
Gamification and Creating Game Developers

Carol Luckhardt Redfield
CRedfield@stmarytx.edu
St. Mary's University
Computer Science
1 Camino Santa Maria
San Antonio, TX 78228-8524

Abstract

Computer games are being used for educational purposes more and more. Instructors can use games to support learning and retention, and students can design and create computer games so that students can learn the beginnings of computer programming. In fact, the gamification of lessons and courses is becoming popular. Bringing game aspects to learning will require more people to develop games and think through what kinds of gaming principles and processes can be applied effectively to what kinds of learning. This paper discusses educational computer games and presents the activities and assignments for a computer game development class that others can use to teach computer game development. Well over 100 students have developed working games with this curriculum in a six week summer program. Providing a PC with some Microsoft tools (Paint and Office) and game development software (GameMaker), students can learn how to evaluate, design, and develop computer games. In this curriculum, the design process utilizes storyboarding. Students play games, evaluate games, present and demonstrate games, and create a game design, a computer game, and a game manual. The timeframe of the course can be anywhere from three to twelve weeks. Some of the student-made games will be made available for play at the presentation. Learning is fun; let's use games to keep it that way.

Keywords: game development, gamification, computer games, educational games, teaching how to make games

1. BACKGROUND

Children are playing over 10,000 hours of computer games by the time they complete their education, around 21 years old, and people on Earth are playing over 3 billion hours of games each week (McGonigal, 2010). We need to pay attention to games and how we might effectively use them.

All games have the following attributes:

- rules to play by,
- players who agree to play by the rules,
- goals to achieve in the game, and

- feedback or payoff that often is immediate during the play of a game.

When gaming, students often earn points, badges, awards, and trophies, and level up in the game. Small achievements get rewarded, while failing at a small achievement is not a problem for gamers. Gamers go on quests and missions. They sometimes even perform tasks in a group as team.

Gamification and Benefits

Gamification is about bringing these attributes of games and gaming to a lesson or classroom (Kapp, 2012). Games can be used to help fulfill learning objectives. Parts of a lesson or course

can be made game-ish or an instructional game can be made that has activities and goals that support the learning process and outcomes. In general, you bring what works about gaming to instruction, including, hopefully, some fun. In the gamefication of instruction, the students become players, the instructor defines rules to play by, goals are selected based on learning objectives, and feedback supports the goals. Some of the feedback should be immediate.

Why would we want to make educational computer games or gamify anything? Educational games can be very beneficial to those who play them. In fact, educational computer games have been around since early personal computers. Some titles that have been around for decades include the Reader Rabbit and Math Blaster series of games. There are many benefits of computer games that have already been shown. Ferdig (2009) published a 3-volume book on the effectiveness of educational computer games with one study done on a commercially available game, Math Blaster Algebra. Also in that book, Ke (2009) recorded 89 empirical studies about the effectiveness of educational computer games. Most of the games were developed in a laboratory for the purposes of the study and not made available outside of the study's classroom. Holz (2012) reported on many scientific studies on computer games that showed great promise to improve perception, decision making, dexterity, and even creativity. Results showed that practiced gamers can pay attention to six things at once without getting confused, as opposed to four that most people can track. Gamers can act on choices six times a second which is four times faster than most people. Surgeons had improved hand-eye coordination. Computer games can be good for us, and we could use more well-made and thought out games.

In order to create an interest in computer science and in game development, we can teach how to make useful computer games. Also, one of the emerging educational fields is gaming and gamification (NMC and Educause, 2013). With growing game industry and this emerging field, there is and will be a growing demand for game developers. We need good courses to teach how to develop good games.

The rest of this paper presents a game development class that has been used successfully with about 150 high school students

to develop a computer game within six weeks of a summer. All students developed a computer game with at least 3 levels or rooms in the games. This curriculum has been used three times as part of the fourth year of the Prefreshman Engineering Program (PREP) (PREP-USA, 2013). PREP is an academic summer science camp that motivates and prepares middle and high school students for success in further studies in engineering, mathematics, science and technology. This game development class has been very successful and can be generalized for students who already have basic computer literacy and understanding, especially about files and folders. No computer programming is required, yet some will be learned in the class. Those students with a programming background often develop more in-depth games. It is the hope that students would get interested in computer programming from this course.

2. GAME DEVELOPMENT CLASS

Beyond needing instructors and students, each person needs to have access to a Windows-based personal computer with

- Microsoft Word or equivalent,
- Microsoft Paint or equivalent,
- GameMaker 8.1 Lite (Yo Yo Games, 2013) (or some other game development tool such as Unity or Teaching World),
- an email account, and
- access to the Internet.

Students should also have a way to backup their files (Word .docx, Paint .jpg, GameMaker .gm81), such as a USB flash drive or a secondary shared drive. Every session or day, students are required to save their work in two physically different places (you may need to inspect that they have done this – they often do not even when reminded). The classroom should have some typical technology such as a PC, document camera, projector and screen.

