Magic Cards

An educational game about Error Detection

Team: Magicians

Team Members

- Chen Shi (010119005)
- Jayam Malviya (011435567)
- Miao Shi (010823527)
- Prateek Sharma (011475620)

What is Magic Cards?

Who can play?

- A fun two-player competitive online game
- For kids 8+ years old

Why is it an educational game?

- It is about an important computer science concept – Error Detection

What is Error Detection?

- A computer science concept
- Data transmission is interfered by noises
- Adding redundancy data to find errors
- Examples: parity bit, checksum
- Usages: data storage, satellite broadcasting

Game instruction and rules

- You will see a grid of cards
- Try to remember the pattern of the cards within 5 seconds.
- One of the cards will be flipped over secretly
- You need to find out which card is flipped based on your memory
- If you and your opponent both pick wrong cards, both of you lose
- If you pick the right card, and your opponent pick a wrong card, you win
- If you and your opponent both pick the right cards, whoever take less time is the winner
- You may learn how to use Error Detection concept in the game

Demo time!!!

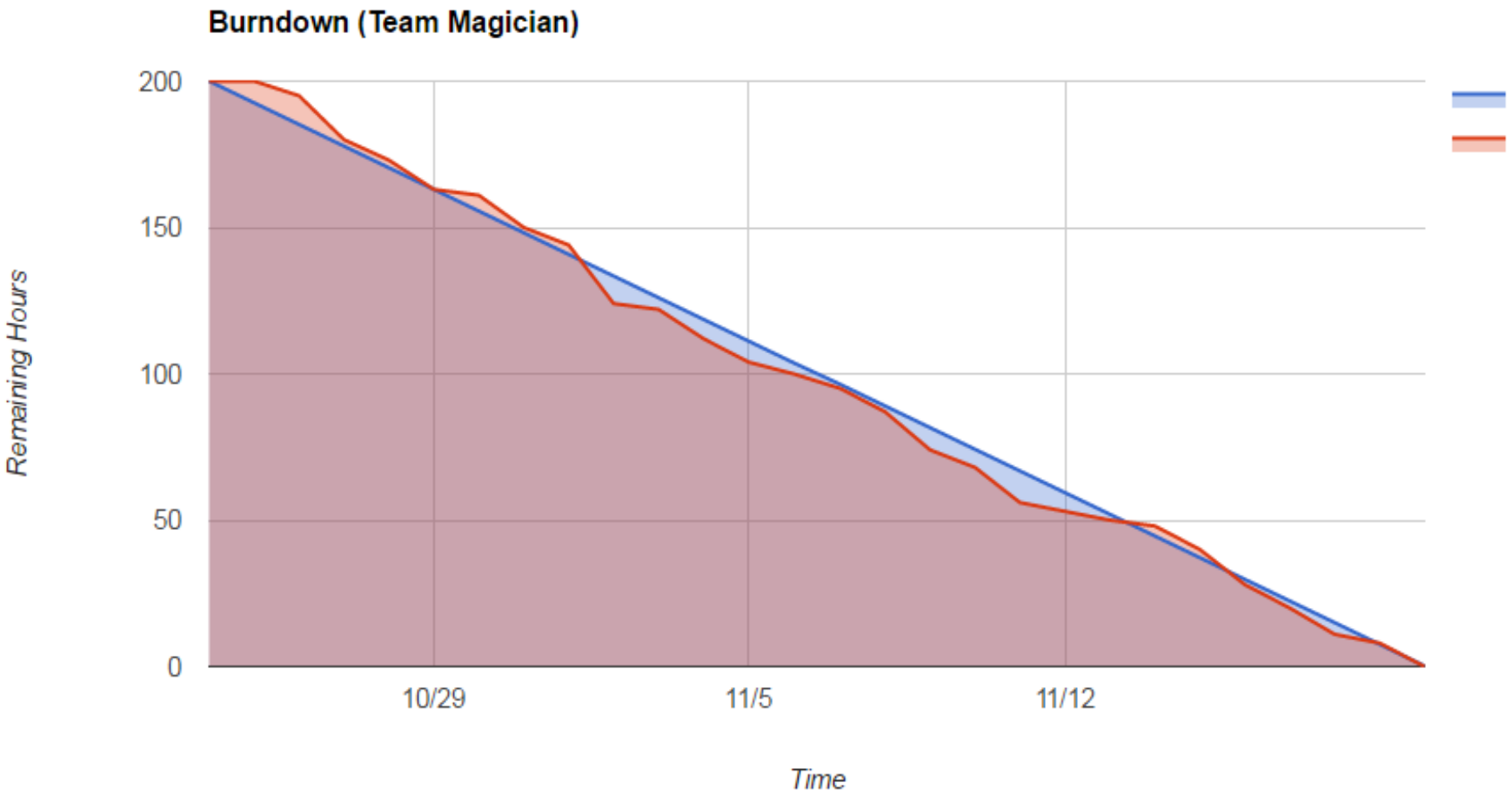
How was this game developed?

