Aryabhatta Journal of Mathematics & Informatics Vol. 6, No. 1, Jan-July, 2014 ISSN : 0975-7139
Journal Impact Factor (2013) : 0.489

"SUPER COLLAPSE PUZZLE" - PLAY AND CREATING SPACE

Sanjay Jain* and Chander Shakher**
*Department of Mathematical Sciences, Government College, Ajmer
Affiliated to M. D. S. University Ajmer, Ajmer-305001, India.
**Research Scholar, Mahatma Gandhi University, Meghalaya
E-mail * drjainsanjay@gmail.com, ** csghorela200921@gmail.com

ABSTRACT:
Space is reconfigured through the participations of both gamers and the game, where game is understood as the programming and hardware of a game technology. Extending our understandings of the contributions of both gamer and game, the outcome of play emerges as the agencies of each are co-constituted. This space is recursive, based on feedback between the state of the game (relations between the objects) and the state of the gamer, which includes their knowledge, skill, mood and attention. The idea of recursive space is developed in two ways. First, as another means of describing a gamer's engagement with space, one that gives a greater account of the participation of technology. Secondly, it gives us a way of thinking about play as a process of creating space.
Keywords: *Space, interaction, recursive space, action, game mechanics, puzzle*

1. INTRODUCTION

Things are really kicking off in "Super Collapse Puzzle" Game. In this game Tiles with more than 3 types (Max 5 in our game) are fitted on whole space, group of random number of tiles are put together to fill complete area of game, when all tiles are available for play, the final phase has begun. The game play of "Super Collapse Puzzle" is based on remove group of 3 or more similar tiles with touch on any of one in group to remove and eating the gamer's brain. The space being generated during play is not straightforwardly a representation of physical space as seen within the frame. It is also a reconfiguration of the relationships between the objects that coexist on the screen. As each configuration of objects sets the parameters of what is seen on the screen, these configurations are often refereed to as space. As the Tiles disappear, the remaining objects reconfigure and a different space is generated. The game play of "Super Collapse Puzzle" makes clear one of the most distinctive features of space in games- we engage with it and create it at the same time.

The idea that a gamer creates space brings with it questions about how space is constructed and by whom. This article argues that space is reconfigured through the participations of both gamers and the game, where game is understood as the programming and hardware of a game technology. Extending our understandings of the contributions of both gamer and game, the outcome of play emerges as the agencies of each are co-constituted. This space is recursive, based on feedback between the state of the game (relations between the objects) and the state of the gamer, which includes their knowledge, skill, mood, and attention. The idea of recursive space is developed in two ways. First, as another means of describing a gamer's engagement with space, one that gives a greater account of the participation of technology. Second, it gives us a way of thinking about play as a process of creating space.

Sophisticated spatial frameworks exist in game studies, and these provide a means of thinking through a gamer's social and cultural experience of the different spaces associated with playing a game (Nitsche, 2008; Stockburger, 2007). The extent to which engagement involves creating space is less often discussed. We know, for instance, that the programming of a game generates the game world, and so technology is clearly involved in creating a space. We also know that gamer input is central to play. Espen Aarseth used the term ergodic to describe the interactive participation of gamers with games (Aarseth, 1997). Since then the terms immersion (Murray, 1998), play, and flow (Csikszentmihaly, 1990) have become familiar. Recently, these often broadly stated modes of engagement have been extended to give a nuanced range of possible interactions. Engagement can be based on both fun or grind, depend- ing on what aspect of the game is engaged with by the gamer (Taylor, 2007). Engagement can also accumulate across narrative, performance, tactics, spatial, affective, and also shared involvements (Calleja, 2007). Alongside these rich and ongoing debates, there remains the question: how is a gamer involved in generating space?