Plan on around 24-30 class periods of around 1-2 hours each. The classes could be held over as little as 3 weeks, although at least 6 weeks seems best. Many students will do additional work outside of class times, but it does not need to be required. The following curriculum outlines how the course is taught over six weeks at the summer PREP program. Learning how to use GameMaker is done in a hands-on approach by developing 3 games during class with each student developing the same game with the instructor, although the students can pick

different sprites or images to use. The 3 games shown in this paper were developed by the author although the first two are similar to the tutorials available with GameMaker. Each week's material can be organized for a different time period such as 3 weeks for a fulltime, intensive course, or spread out over about a 9-12 week period.

Learning Outcomes and Assignments

When the course is taught over a six week period, in the first week the instructor should have the students

- get set up on machines, logins if needed,
- understand what a game is, and
- learn how to evaluate a game.

For assignments, students play games with a purpose of looking for what elements make them enjoyable or fun or good and what they might like in their own games. Students also show a game that they like for 1-2 minutes in front of the class over the Internet or on a handheld device with a document camera. The instructor can direct students toward educational games. Many games can be found at the websites below that students can use to play and evaluate. These sites have educational games about math and science:

www.arcademicskillbuilders.com/games
www.coolmath-games.com
www.e-learningforkids.org
www.funbrain.com
www.internet4classrooms.com
www.kidsgames.com
www.learninggamesforkids.com
www.multiplication.com/games
www.playkidsgames.com
www.tryengineering.org/play.php

These websites have other educational games of multiple subjects:

www.clarkchart.com
www.educationarcade.org
www.funschool.kaboose.com
www.gaming2learn.org
www.games2train.com
www.gamesforchange.com/play
www.howstuffworks.com
www.learnxscape.com/apps
www.mission-us.org
www.nobelprize.org/educational
www.playgen.com//play
www.socialimpactgames.com
www.softschools.com

These websites have various, typically non-educational games:

www.armorgames.com

www.flashgamesite.com
www.fupa.com/games/1/playcrafter.html
www.kongregate.com
www.miniclip.com
www.worldofkitsch.com/interactive/games.html

Another source to find games can be found at www.wingz2fly.com (moving soon to www.i-learn.org), a searchable, online database with over 1500 educational computer games cataloged).

In summary, the student assignments for this first week are to

- play games, and
- show a cool game.

In the second week, the instructor presents materials and activities for students to be able to

- understand the game design process with storyboards,
- create sprites with Paint, and
- make 2 simple games (described below).

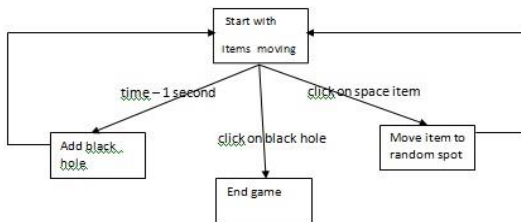
Assignments for the second week include evaluating and writing game reviews of around 5 different games. One of the objectives of the reviews is to make sure students comprehend the difference between what a player actually does (press the space bar, press the up arrow key, hover over an object, click on an area) distinct from what it means in the game (shoot, move an object up, add points, lose a life). The game evaluations should include paragraphs about the game play, ratings for various aspects of a game (such as multimedia, design, support, appeal), and a summary of what is bad and good about a game. Another assignment is for students to decide the main idea of the game they intend to develop. They can make games with combinations of clicking on items, colliding with objects, dragging objects, moving through mazes and rooms, and jumping over obstacles. Some student can learn more about the capabilities in GameMaker by reading manuals and viewing videos online.

During this week, the instructor presents making a simple game that each student will create. The students follow along on their computer doing the same actions as the instructor to create the game. The simple game should have one interface or room with one type of action such as clicking on an object. The instructor should show an example storyboard document to see the kind of pages that students will need to make later. The storyboard document should include a

flow chart of the rooms or levels and draft screens of the rooms and levels. Figure 1 below shows the simple flow chart with the four states of a game called Pick Your Space. The game requires a player to click on moving objects to gain points. (Figures 2, 3 and 4 are in the appendix.)

Figure 1: Flow Chart for Pick Your Space

Figure 2 shows an example storyboard draft screen of Pick Your Space, a possible first game to make. Figure 3 shows the actions to make the



game in GameMaker. Figure 4 shows what the game looks like in the GameMaker interface. Figure 5 below shows an example screen of the Pick Your Space game running.

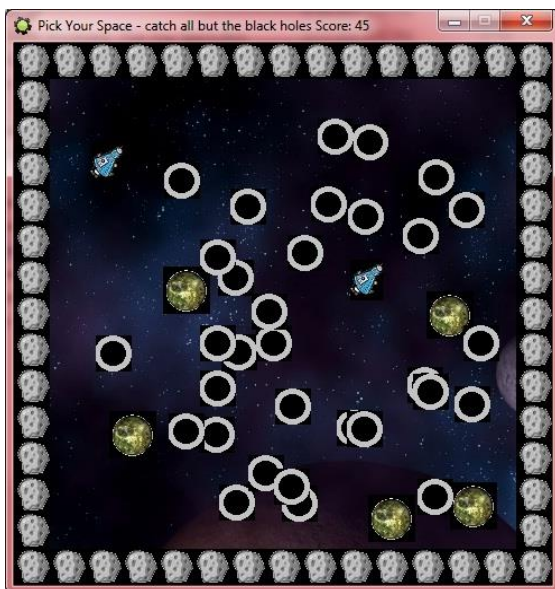


Figure 5: Pick your Space Game

The typical order of activities to create a working game in GameMaker include

1. create a sprite (a small image)
2. create an object
3. tie the sprite to the object
4. add behaviors or events to the object
5. create a room

6. place the object in the room
 Other sprites, objects and behaviors, and rooms can be created to match the design of the game.

