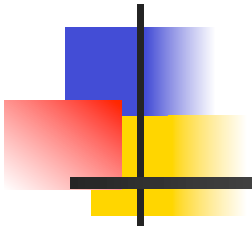
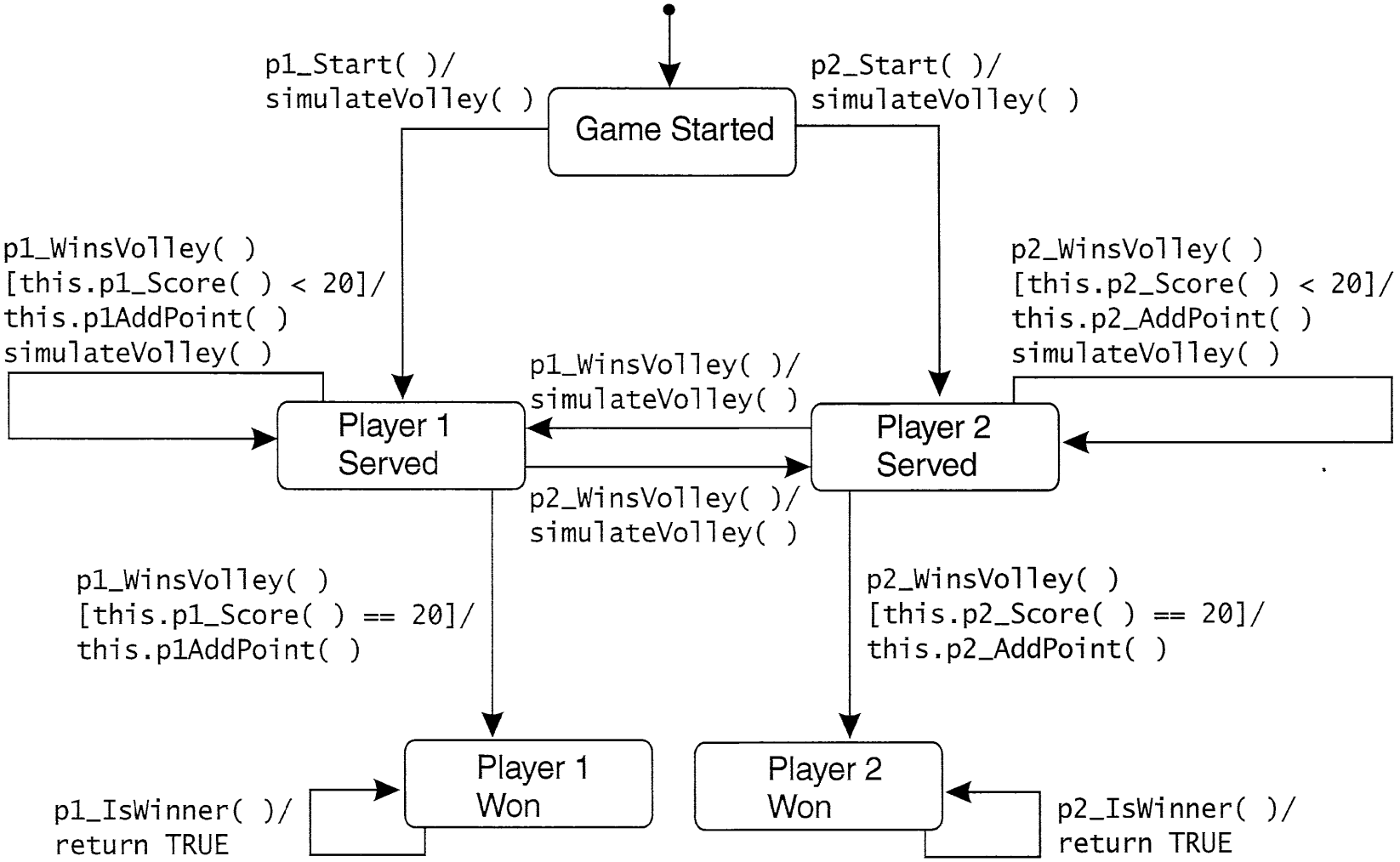
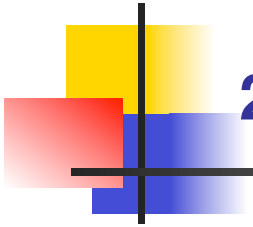


