Mt Si AP CS Project 2

Breakout

DUE 02-24-14

Specifications

Breakout is a computer game developed by Atari in 1972 and written by Steve Wozniak and Steve Jobs. It’s also known as Brickles and an updated version called Arkenoid was created. You can view an example of a breakout game here: <http://www.play.vg/games/10-Breakout.html>

This shows the style of game play. You do not need to match the above example in terms of features. You are free to modify the game in terms of look / feel and features but you must meet the below requirements.

1: The player controls a paddle. They are able to move the paddle left and right on the screen using some form of user input (I’d suggest the keyboard). If you opt not to use the arrow keys, make sure you let us know how to control your game either by providing in game instructions or letting us know when you submit it. You may choose to control other elements on the screen or move the paddle in more directions than just left or right.

2: A ball bounces around the screen and is reflected off the paddle. The position on the paddle the ball hits should determine the direction it travels off the paddle. IE, if the ball hits around the center of the paddle, it should bounce straight up. If it hits an edge, it should bounce off at an angle. If the use misses the ball with the paddle, the player loses a life. The player should have at least 3 lives. Once a player loses all 3 lives, the game is over. You DO NOT need to provide a player with a method of restarting the game once it’s over. It’s sufficient to have them close the game and restart it.

3: The screen contains a number of bricks. If the ball hits a brick, the brick disappears and the player’s score is increased. The ball is reflected off the brick based on the direction it hits the brick (the direction should be reversed). If the player removes all the bricks from the screen, the player reaches the next level. All bricks reappear and the ball’s speed is increased making it more difficult.

4: The game displays the player’s score, the current level and the number of lives the player has. These should be displayed at the top of the screen. These should be continuously updated to reflect the current state of the game.

5: At least one powerup and powerup brick is required. Your powerup brick should extend from your brick class and add functionality to it. It should drop a ‘pill’ that contains a powerup. When your paddle ‘collects’ or collides with the pill, you should receive a powerup. It’s up to you what sort of powerup you want (see the example game above for ideas or create your own). If the powerup is time based, make sure you undo the powerup after a set period of time (tip, count the number of frames elapsed).

Extra marks

Don’t go for extra marks unless you get the basic functionality working. If your core functionality does not work, we may decide not to give you extra marks even if you work on them!

Wow factor: Look and feel

What we expect here would be some major changes in the way the game looks and feels. This is not just something small like changing the colors. We would expect things like using images instead of rectangles and ovals. Adding additional game elements may help as well. Use your imagination! Things like altering the feel of the game could also help. Maybe you can improve the controls to make them smoother. Maybe you could make the angle the ball reflects off the paddle smooth rather than having set angles.

We have sound!

We’re not going to tell you how to do this, but it’s pretty simple so you should be able to figure it out yourselves. Sounds should be appropriate and could occur when the ball bounces off something. You could also include sound when the paddle moves. If you really want a challenge, add music to your game!

Cool new features

Sick of the boring old breakout game? Add extra features! Maybe you could have a paddle up the top as well as down the bottom. Different sized balls? More powerups (I like lasers!). See below for some examples of additional features in arkenoid / breakout style games:

<http://www.arkanoidgames.net/>

Checkpoints

In software development, it is very common to define milestones at the beginning of a project. A milestone is essentially a checkpoint along the way when certain functionality is expected to be available. This approach makes it easier to keep track of the overall progress of the project. The breakout project will use checkpoints as well. This will allow you to break down your work into multiple parts that logically build on previous milestones. It is much easier to stay organized that way rather than trying to work on everything at once.

Each checkpoint is worth 2 points each out of 100 for the project. You will still get credit for all functionality you successfully implement – even if it is delivered late. However, by pacing yourself appropriately and turning in each piece of functionality on time you can earn up to 10 points!

Checkpoint 1: Control that paddle!

For this checkpoint, you will need to add a paddle to the game that the user can control. In addition, the ball should bounce off the paddle if it hits it. The direction it bounces off in should be determined by where on the paddle it hits. If it hits the center of the paddle, it should bounce straight up. If it hits the left side, it should bounce to the left. If it hits the right side, it should bounce to the right.

Checkpoint 2: Build a wall and knock it down!

Build a wall of bricks! You should have at least 4 rows of 5 bricks. When the ball hits a brick, it should disappear. The ball should be ‘reflected’ off the brick when it hits it. For example, if it’s travelling up, the ball will bounce down off the brick. In this case, the ball’s trajectory should be based on the direction it is travelling in (unlike when it hits the paddle). You should essentially reverse the X and Y directions of the ball.

Checkpoint 3: Scores lives and levels!

Time to implement scores, lives and levels. The player should start with 3 lives. Every time they miss the ball with the paddle, a life is lost. The score should be incremented for each brick hit. You could also give the player a bonus score for clearing the board if you want. Finally, if the board is cleared, you want to move the player to another level. The ball should move faster on harder levels. In addition, the player should get more points for each brick hit.

Checkpoint 4: Powerups!

Let’s powerup the game with some wicked powerups! Make sure you extend from your brick class. Create a powerup pill when you hit it. Collect it with the paddle and watch magic happen! You could grow the paddle, create more balls, create a laser paddle or create a magnetic paddle! Please don’t just slow the ball down. We’d like a little more than that!

Checkpoint 5: Fit and polish. Debug, debug, debug!

Once the previous checkpoints are done, we’ll give you some time to make sure everything’s working with no bugs. You can use the time to refine your controls, graphics and everything else about your game. You could even get some play testers! Get your brother, sister, mom or dad to play. If that doesn’t work, get a class mate to play. Have them give you feedback and let you know what could be improved!

Grading

The project is worth a total of 100 points broken down as follows (with up to 30 bonus points available):

|  |  |
| --- | --- |
| **Requirement Points** | |
| User can control the paddle | 5 |
| Ball reflects correctly off the paddle | 10 |
| Bricks are drawn and disappear when hit. | 10 |
| The ball reflects correctly off the bricks. | 8 |
| Score is displayed on screen | 5 |
| Level is displayed on screen | 5 |
| Lives remaining are displayed on screen | 5 |
| Lives are decremented. Game ends when none remaining. | 10 |
| Game gets more difficult for each level | 5 |
| Score is incremented on each brick hit | 5 |
| Powerups. Marked brick can be hit and drops a powerup pill that can be collected and helps the player in some way. | 10 |
| 5 Checkpoints (2 points each) | 10 |
| No graphical glitches. No sprites moving off screen | 5 |
| Clear, well commented code. | 7 |
| *Wow factor* | *10* |
| *Sound* | *10* |
| *Cool features* | *10* |