The instructor then should lead the class in developing another simple game with a different activity in the game. This second game could be a simple one-room shooter. Figure 6 below shows an example shooter game called Math Shot with the player as an eraser, shooting out chalk at mathematical tools. Figure 7 in the appendix gives the instructions to make Math Shot in GameMaker.



Figure 6: Math Shot

In summary, the assignment for the second week is the

- game evaluations of 5 computer games.

In the third week, students will

- make a multi-room game,
- see some educational games, and
- find or create sprites for their game.

The storyboard document for their game should be completed by the end of this week for their assignment. In the document, they need to discuss how the game will be played, what the rules are, what the goals are, and what the feedback is. A flow chart needs to be included that shows the relationships between rooms or levels, as well as many pages of mock screens. Each room should have a draft screen for how it starts, interactions in the room, and how the room or level ends. Each screen needs a list of the possible actions a player can take in the room or level and the results of taking those actions. The idea is that a developer would have everything needed in order to make the game from the storyboard pages.

The instructor should create one more game with the class. This third game should include multiple rooms to show possible transitions between rooms. Figure 8 and 9 below show example rooms of a maze for the third game. Figure 10 in the appendix shows this third game inside the GameMaker interface.

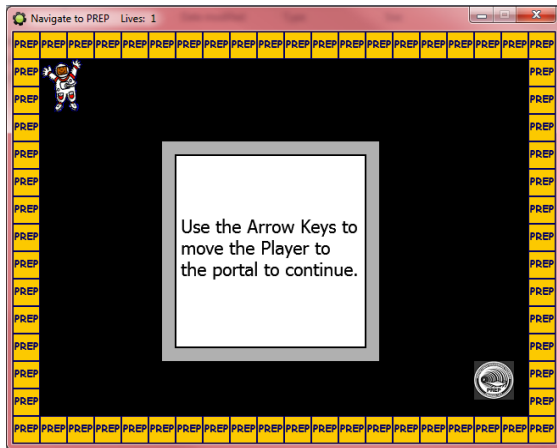


Figure 8: First Room for a Third Game

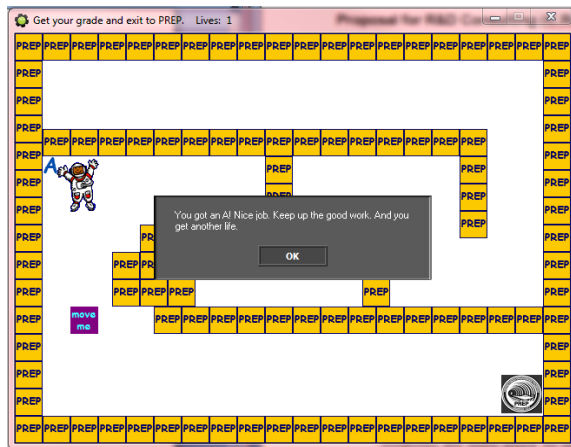


Figure 9: Maze for Multiple Rooms

In summary, the student assignment for the third week is the

- storyboard document.

In the fourth week, students should

- develop sprites, objects with events, and at least one room for their game.

Instructors can share an aspect about gaming that interests them such as artificial intelligence in games. This week, students need lots of time to develop their games. No assignment is due except to show the progress on their games to

an instructor. The instructor should meet with each student to verify work on the game is progressing. The instructor might look for the main idea appearing in the game with at least one room filled with some objects and some behaviors they need, and the plan of 3 or more rooms getting created for the game. Instructors can look at the list of sprites and objects in the game.

The fifth week also should be focused on development of the game with an eye on students finishing the game and starting a game manual document. By the end of this week, students should have completed their game and started on a game manual. Nothing has to be turned in, but again an instructor should meet with each student individually to check on the progress of the game. All of the rooms should be created at this point, with the desired objects in the rooms and events, actions and behaviors on the objects. The instructor should watch out for any students who might need extra help or could use suggestions to simplify or work around something in order to have a working game. Students who have become proficient with the game development tool and are almost done with the development of their games can help other students.

The game manual is to be turned in with or right after the game has been turned in. The game manual should include discussions on

- what the game is,
- hardware and software requirements to run the game,
- design of the game – rework the storyboard pages that show how the game actually looks and runs,
- how to install the game by making an executable,
- how to use the game, and
- some sample sessions of running the game by showing screens shots.

For the fourth and fifth week, there are no assignments to be turned in, but the student should

- show an instructor their progress on the game.

In the final week, students should be encouraged to play and test each others' games, after they have tested their own. In one class, students had their games ready to run on a computer, and all the students went around and played as many games as they could. The

students put nominations on the board up front for games that had or were the

- best game play,
- most depth,
- most unique,
- most educational, and
- most PREP-topic-related.