- UML Diagrams
- Design patterns
- Server as Restlet application
- Docker Cloud service

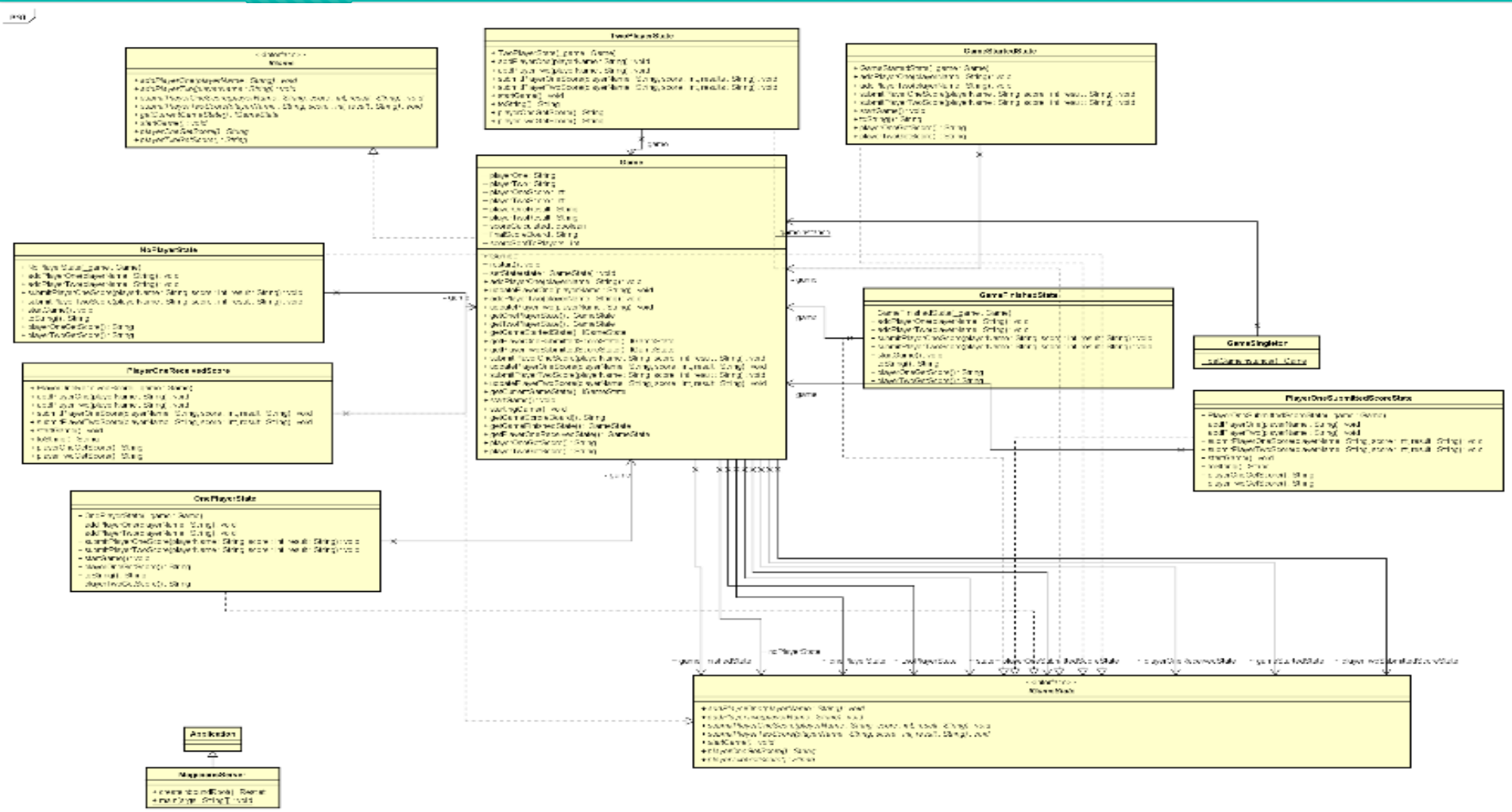
Sprint Sheet

				Week #1 (10 hrs / week)							Week #2 (10 hrs / week)							Week #3 (10 hrs / week)							Week #4 (10 hrs / week)						
Backlog Item	Task	Task Owner	Initial Estimate (Total Sprint Hours = 40 x 5)	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15	D16	D17	D18	D19	D20	D21	D22	D23	D24	D25	D26	D27	D28
				10/24	10/25	10/26	10/27	10/28	10/29	10/30	10/31	11/1	11/2	11/3	11/4	11/5	11/6	11/7	11/8	11/9	11/10	11/11	11/12	11/13	11/14	11/15	11/16	11/17	11/18	11/19	11/20
			200	200	193	185	178	170	163	156	148	141	133	126	119	111	104	96	89	82	74	67	59	52	45	37	30	22	15	8	0
			200	200	200	195	180	173	163	161	150	144	124	122	112	104	100	95	87	74	68	56	53	50	48	40	28	20	11	8	0
Create Menu World	Create button to navigate world	Miao	3	3	3	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Integerate Grid class	Jayam	3	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Integrate Card class	Jayam	4	3	2	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Integrater Maigician class	Prateek	6	6	4	4	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Design User Interface	Miao	2	2	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Complete Test	Chen	2	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Complete Test	Miao	2	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Create Theory World	Desgin User Interface	Chen	3	3	3	3	3	2	2	2	4	4	4	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Implement Navigation Button within Theory World	Chen	5	5	5	5	5	5	5	5	4	4	4	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Implement Back Button to Navigate to Previous World	Miao	3	3	3	3	3	3	3	3	3	3	3	2	2	2	2	2	2	1	1	1	1	1	1	1	0	0	0	0	0