Space is actively created when a gamer becomes entangled with the game world and the possibilities of a game's code. Through this entanglement, a gamer encounters space in two different modes. Briefly stated, a gamer's encounter occurs between the object/objects they control on the screen (an avatar or object to be moved) and also represented space (the location of any actions that occur in the game world). In playing, we see represented space and generate it too via our inputs. The inputs of a gamer are usually motivated by the necessity to move an object or avatar, and the space visible within the frame reconfigures accordingly. In this sense, a gamer's input creates space as a pathway is cleared from, say, encroaching zombies. epresented space has particular orientations and/or perspectives. aurie Taylor uses insights from Gilles eleuze and Fel x Guattari (1987) to additionally argue that the game world consists of multiple configurations of space (Taylor, 2006). This point is also made by Michael Nitsche, who states that part of play involves shifting between different configurations of space: ‘‘the constant renegotiation of the player's position is key and the multi- layered quality of the game space opens it up for dynamic engagement (Nitsche, 2008, p. 181).’’ Both Taylor and Nitsche effectively describe the complexity of engagement with game world space as a dynamic shifting between different organizations of space. We can add to our understanding of engagement by looking at how a gamer is involved in generating the configurations of those spaces as well as traversing them.

2. CONSTRUCTING GAME SPACE

What do we already know about constructions of game space? While discussing games as spatial art, Henry Jenkins and Kurt Squire make the point that game worlds are constructed environments:

Game worlds are totally constructed environments. Everything there was put on the screen for some purpose—shaping the game play or contributing to the mood and atmosphere or encouraging performance, playfulness, competition, or collaboration. If games tell stories, they do so by organizing spatial features. If games stage combat, then players learn to scan their environments for competitive advantages. Game designers create immersive worlds with embedded rules and relationships among objects that enable dynamic experiences. (Jenkins & Squire, 2002, p. 65) Jenkins and Squire describe games as telling stories by organizing spatial features, designed in part via relationships among objects. We can further explore the idea that game worlds are constructed environments, asking how the intersection of a gamer with a game's technology and its embedded rules leads to a dynamic experience engendered by the changing relationships between objects. Espen Aarseth writes that ‘‘Computer games are essentially concerned with spatial representation and negotiation, and therefore a classification of computer games can be based on how they represent—or, perhaps, implement—space (Aarseth, 2000, p. 154).’’ The word implementation brings with it the suggestion that the technology, the hardware of a platform, and specific program of a game, constructs space. Most research, how- ever, has gone instead in the direction of spatiality, a complex view of a gamer's experience of space developed through Henri efebvre's idea of social space: ‘‘[Social] space is not a thing among other things, nor a product among products: rather, it subsumes things produced and encompasses their interrelationships in their co-existence and simultaneity (efebvre, 1991 , p. 73).’’ Central to efebvre’s notion of the ‘‘production of space’’ are the relations between different kinds of spatial experience, an approach used to explore a range of modalities of engagement with digital games (Jenkins, 2002; Nitsche, 2008; Stockburger, 2007; Walz, 2010).

Playing when so described as the production of space, involves a gamer in the process of constructing an experience of spatiality. A gamer actively participates in producing an experience through a combination of engagements with culturally embedded sound and imagery, a physical relation with a game interface, ‘‘imaginative’’ connections with the game, its rules and objects, as well as the avatar’s of other players in multiplayer gaming environments. Axel Stockburger argues:

Game space clearly has to be regarded as a cultural product and practice that is informed by spaces created through the use of verbal signs or language (narrative spaces), yet it appears equally informed by a spatial practice operating on the basis of bodily involvement in the form of gestures (user action) as well as non-verbal sets of symbols and signs (representational spaces). All of these dimensions of space are equally present in digital games and are constantly mediating between each other. (Stockburger, 2007, p. 228)

Stockburger uses the term mediation to signal the centrality of relations between spatial dimensions in any construction of spatiality, and the effectiveness of mediation relies on achieving a coherent experience appropriate to the game and mode of play. A turn-based game such as Starship Unlimited, for instance, has a different rhythm to a real-time strategy game but equally generates a coherent experience of play.

