure src project1E7 Controller DoorController
DoorController.java
1 package project1E7.Controller;
2
3 import ...
4
5
6
7
8 public class DoorController {
9     Door model;
10    DoorView view;
11
12    public DoorController(Door model, DoorView view) {
13        this.model = model;
14        this.view = view;
15    }
16
17 }
```



3.3.6 KeyController Class

May be leaving this class behind.

A screenshot of an IDE window showing the code for KeyController.java. The code is as follows:

```
1 package project1E7.Controller;  
2  
3 import ...  
4  
5  
6 public class KeyController {  
7     Key model;  
8     KeyView view;  
9  
10    public KeyController(Key model, KeyView view) {  
11        this.model = model;  
12        this.view = view;  
13    }  
14 }  
15
```

The IDE interface includes a menu bar at the top, a toolbar, and a sidebar on the left with 'Project' and 'Favorites' views. The bottom status bar shows the time as 6:14, CRLF line endings, UTF-8 encoding, 4 spaces for indentation, and the Git branch 'development'.

3.3.6 FlavorTextController Class

This is a temporary class and will be leaving to the respective view classes.

A screenshot of an IDE window showing the code for FlavorTextController.java. The code defines a public class with various private String attributes for intro, help, weapons, room flavors, encounter flavors, and attack flavors. The IDE interface includes a menu bar, a toolbar, a project explorer on the left, and a status bar at the bottom showing the current file, encoding, and date/time.

```
5 public class FlavorTextController {
6
7     private String introFlavorText;
8
9     private String helpText;
10
11     private String weaponWarrior;
12     private String weaponThief;
13
14     private String roomFlavorText1;
15     private String roomFlavorText2;
16     private String roomFlavorText3;
17     private String roomFlavorText4;
18     private String roomFlavorText5;
19
20     private String encounterFlavorTextSpiderling1;
21     private String encounterFlavorTextSpiderling2;
22     private String encounterFlavorTextBat1;
23     private String encounterFlavorTextBat2;
24     private String encounterFlavorTextSlime1;
25     private String encounterFlavorTextSlime2;
26     private String encounterFlavorTextSkeleton1;
27     private String encounterFlavorTextSkeleton2;
28     private String encounterFlavorTextOwlbear1;
29     private String encounterFlavorTextOwlbear2;
30     private String encounterFlavorTextBoss;
31
32     private String attackFlavorText1;
33     private String attackFlavorText2;
34 }
```



4. Test Results

Table 2 below contains the current status of implemented and tested requirements.

<Instructions: This table shall map 1-1 to the table in Chapter 2. The test result for each requirement shall be one of the following: NOT IMPLEMENTED, PASSED or FAILED.>

Table 2 - Test Results

Req. No	Req. Name	Test Result
1	MVC Architecture	PASSED
2	Start Menu	PASSED
3	Normal game menu	NOT IMPLEMENTED
4	Implement character class	PASSED
5	Implement hero subclass	PASSED
6	Create initial story elements	PASSED
7	Design Map Outside Java	PASSED
8	Fill Room Class	NOT IMPLEMENTED
9	Implement Designed Map	FAILED
10	Create initial story elements	PASSED
11	Add Monsters	NOT IMPLEMENTED
12	Map navigation	NOT IMPLEMENTED
13	Encounter Menu	NOT IMPLEMENTED
14	Printable map screen	NOT IMPLEMENTED
15	Death Menu	NOT IMPLEMENTED
16	Treasure Score	NOT IMPLEMENTED
17	High Score	NOT IMPLEMENTED
18	Additional Flavor Text	NOT IMPLEMENTED
19	Save game function	NOT IMPLEMENTED
20	Implement Item Functions	NOT IMPLEMENTED
21	Key and Doors	NOT IMPLEMENTED
22	Level System	NOT IMPLEMENTED
23	Boss Fight	NOT IMPLEMENTED
24	End Screen Menu Win	NOT IMPLEMENTED



25	Difficulty System	NOT IMPLEMENTED
----	-------------------	-----------------



5. Summary and Conclusion

The text adventure game is coming along it is our belief we have finished all the classes we need for the game to operate as far as we know now. The start menu has been implemented currently you can only start a game and exit as we have not finished the view high score and load game functions. The choose hero menu has been implemented with a star chart of the heroes stats, you can also choose your hero. The world has been created using a object multidimensional array for Room we have filled the rooms with items and monsters where we desire them, we also have been developing the functionality for random item and monster drops in rooms. That feature is not yet tested we are still trying to figure out how to move around the map.