	Complete Test	Carlos	2	6	6	6	6	6	6	6	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	1	0	0	0	0
Create Demo World	Desgin User Interface	Jayam	2	2	2	2	2	2	2	2	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Implement Navigation Button within Demo World	Chen	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	3	2	0	0	0	0	0	0	0	0	0	0	0
	Implement Back Button to Navigate to Previous World	Miao	3	3	3	3	3	3	3	3	3	3	3	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Complete Test	Prateek	5	5	5	5	5	5	5	5	5	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Create multip-player functionality	Create Team AWS & Docker Account	Prateek	4	4	4	4	4	4	4	4	4	4	3	3	3	2	2	2	2	2	1	0	0	0	0	0	0	0	0	0	0
	Develop Server code and integrate with Restlet	Prateek	6	6	6	6	6	6	6	6	6	6	5	5	5	4	4	4	4	4	4	4	3	1	1	1	1	1	0	0	0
	Modify applicatio to communicate with Server	Jayam	5	4	4	4	4	4	4	4	4	4	3	3	3	3	3	3	3	3	0	0	0	0	0	0	0	0	0	0	0
	Develop new logic to handle multiple players	Carlos	6	6	6	6	6	6	6	6	5	5	4	4	4	4	4	4	4	4	4	4	4	4	4	4	3	2	1	0	0
	Implement Design Pattern 1	Carlos	5	6	6	6	6	6	6	4	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	2	0
	Implement Design Pattern 2	Miao	5	4	4	4	4	4	4	4	4	4	4	4	4	3	3	3	3	3	3	1	1	1	1	1	0	0	0	0	0
	Implement Design Pattern 3	Prateek	5	5	5	5	5	5	5	5	5	5	5	5	5	5	4	4	4	3	2	2	2	2	2	2	2	2	2	1	0
	Implement Design Pattern 4	Jayam	5	3	4	4	4	4	4	4	4	4	4	4	4	4	4	3	3	1	1	1	1	1	1	1	1	1	1	1	0