Awards were given at another class session to the top games in these categories. The game manual and final game are due at this point in the class. Students should show their games to the entire class so that they practice presenting to a group and presenting a product. Knowing that they will be presenting their game to the class may encourage them to do well in the development of their game. The instructor can collect the student game files onto a USB flash drive that can hold all the student games and documents including the .gm81 and .exe files plus the .docx game manual.

In summary, students turn in the

- game they developed, and
- game manual.

Also in the last week, some videos can be shown that are inspirational about computer gaming such as some TED talks at www.TED.org. For example, Jane McGonigal (2010) encourages the audience to play more games to make a better world, and she even has us live longer by following along in her talk on how playing games can lengthen your life (McGonigal, 2012). She recommends that we play games we enjoy up to 21 hours per week (a half-time job!), and with people you know in the same physical space, and in a cooperative or creative mode.

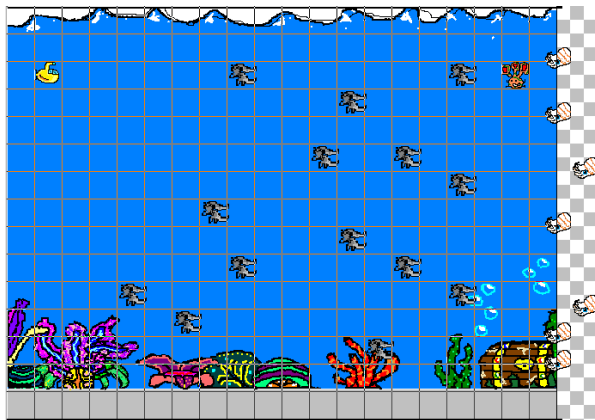


Figure 11: Nautilus Game

Over the three summers that this course has been taught, the games that have been created range from climbing the clouds through the atmospheric layers to re-designs of Tron and Pac-man (making a two-player version) to a set of mazes avoiding obstacles to collect tools to fix your submarine. The students all developed working games with at least 3 rooms or levels. One student put together over 40 rooms for his game. Figures 11-16 show some screens from a



few student games.

Figure 12: Mario Crates Game

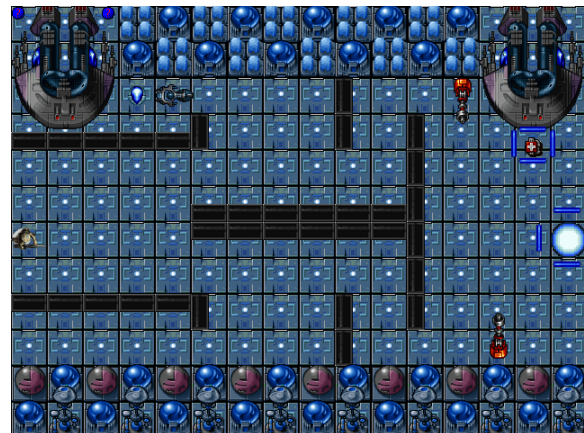


Figure 13: Ziltoidia Attacks Game



Figure 14: Its Raining Cats and Dogs Game

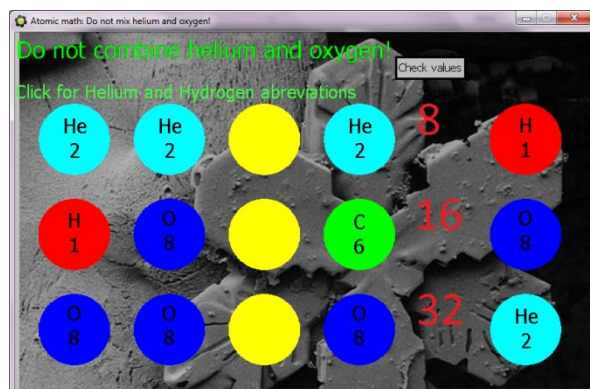


Figure 15: Atomic Math Game

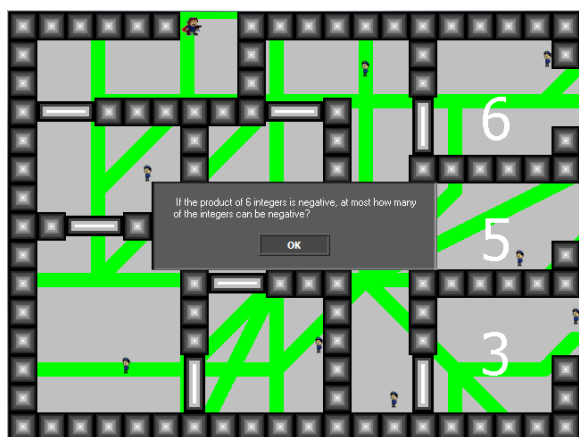


Figure 16: Homework Heist Game

Assessment

The assignments that are due could be graded or not. A minimum standard could be a pass or fail based on if the assignments were completed or not. The PREP program is graded and does provide some motivation for the students and

the grades can count toward their school grade point average if a student selects to have the grade counted. High grades in PREP can lead to some financial scholarships at local universities in the area where PREP is held.