Lefebvre’s influence has not lead very obviously to questions about technology. The emphasis on relations between is, however, central to how gamers and games interact. This does not simply mean the physical connection of making something happen, but the coming together of gamer and code as both are involved in reconfiguring the objects on a screen, a process that generates configurations of space. Science and technology studies use the idea of mediation in ways that are useful for thinking about how gamers interact with technology. For instance, technological objects are understood to mediate, in the sense that they can mediate, transform, distort, and modify an input (Latour, 2007). In interactions between objects, including technological and human ones, inputs are modified by the mediating influences of other objects. From this, any outcome of play can be described as emerging through the intersection of gamer and technology. In “Super Collapse Puzzle”, I have a range of options through which to touch the object to remove from screen and make space, these are given by the game. The ones I can actually use depend on how my play has gone up to that point. The game and gamer mediate each other, the outcome, which layout of Tiles/Objects I use to draw the space, is never completely in my control. Just as is true for efebvre’s view of spatiality, an outcome can be neither reduced to a single aspect of the gamer nor of the game, but is given by the connections between. ucy Suchman’s work has been influential in describing interdependent intersections between humans and computers via the trope of a ‘‘situated action’’ (Suchman, 2007). The term situated already has some purchase within game studies. Steffan Walz cites Suchman’s argument that machines are cocreators of situatedness when he suggests that media have become embedded in people’s lives, to the extent that they are treated as though they are part of the ‘‘real’’ environment: ‘‘media have become indistinguishable from real life (Walz, 2010, p. 25).’’ That is, gamers respond to media technologies from within the range of reactions that are also brought into action during our experiences of the actual world.

The idea that gamers treat media as though they were part of an environment tends, however, to obscure the difference technology makes. Its mediating role is transparent. Thinking about play has always had the potential to reintroduce the intersections of gamer and game. For instance, many commentators, beginning with Aarseth, have pointed out that a player's input into the game is central to play (Aarseth, 1997). Recently, Michael Nitsche put it in the following way: ''the player in a video game is both reader (of the computer's output) and producer (via input of events). For video game spaces, this means the player not only enters the game worlds but also changes them and their ingredients (Nitsche, 2008, p. 31).'' How the player enters the game world can be dealt with through perspective and immersion, but what is at stake here is knowing more about how a gamer changes the game world and its ingredients or, how the gamer and game mediate one another. Often, the change brought about by a player is meant in terms of taking over or conquering space (Jenkins & Fuller, 1995; Newman, 2004). For instance, a gamer has fought their way through numerous levels (Max Payne), completed a set of tasks (Monkey Ball), achieved their highest score (Brick Breaker), won the race (Dirt), solved a series of puzzles (Zen Bound), completed a journey (The Longest Journey) or participated in a multiplayer strategy game (RuneScape). Though a gamer's input is clearly crucial to achieving these ends, such descriptions do not really draw out the ways in which an input changes the configurations of the objects that make up space. They are closer to a description of how the avatar and/or gamer's relationship with an already configured space changes through the duration of play.

Introducing the idea of recursive space aims to make clearer that the input of the gamer not only intersects with the action-based elements of play (whether they are driving, fighting, running, or puzzle solving) but also with the configurations of the objects making up visible space. The word recursive has been chosen to describe this intersection because of the way in which both a gamer and game refer back to each other. This referring back occurs through a combination of both seeing and also creating space. When something is recursive, it involves a repeated procedure in which the outcome of each step is defined in terms of the results of previous steps. The use of the term does not come with the rigour of a mathematical function; however, it does signal that the input of a gamer is made in relation to the imagery already on the screen. The imagery on the screen combines objects not directly responsive to a gamer's input, and objects responsive to input, an avatar for instance. The relationship between all these objects defines space. In recursive play, the imagery reconfigures in relation to an input, which provokes another input, which again reconfigures the imagery. Take for instance; the iPod Touch and iPhone game touch Physics where the idea is to remove an object or a group of objects, so that it makes contact with another object. Other objects on the screen remain unresponsive to input but are nevertheless part of the game's space. Playing involves thinking within the dynamics of weight-bearing space, drawing counterweights and planks in such a way as to ensure the maximum similar type of connecting objects (Tiles) are removed. In drawing counterweights and planks, a gamer can be said to close off all the possible ways the objects constituting space might be reconfigured as they opt for a particular action. As a consequence of this action, space is reconfigured, another space is generated, and that offers a further set of possibilities for transformation. Though the game is visually simple, touch Physics demonstrates this recursive transformation explicitly. The task is to get a group to touch to create empty space. There is any number of ways a gamer might do this. The capacity to transform space is less obvious in more visually complex games but remains central to our experience of games, as one transformation leads to another and then another, so generating a recursive space. In more sophisticated games, each time a gamer enters a space something different can happen. And with each input, a gamer alters the configurations of objects generating a different space.