Two-Player Game State Machine

2-Player Game State Diagram





2-Player Game Java Interface

```
class TwoPlayerGame extends Object {
   private int     p1_Points,p2_Points;
   public          TwoPlayerGame( )   {/* Constructor */ }
   public  void    p1_Start( )        {/* P1 serves first */}
   public  void    p1_WinsVolley( )   {/* P1 ends the volley */}
   private void    p1_AddPoint( )     {/* Increase P1's score */}
   public  boolean p1_IsWinner( )     {/* True if P1's score is 21 */}
   public  boolean p1_IsServer( )     {/*  True  if  P1  is  server  */}
   public  int     p1_Score( )        {/* Returns P1's score */}
   public  void    p2_Start( )        {/* P2 serves first */}
   public  void    p2_WinsVolley( )   {/* P2 ends the volley */}
   private void    p2_AddPoint( )     {/* Increase P2's score */}
   public  boolean p2_IsWinner( )     {/* True if P2's score is 21 */}
   public  boolean p2_IsServer( )     {/*  True  if  P2  is  server  */}
   public  int     p2_Score( )        {/* Returns P2's score */}
}
```

State-State Transition Table

Current State	Resultant State/Event/Action				
	Game Started	Player 1 Served	Player 2 Served	Player 1 Won	Player 2 Won
Game Started		p1_Start()	p2_Start()		
		simulateVolley()	*simulateVolley()*		
Player 1 Served		p1_winsVolley() [p1_Score() < 20]	p2_winsVolley()	p1_winsVolley() [p1_Score() == 20]	
		this.p1_AddPoint(); *simulateVolley()*	*simulateVolley()*		
Player 2 Served		p1_winsVolley()	p2_winsVolley() [p2_Score() < 20]		p2_winsVolley() [p2_Score() == 20]
		simulateVolley()	*this.p2_AddPoint();* *SimulateVolley()*		
Player 1 Won				p1_IsWinner()	
				return TRUE	
Player 2 Won					p2_IsWinner()
					return TRUE

Paired Event-State Transition Table

Event	Guard	Current State/Action/Next State				
		Game Started	Player 1 Served	Player 2 Served	Player 1 Won	Player 2 Won
p1_Start()		*simulateVolley()*				
		Player 1 Served				
p2_Start()		*simulateVolley()*				
		Player 2 Served				
p2_WinsVolley()	DC		*simulateVolley()*			
			Player 2 Served			
	p2_Score() < 20			*this.p2_AddPoint(); simulateVolley()*		
				Player 2 Served		
	p2_Score() == 20			*this.p2_AddPoint()*		
				Player 2 Won		
p1_WinsVolley()	DC			*simulateVolley()*		
				Player 1 Served		
	p1_Score() < 20		*this.p1_AddPoint(); simulateVolley()*			
			Player 1 Served			
	p1_Score() == 20		*this.p1_AddPoint()*			
			Player 1 Won			
p1_IsWinner()					*return TRUE*	
					Player 1 Won	
p2_IsWinner()						*return TRUE*
						Player 2 Won

dc = don't care

Separate Event-State Transition Table

Event	Guard	Current State: Game Started	Player 1 Served	Player 2 Served	Player 1 Won	Player 2 Won
p1_Start()		*simulateVolley()*				
p2_Start()		*simulateVolley()*				
p2_WinsVolley()	DC		*simulateVolley()*			
	p2_Score() < 20			*this.p2_AddPoint()* *simulateVolley()*		
	p2_Score() == 20			*this.p2_AddPoint()*		
p1_WinsVolley()	DC			*simulateVolley()*		
	p1_Score() < 20		*this.p1_AddPoint()* *simulateVolley()*			
	p1_Score() == 20		*this.p1_AddPoint();*			
p1_IsWinner					*return TRUE*	
p2_IsWinner						*return TRUE*

Output Action

p1_Starts		**Player 1 Served**				
p2_Starts		**Player 2 Served**				
p2_WinsVolley()	DC		**Player 2 Served**			
	p2_Score() < 20			**Player 2 Served**		
	p2_Score() == 20			**Player 2 Won**		
p1_WinsVolley()	DC			**Player 1 Served**		
	p1_Score() < 20		**Player 1 Served**			
	p1_Score() == 20		**Player 1 Won**			
p1 IsWinner					**Player 1 Won**	
p2 IsWinner						**Player 2 Won**

Resultant State