	Implement Design Pattern 5	Chen	5	5	5	5	5	5	5	5	4	4	4	4	4	3	3	2	2	2	1	1	1	1	1	0	0	0	0	0	0
Deployment and Documentation	Set up Docker individually	Miao	6	6	6	6	6	6	6	6	6	6	5	5	4	3	3	2	2	2	2	1	1	1	1	1	1	1	0	0	0
	Set up Docker individually	Carlos	6	6	6	6	6	6	3	3	2	2	2	2	2	2	2	2	2	2	2	0	0	0	0	0	0	0	0	0	0
	Set up Docker individually	Jayam	6	6	6	6	6	6	6	6	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Set up Docker individually	Chen	6	6	6	6	3	3	3	3	3	3	3	3	3	3	3	2	2	2	2	2	2	2	2	2	2	2	0	0	0
	Set up Docker individually	Prateek	6	6	6	6	6	6	6	6	6	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Set up AWS VM Individually	Miao	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	5	5	4	4	4	2	0	0	0	0	0	0	0	0
	Set up AWS VM Individually	Carlos	6	6	6	6	6	6	3	3	2	2	2	2	2	2	2	2	2	2	2	0	0	0	0	0	0	0	0	0	0
	Set up AWS VM Individually	Prateek	6	6	6	6	6	6	6	6	6	6	6	6	6	5	4	4	4	3	3	3	3	3	3	3	3	3	2	2	0
	Set up AWS VM Individually	Chen	6	6	6	6	4	4	4	4	4	4	4	4	4	3	3	2	2	2	2	2	2	2	2	2	2	1	0	0	0
	Set up AWS VM Individually	Jayam	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	2	2	2	2	2	2	2	1	0	0	0	0	0
	Update ReadMe file	Miao	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
	Update UML Class Diagram	Jayam	4	4	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	3	2	2	2	2	2	2	2	2	0	0	0
	Update UML Sequence Diagram	Prateek	4	5	5	5	5	3	3	3	3	3	3	3	3	3	3	3	3	3	2	2	2	2	2	2	1	0	0	0	0
	Update UML Object Diagram	Carlos	4	5	5	5	5	5	5	5	3	3	3	3	3	3	3	3	3	3	2	2	2	2	2	2	2	2	2	2	0
	Update UML Use Case Diagram	Miao	4	5	5	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	0	0	0	0	0