So far I have talked about gamers and how they create space. But how can we understand the ways that a game too creates space, making the contributions of technology to the generation of space more explicit. Alexander Galloway comments:

But a video game is not simply a fun toy. It is also an algorithmic machine and like all machines functions through specific, codified rules of operation… I adopt the terms ''operator'' and ''machine'' not to diminish the value of fun, meaningful play but to stress that in the sphere of electronic media, games are fundamentally cybernetic software systems involving both organic and nonorganic actors. (Galloway, 2006, p. 5)

In drawing out the place of technology in constructing space, I am not aiming for a deterministic understanding of technology (Dovey & Kennedy, 2006). We already know that the game does not exist in any active sense without the input of the gamer. Similarly, recursive space relies on the game technology and the gamer: together they create the space through an interaction involving feedback into the state of the game, the view of the game, and between the two.

3. APPROACHING RECURSIVE SPACE

Recursive space proposes a mode of engagement in which the gamer is both embedded within a space defined by the organization of objects, and also creating that space at one and the same time by altering the organization of objects. If only thinking about the game, the space it generates is precisely defined through the coding of a program. The execution of the program code builds objects, and the relations between these objects present a specific space with particular orientations and perspectives. In such a view, the landscape of a game world or the parameters of a puzzle might seem to be a space, based on the relations between objects, into which a gamer enters. However, through the very fact of entering a game world or beginning a puzzle, a gamer alters the configurations of objects making up space. In a game, the capacity to interact is technologically mediated; entering the game involves the gamer making inputs that are fed back into the program. This feedback is relational, emerging from and into the configurations of the programming.

To understand this relational intersection, it is useful of think some more about how game technology works and the means through which it responds to the input of the gamer. My starting point is to suggest that in games, the imagery seen within a frame consists of independent elements, and the combinations of these elements form what we can call the ''space-in-a-frame.'' In Angry Birds, the elements include yellow, red, and blue birds to be used as missiles by the player, and various glasses, stone, and wooden structures inhabited by chuckling green pigs that are targets for the bird missiles. The game technology configures these elements separately for each and every frame in response to a gamer's input to ensure the illusion of real- time play. When a player uses the touch screen to catapult a bird toward the pigs, the frame is reconfigured to show the appropriate changes: A bird in the catapult sling pulled to the position, given through the touch screen input, any point along the trajectory of the bird, the point of impact, and the subsequent major or minor damage to the structures. Consequently, the inputs from the gamer are incorporated into the spatial configurations of each frame.