5.1 Weekly Progress

Below is a short summary of what was done each week.

5.1.1 Week 1

During this week we spent several hours planning the structure of our text based game. This included deciding which features were to be implemented and which were to be cut. This included deciding how to structure the classes and methods. There is still work to do on some aspects of structure and decision such as how we will implement the classes within the game and what abilities each will have as well as how they will be used through methods, classes, or some specific condition. This is shown on our requirements list as the least prioritized as it was decided to be worked on later when we choose the different abilities for each character. Our biggest challenge was talking through exactly which features of the game we were comfortable in completing in the allotted time and which ones we decided not to. Luckily being the optimistic group we are most features are staying in the plan right now. We learned it is a time consuming process to plan a game, even as simple as this one. After this week we are feeling much more confident and on the same page now.

5.1.2 Week 2

Most of time has been spent on figuring out git as it was a new experience for a majority of the group and issues popped up frequently. These issues have continuously have costed hours of valuable programming time our solution will be creating a branch for each of us after we merge to the master and merging through out the week when we meet up to discuss progress and talk game plan. As a group we decided we wanted to take the next step in learning about development architecture so we have converted our text adventure game to a MVC architecture. We have had some trial and error figuring out exactly how we go about implementing but I believe we are starting to get the hang of it. We have implemented the Start game menu which only is capable of quitting and starting a game. We have also implemented the choosing of the hero menu where we also print out a bar chart made of stars as a visual representation of the heroes skills. Neither of these features are yet to be break proof yet, we need to learn more about try catch or other features that can combat the wrong variable type. We have also implemented the starting story description but the place at which it is located is subject to change. Due to set backs with git we were not able to get the map out in a timely manner and have just gotten within the past 2 days and needs some changes as to how it builds the map. We have started on developing the fighting logic but have yet to be able to test as we are still learning the MVC figuring out how to get a specific monster from a list of monsters or even have the room return the monster has proven difficult. Although we are confident these things can be worked out and we believe we are headed in the right direction.

5.1.3 Week 3

TODO.



5.1.4 Week 4

TODO.

5.1.5 Week 5

TODO.

5.1.6 Week 6

TODO.

5.1.7 Week 7

TODO.

5.2 Difficulties and challenges

Below is a list of notable challenges that came up during this project and that took a long time to solve.

5.2.1 Getting Started with GIT Difficulty 1

Due to the inexperience with git we have run into numerous issues and difficulties costing us hours of potential coding time. Our solution is to make separate branches for each member of the group and merge when we meet during the week.

5.2.2 Hero Navigation Between Rooms Difficulty 2

We had thought we solved it and moved on but turns out not to be the case. We have the basic idea as to how it must be done but solving has given us the most trouble in how we can give it the necessary functionality, without having to write long code. We decided since it is close to deadline to prioritize that above all for the next week and moved on fine tuning our current code.

5.2.3 Learning MVC architecture 3

Although it's not required we want to implement this architecture design to our game. It has probably set us back a bit but we believe we are finally getting the hang of it and are sure we can make it work for the better of our project. It is also a very important lesson to be learned.

5.3 Correctness of time estimates

We did not have the correct requirements during our first meeting and also didn't have the correct times on the last report so although I'd like to say we were accurate I can't say for sure but I believe we would probably be ahead of schedule had we not had issues with our git setup. So we have a solution in mind for after we merge to the master.



5.4 Priority decisions

Our priorities may have been a little out of order but our primary problem was trying to do too much quickly we should have developed slowly making improvements 1 step at a time instead of trying to do everything at once. We should have attempted to get a minimum viable product out rather than what we have, but there is hope for us yet we can definitely improve on what we have and produce a high quality text adventure game.

5.5 Conclusion

<Instructions: Look back on the whole project. Here you can write a bit more freely about your thoughts on this project. What was your overall experience? How was the teamwork? What did you learn? Can you list some points that you will do better in next project? Other thoughts. >



6 References