If assignments are going to be graded, the following grading scheme from the student's perspective might be used:

- 10% show a cool game
- 20% game evaluations
- 20% storyboard document
- 10% show your game to the class
- 20% game you developed
- 20% game manual for your game

3. THE CHARGE – GAME ON

As in the PREP program, other students from high school and above can easily learn how to evaluate, design and create computer games using storyboarding techniques and free game development tools. The students only need basic computer literacy skills, not computer programming, although it can help. The fundamentals of game development can be taught within a 3-12 week period with this curriculum. Of course students need to learn computer programming skills if they continue on in game development later. The demand to create games in order to gamify learning, teaching, and instruction is expected to increase, and curriculum such as this one may become more and more valuable.

4. REFERENCES

- Ferdig, Richard (Ed.) (2009). *Handbook of Research on Effective Electronic Gaming in Education*. Hersey, PA: IGI Global.
- Holz, Robert (2012, March 5). When Gaming is Good for You, *The Wall Street Journal*. Retrieved May 26, 2013 from <http://online.wsj.com/article/SB10001424052970203458604577263273943183932.html>
- Kapp, Karl (2012). *The Gamification of Learning and Instruction: Game-based Methods and Strategies for Training and Education*. USA, Pfeiffer.
- Ke, Fengfeng (2009). A Qualitative Meta-Analysis of Computer Games as Learning Tools. In Ferdig, R. (Ed.), *Handbook of Research on Effective Electronic Gaming in*

- Education* (pp. 1-32). Hersey, PA: IGI Global.
- McGonigal, Jane (2010, February). Jane McGonigal: Gaming can make a better world. Retrieved on May 28, 2013 from www.ted.com/talks/jane_mcgonigal_gaming_can_make_a_better_world.html.
- McGonigal, Jane (2012, June). Jane McGonigal: The game that can give you 10 extra years of life. Retrieved on May 28, 2013 from www.ted.com/talks/jane_mcgonigal_the_game_that_can_give_you_10_extra_years_of_life.html.
- NMC and Educause (2013). NMC Horizon Report: 2013 Higher Education Edition. Retrieved August 29, 2013 from <http://net.educause.edu/ir/library/pdf/hr2012.pdf>.
- PREP-USA (2013). PREP-USA: About: PREP Mission Statement. Retrieved May 25, 2013 from www.prep-usa.org/portal/main/generaldetail.asp?ID=296.
- YoYo Games (2013). GameMaker for Windows. Retrieved May 27, 2013 from www.yoyogames.com/gamemaker/windows (as of Aug 29, 2013, for a fee at www.yoyogames.com/legacy).