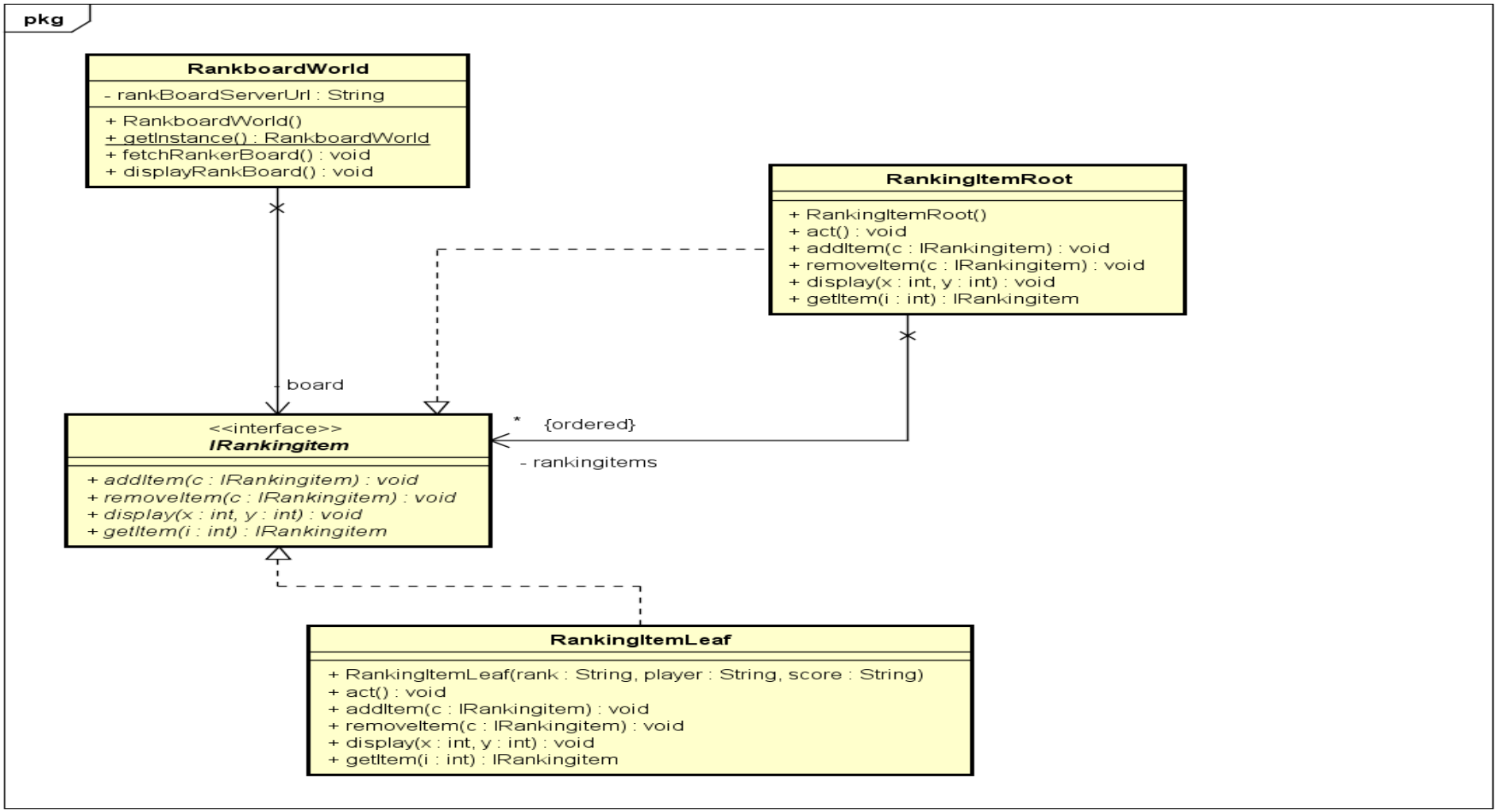
Burndown Chart



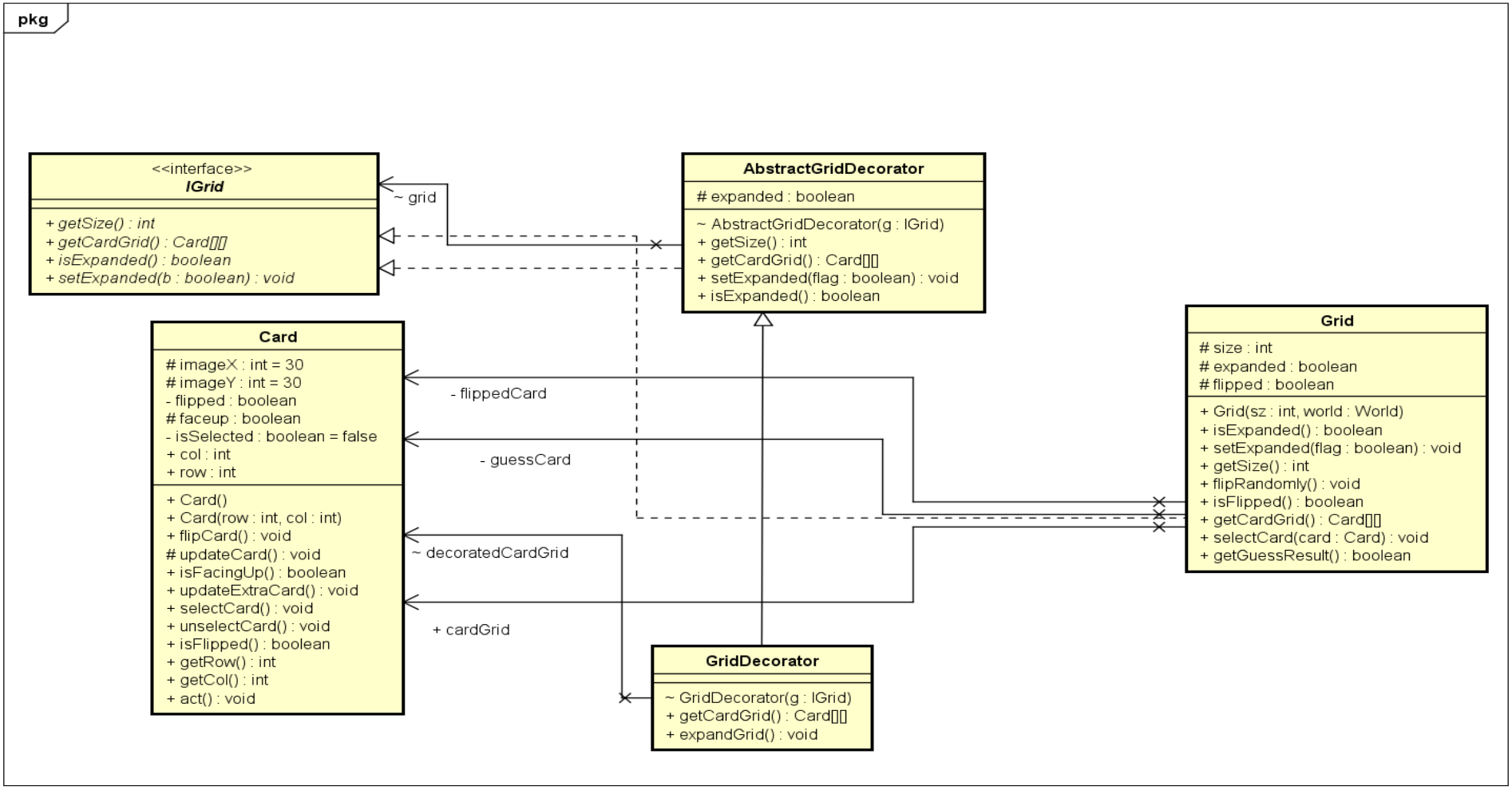
