

The readability of source code has a direct impact on how well a developer comprehends a software system. Code maintainability refers to how easily that software system can be changed to add new features, modify existing features, fix bugs, or improve performance.

Refactoring is the process of changing a software system in such a way that it does not alter the external behavior of the code yet it improves its readability, maintainability, and extensibility. When you refactor you are improving the design of the code after it has been written.

Perform a Code Quality Review of the code provided. Provide specific comments on making alterations to the code to improve its “quality” along with a detailed explanation to support the changes (make reference to one or more specific code smells if one is identifiable).

```

001 public void Update(GameTime gameTime)
002 {
003     bool colliding = false;
004     foreach (var block in _blocks)
005     {
006         block.Update();
007         if (collisionHandler.checkForCollision(mario, block))
008         {
009             if (block.CurrentPowerUpState is HiddenState)
010             {
011                 if (collisionHandler.checkSideCollision(mario, block) == CollisionTypes.Bottom)
012                     mario.Halt();
013             }
014             else
015             {
016                 mario.Halt();
017                 colliding = true;
018             }
019         }
020     }
021
022     if (Mario.invincibleTimer == 0)
023     {
024         if (mario.PowerUpState is FireStarState)
025             mario.ChangeToFireState();
026         else if (mario.PowerUpState is StandardStarState)
027             mario.ChangeToStandardState();
028         else if (mario.PowerUpState is SuperStarState)
029             mario.ChangeToSuperState();
030
031         foreach (var enemy in _enemies)
032         {
033             if (collisionHandler.checkForCollision(mario, enemy) && !enemy.IsDead())
034             {
035                 colliding = true;
036                 enemy.boxColor = Color.Black;
037                 if (mario.PowerUpState is SuperStarState || mario.PowerUpState is FireStarState ||
038 mario.PowerUpState is StandardStarState)
039                     enemy.JumpedOn();
040                 else if (collisionHandler.checkSideCollision(mario, enemy) == CollisionTypes.Top)
041                 {
042                     enemy.JumpedOn();
043                     mario.Halt();
044                 }
045                 else
046                 {
047                     if (mario.isCollidable == true)
048                     {
049                         if (enemy.Hurts())
050                             mario.EnemyHit();
051                         else
052                         {
053                             enemy.JumpedOn();
054                             mario.Halt();
055                             if (collisionHandler.checkSideCollision(mario, enemy) == CollisionTypes.Right)
056                                 ((KoopaTroopa)enemy).ChangeShellVelocityDirection();
057                         }
058                     }
059                 }
060             }
061             else
062             {
063                 enemy.boxColor = Color.Red;
064                 mario.isCollidable = true;
065                 foreach (var block in _blocks)
066                     if (collisionHandler.checkForCollision(enemy, block))
067                         ((KoopaTroopa)enemy).ChangeShellVelocityDirection();
068             }
069             enemy.Update(Viewport);
070         }
071     }
072     else
073     {
074         Mario.invincibleTimer--;
075         foreach (var enemy in _enemies)

```

```

076         {
077             if (collisionHandler.checkForCollision(mario, enemy) && !enemy.IsDead())
078             {
079                 colliding = true;
080                 enemy.boxColor = Color.Black;
081                 if (mario.PowerUpState is SuperStarState || mario.PowerUpState is FireStarState ||
082 mario.PowerUpState is StandardStarState)
083                     enemy.JumpedOn();
084             }
085             else
086             {
087                 enemy.boxColor = Color.Red;
088                 mario.isCollidable = true;
089                 foreach (var block in _blocks)
090                     if (collisionHandler.checkForCollision(enemy, block))
091                         ((KoopaTroopa)enemy).ChangeShellVelocityDirection();
092             }
093             enemy.Update(Viewport);
094         }
095     }
096
097     foreach (var item in _items)
098     {
099         if (item.isCollidable)
100         {
101             if (collisionHandler.checkForCollision(mario, item))
102             {
103                 colliding = true;
104                 item.boxColor = Color.Black;
105                 if (item is Coin)
106                 {
107                     //Add code to add coin to total coins
108                 }
109                 else if (item is Star)
110                     mario.ChangeToStarState();
111                 else if (item is FireFlower)
112                     mario.ChangeToFireState();
113                 else if (item is Mushroom1Up)
114                 {
115                     //Add code to add extra life
116                 }
117                 else if (item is MushroomSuper)
118                     mario.ChangeToSuperState();
119                 item.makeInvisible();
120                 item.isCollidable = false;
121             }
122             else
123                 item.boxColor = Color.Green;
124         }
125     }
126     if (colliding)
127         mario.boxColor = Color.Black;
128     else
129         mario.boxColor = Color.Yellow;
130     mario.Update(Viewport);
131 }

```