Appendix – Figures 2, 3, 4, 7, and 10

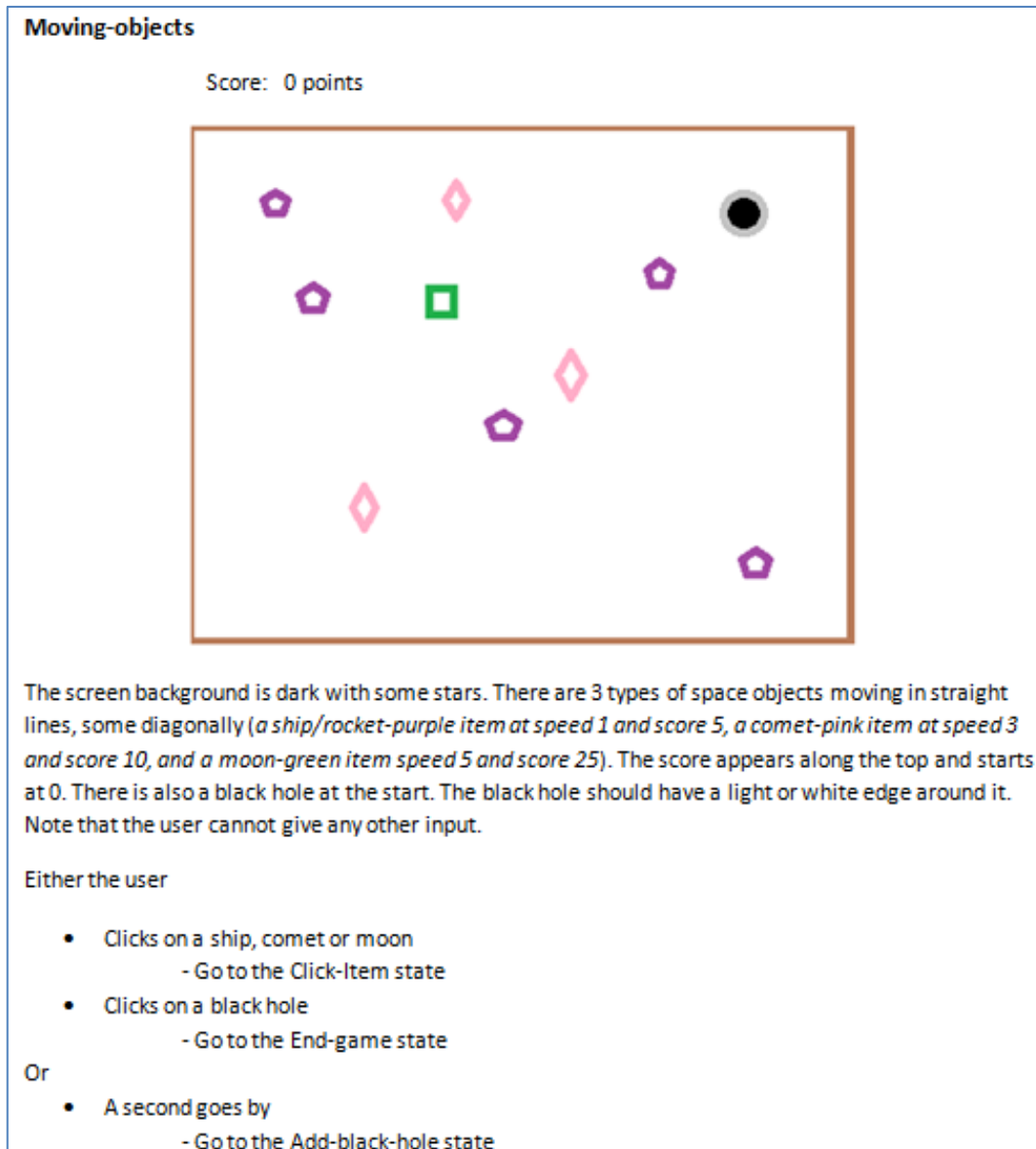


Figure 2: Example Storyboard Screen Page of Moving-Objects State of Pick Your Space

INSTRUCTIONS-actions FOR A FIRST GAME in GameMaker (show storyboard for PickYourSpace) Doc Redfield

Either make or select_sprites (Paint - resize to 32x32; outside color grey, inside color black, circle for blackhole)

GameMaker (continue with Lite; Yes to Advanced Mode; click or select each --; sound optional)

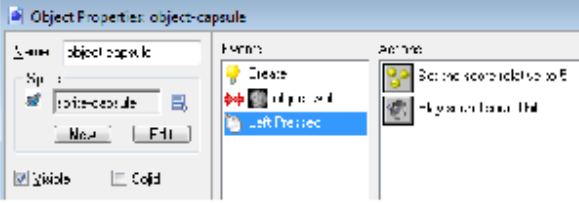
create sound (speaker): Name **sound-hit**, Load sound, select **laser**, Ok (for clicking on object)(bring ear buds)
+ sound: Name **-hole**, Load sound, select **miss**, Ok (for black hole click)

create sprite (red ball): Name **sprite-wall**, Load sprite, select **wall**, Ok
+ sprite: Name **-planet**, Load sprite, select **planet**, Ok
+ sprite: Name **-capsule**, Load sprite, select **capsule**, Ok
+ sprite: Name **-blackhole**, Load sprite, select **blackhole**, Ok

create background (pic): Name **background-nebula**, Load sprite, select **nebula**, Ok

create object (blue ball): Name **object-wall**, Sprite list, select **-wall**, Solid, Ok
+ object: Name **-capsule**, Sprite list, select **-capsule**,
Add event - Create; drag Action- move-MoveFixed
Click all directions (blue arrows), Speed 3, Ok
Add event-Collision - object-wall; drag Action- move-Bounce, Ok
Add event-Mouse-LeftPressed; drag Action- score-SetScore 25, Relative, Ok
drag Action- main1-PlaySound, sound-hit, Ok

Ok



right click on **object-capsule**: Duplicate:
Name- **object-planet**,
Sprite - **-planet**,
select Create, double click MoveFixed - click directions, speed 1, Ok
select LeftPressed, double click SetScore 5, Ok

create room (empty window): Settings: Name **room-space**, Caption **Pick Your X**, Width 480 (640), Height 480
Background: select menu **background-nebula**
Objects:
Object to add menu: **object-wall**, click to add around (right click to delete)
Object to add menu: **object-capsule**, click to add 5
Object to add menu: **object-planet**, click to add 3

[double click to edit an item]
(repeat to test each time) File | Save; File | Run normally (F5) - TEST IT

Create object: Name **object-blackhole**, Sprite list, select **-blackhole**
Add event - Create, drag Action move: JumpToRandom, Ok
drag Action main2-SetAlarm, number of steps 30, Ok (1 sec)
Add event - Alarm, Alarm0, drag Action main1-CreateInstance, object **-blackhole**, Ok
Add event - Mouse- LeftPressed; drag main1-PlaySound, **sound-hole**, Ok
drag main2-Show Message- You lose..., Ok
drag score-ShowHighScore, may change colors, Ok
drag main2-EndGame

in (double click) **room-space**: select **object-blackhole**, click in room to add 1
double click Game Information: type in game information and help instructions (for F1) [move order of rooms to test]
File | Create Executable - **PickYourSpace.exe** (go find the exe and run it or run from in GameMaker)