Class Diagram For State Pattern



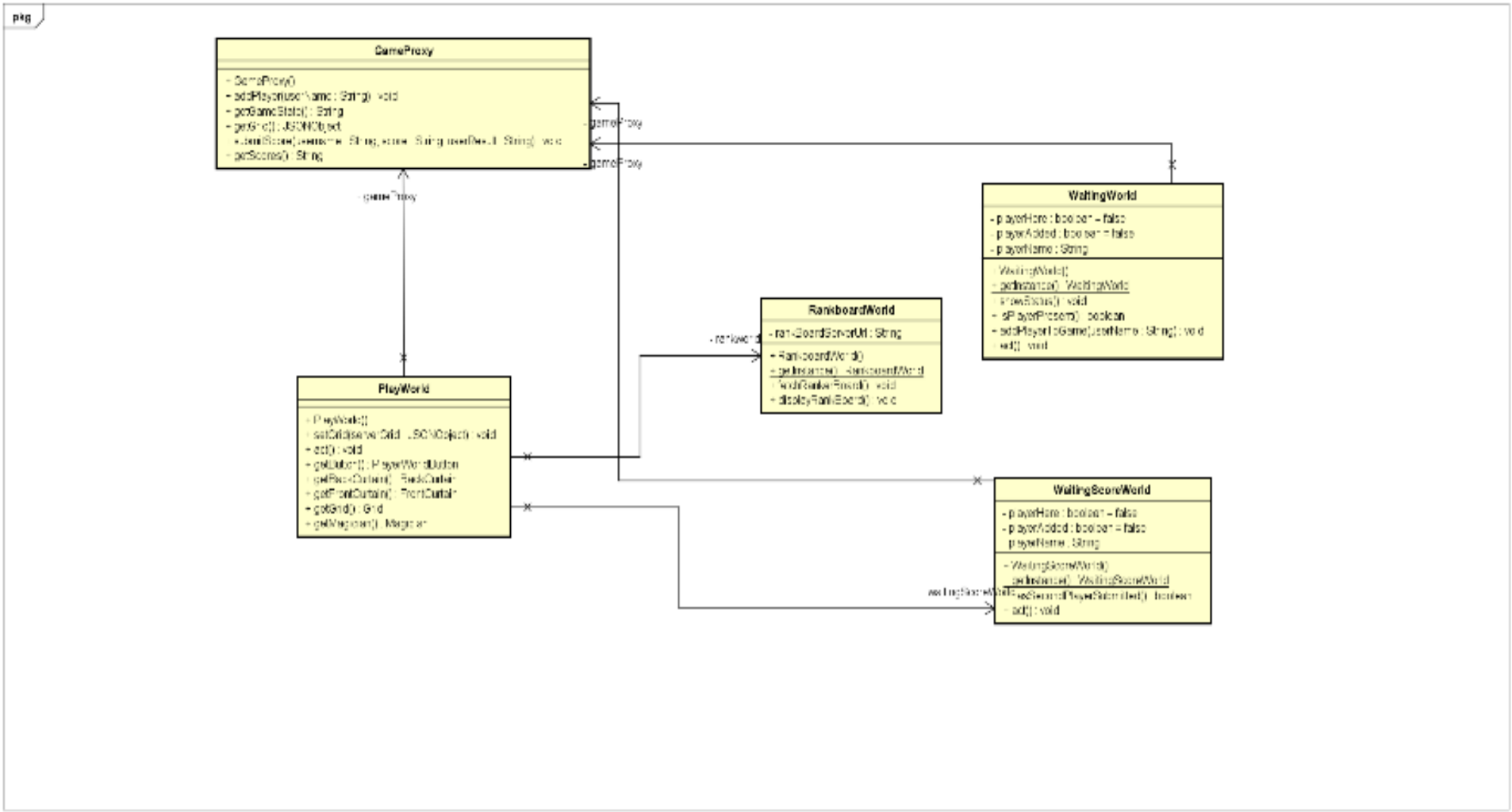
Class Diagram for Composite Pattern



Class Diagram for Decorator Pattern



Class Diagram for Proxy Pattern



Thank You!