

Program 3

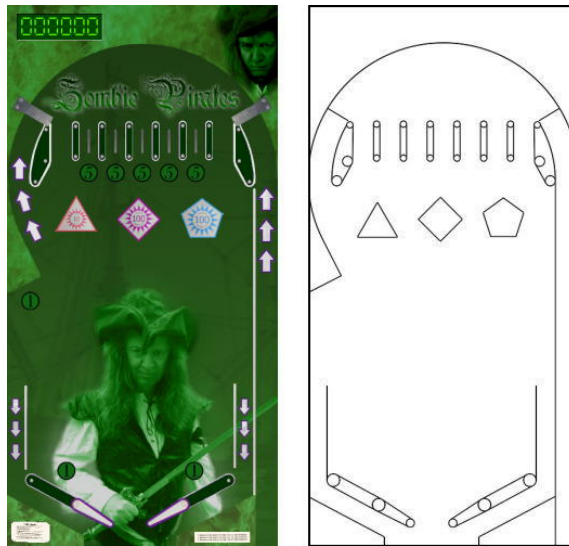
Due 9am, Monday October 25 2021

Your GitLab repository should now contain a folder called Program-3 containing the code for a modified version of the Pinball Game with polygonal bumpers as shown below. Unfortunately, there is no collision detection and response for these new bumpers. Add code to function

```
void CObjectManager::MakeBumper(UINT, const Vector2&, float,  
    eSprite, eSprite, eSound, UINT)
```

in `ObjectManager.cpp` after the comment **YOUR CODE STARTS HERE** on line 292 to add the appropriate shapes to the `CPolygon` pointed to by `pBumper` using its `AddShape` function inherited from `CCompoundShape`. Once you do so, the ball will collide and rebound from the polygonal bumpers correctly, although the bumpers will not light up and the score will not change.

Hint: Start by figuring out how to construct a regular polygon given the position of its centroid, number of sides, and radius (the distance from the centroid to any vertex). The sprites have the centroid of the polygon in the center of the image and the radius of the polygon equal to half the image width.



Students in CSCE 5255, also add code after the comment **YOUR CODE STARTS HERE** on line 679 of `ObjectManager.cpp` in function

```
bool CObjectManager::NarrowPhase(CShape*, CDynamicCircle*)
```

to add the bumper scores to `CObjectManager::m_nScore` and make them light up when hit.

Hint: Examine the code in the **else** block before the comment line to see how this was done for the original round bumpers (which are static shapes). The pointer to the `CObject` for the polygonal bumpers is stored in the user pointer of the polygon's center point, which can be obtained by calling `CPolygon::GetCenterPoint()`.

Once you have completed the task, commit your Program 3 folder before the deadline at the top of this page. The finished program must compile and run without crashing, and your source code must be correct, compact, logical, and legibly commented to receive full credit.