Figure 3: How to Make Pick Your Space with GameMaker

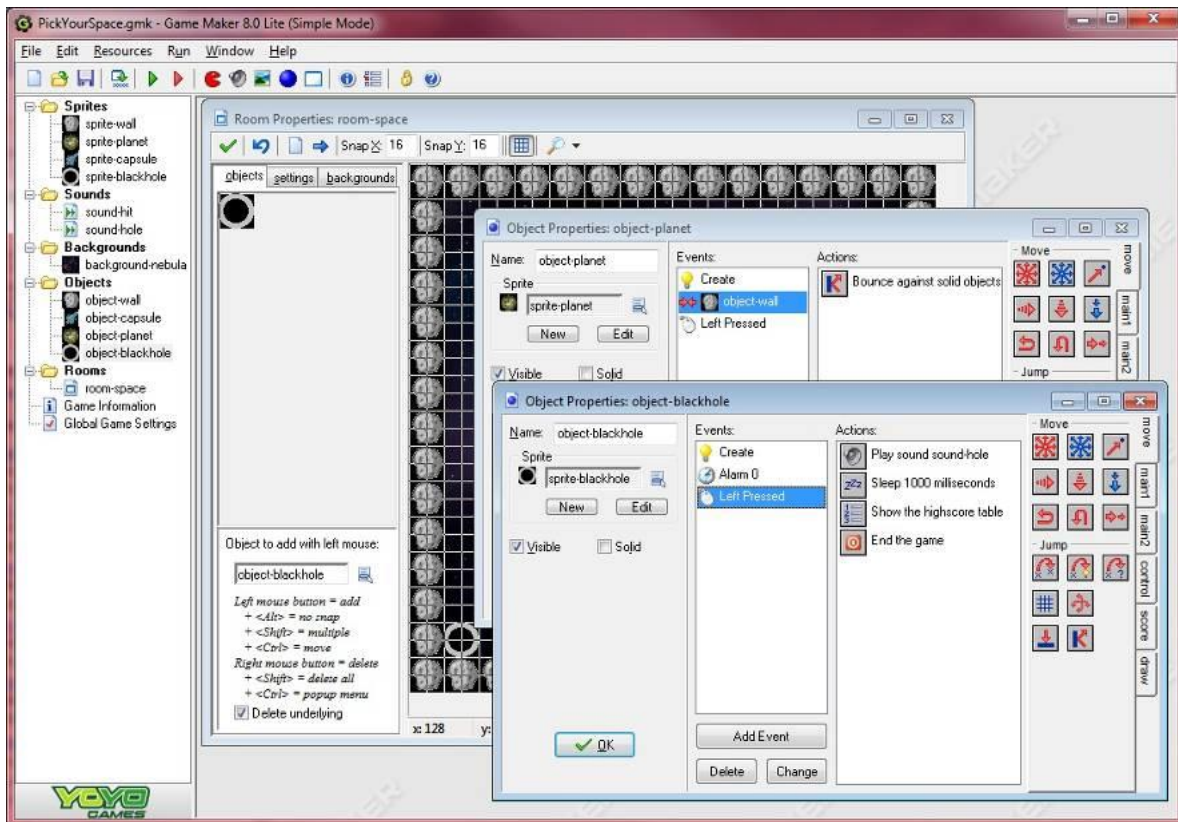


Figure 4: Pick Your Space in the GameMaker tool

INSTRUCTIONS-actions FOR A SECOND GAME in GameMaker Doc Redfield
need images for shooter, bullet, 3 or so targets, background (cloud), and 3 sounds

GameMaker (continue with Lite; click or select each -- ; double click to edit an item)
double click Game Information: type in game information and help instructions (for F1)

create sound (speaker): Name **sound-shoot**, Load sound, select **-sound1**, Ok (for clicking on object)
+ sound: Name **-hit**, Load sound, select **-sound2**, Ok (for black hole click)
+ sound: Name **-done**, Load sound, select **-sound3**, Ok (for losing)

create background (pic): Name **background-digits**, Load sprite, select **digits**, Ok
create sprite (red ball): Name **sprite-shooter (eraser)**, Load sprite, select **image1**, Ok
+ sprite: Name **-bullet (chalk)**, Load sprite, select **image2**, Ok
+ sprite: Name **-target1**, Load sprite, select **image3**, Ok
+ sprite: Name **-target2**, Load sprite, select **image4**, Ok
+ sprite: Name **-target3**, Load sprite, select **image5**, Ok (for a third target)

save
create object (blue ball): Name **object-shooter**, Sprite list, select **-shooter**, Solid,
Add event- KeyBoard-Left, Jump to position -3,0 Relative, Ok
Add event- KeyBoard-Up, Jump to position 0,-4 Relative, Ok
Add event- KeyBoard-Right, Jump to position 3,0 Relative, Ok
Add event- KeyBoard-Down, Jump to position 0,4 Relative, Ok
+ object: Name **-bullet**, Sprite list, select **-bullet**, Solid
Add event - Create; drag Action- move-MoveFixed- right, speed 7, Ok
Add event Other-OutsideRoom- main1-destroy instance
+ object: Name **-target**, Sprite list, select **-target**, Solid
Add event - Create; drag Action- move-MovingFixed-left, 4, Ok
Add event Other-Outside room; drag JumpToPosition- 640, random(480)
Add event- Collision with **shooter**, drag Action main1-Play sound, **cannon**
drag Action main2-message, game over
drag Action main2-EndGame

save
right click on **object-target1**: Duplicate: for target2 (and OTHER TARGETS)
Name- **object-T2+**,
Sprite - **-T2+**,
select/set: Create-moving speed (can also change collision- sound, message)

double click **object-bullet**: Add event -Collision - **object-target1**; drag Action- SetScore, 10 Relative, Ok
drag play sound
drag main1-destroy instance- self
drag main1-destroy instance- other (or object to get all)
drag create instance(self), **target1**, 640, random(480)
Duplicate collision-event FOR EACH TARGET; change score values, destroy, create

double click on **object-shooter**: Add event -KeyPress-space, Create instance: object **bullet**, 35,10, relative, Ok
create room: Settings: Name **room-shooting**, Caption **Shoot the targets**, keep width 640 + height 480
Background: select menu **background-clouds**, Hor. Speed -1
Objects:
Object to add menu: **object-shooter**, click to put middle left side
Object to add menu: **object-target1**, click to add one to far right
Object to add menu: **object-target2**, click to add one to far right
Object to add menu: **object-target3**, click to add one to far right

(repeat to test each time) File | Save; File | Run normally (F5) – TEST IT
File | Create Executable – **ShooterYourTarget.exe**

Figure 7: How to Make Math Shot with GameMaker

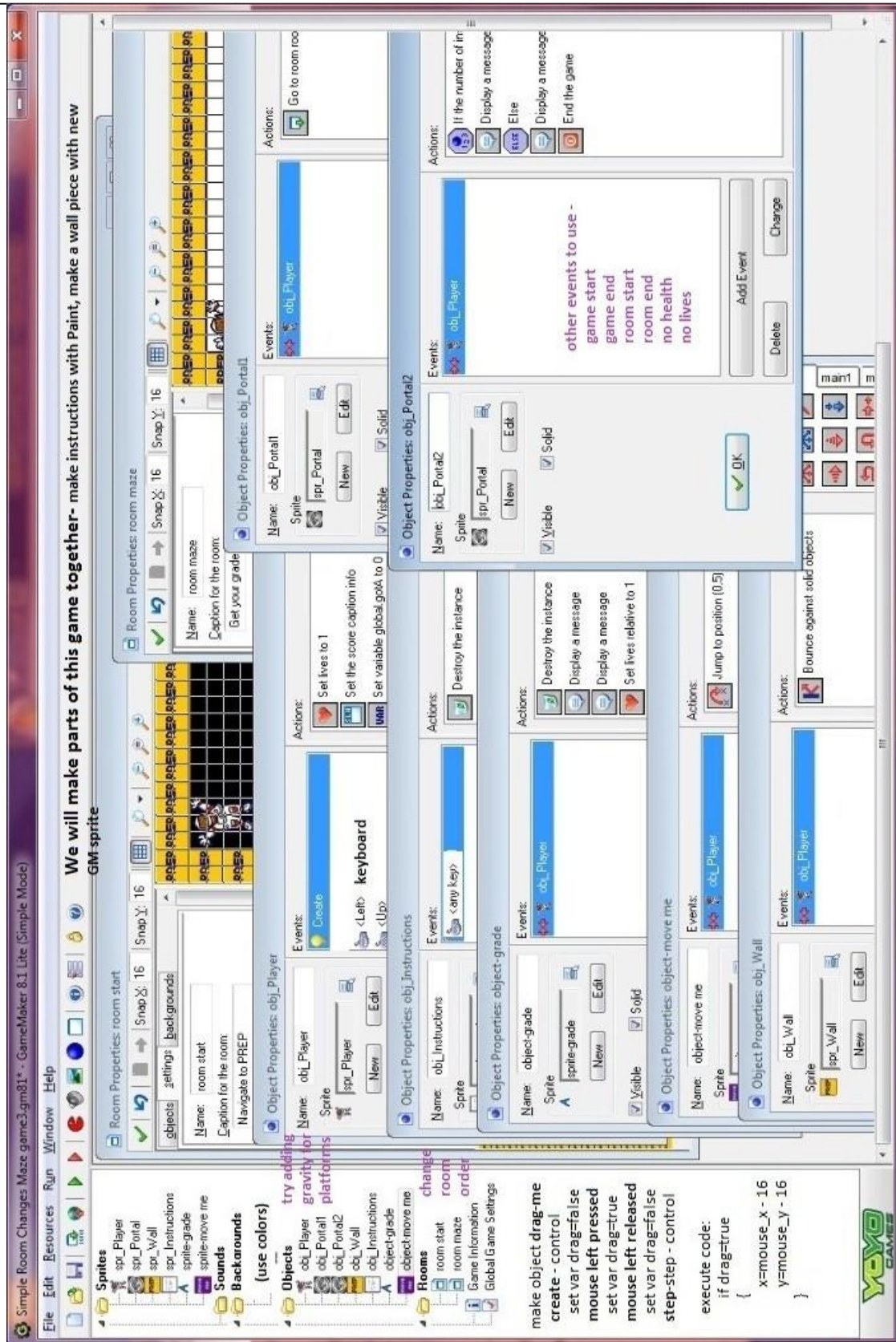


Figure 10: Multiple Room Game in GameMaker