The space-in-frame is, then, built up from a series of elements that combine to make up the images. When speaking in terms of program code such elements are often referred to as objects. Many of these objects, including character models, are separately available within the coding of the program. That is, they exist as constellations of electronic data that can be called as a consequence of both the code of the program and the gamer's input (Keogh & Giannini, 2004). In influencing how and when these objects are called, a gamer's inputs have the potential to shape the configurations of space. The intersection between gamer and code is described in the following section through a discussion of object space, camera space (the orientation of objects visible on the screen), and finally the different ways in which a program can respond to input by ''calling'' objects into a frame. Game code, whether for the complex simulated environments of Call of Duty or the geometric shapes of Tetris, is written in a way that assumes interaction. Object space refers to the space taken up by any object. It defines the space taken up by a tree, a table, chair, vehicle, animal or human, or some more abstract entity. Objects are notionally tethered to an origin (a mathematical construct), which gives the objects their orientation and also their relationships with each other (Rabin, 2005). As each object is separately tethered to this origin through a set of coordinates, they exist as distinct entities. In Tetris, an object is a shape, with its orientation in space defined by coordinates within the program. Coordinates identify how shapes line up in relation to each other within the program's code. The game's program can read the coordinates of several objects in order to make a decision about whether or not the line seen by the gamer is to be removed, freeing up more game space. In addition, in object space, objects can be transformed as their individual coordinates are subject to computations that arise from a gamer's controlling inputs. Knowing the objects constituting space exist as entities that can be separately called and transformed according to both the gamer and the game, it becomes easier to think about space as recursive.

Another important feature through which space is constituted is the camera, the framing device that defines the limits of space visible on the screen from a gamer's viewpoint (Thompson, 2007). Game perspective is often discussed in relation to the immersive qualities of a game or the ways in which a gamer can be understood to enter game space (Giddings & Kennedy, 2006; von Borries, Walz, & Bottger, 2007). Immersion, initially associated with the early work of Janet Murray in Hamlet on the Holodeck (1998) , is effectively described as ‘‘passing through the surface of the image or picture to enter the very space depicted on the surface’’ (ister, ovey, Giddings, Grant, & Kelly, 2008, p. 115). On apparently entering a space defined by a technology, vision is no longer dependent on the position of the human eye but is instead configured through the perspective created by image-making technologies. In digital games, perspective is acknowledged to be a consequence of game design and a facet of the interface. In 3-D games, avatar perspective frequently integrates space, while 2-D games present a coherent view of the space in which the play will take place, for instance, the 2-D versions of Super Mario Brothers or the abstract puzzle Minesweeper (Fernandez-Vara, 2005; Nitsche, 2008; Wolf, 2001). It is easy to solely equate the camera with perspective, but the camera is also an element of the program ensuring that in each render, the imagery is in camera space. When objects are called they are organized and framed so as to be in the appropriate orientation. For instance, when the camera turns right, the objects on the screen move left and are placed in the foreground or distance as necessary. The alignment between perspective and camera space adds to the dimensions of recursive space. Feedback exists between the camera space and what is visible to the viewer. Michael Nitsche argues that camera styles in games are getting more complex, and enabling greater sophistication to emerge in visual styles (Nitsche, 2008). He also comments that cameras ‘‘do not directly affect the action but narrate the event (Nitsche, 2008, p. 112).’’ While cameras do not affect the action, when associated with the moving viewpoint of an avatar, the camera defines both the space of the game and the way in which the viewer is embedded in that game. In this sense, the camera is more than an element through which a gamer enters into the game world, it mediates as the input of the gamer is translated into the reconfigurations of space. For instance, in a first- person game, the camera space ensures that objects are oriented to match the view- point of the avatar, at the same time, the gamer orients the avatar so that the two dimensions of space loop into each other. The potential to see space as recursive is apparent, as space is configured both through the demands of the programming and through the game design, while also controlled by the player.

The way in which the elements of any space-in-frame are called into the frame, how the coded objects are called within the program, is also relevant to thinking about recursive space. It is another instance where the program and gamer's input intersect. epending on the design of the game and the kind of programming used, there are a number of different ways in which objects can be called into active participation in the construction of space. The degrees to which the game and gamer become entangled vary with the ways a program reconfigures the gamer's input in the creation of space. In games developed through object-oriented programming, an object is called forward from a storage system known as dynamic link libraries (Bogost, 2008). An object is a collection of data (such as the model for an avatar), which has associated with it a set of operations or attributes that function in relation to the data (for instance, costume, behaviors, or an action routine). Object-oriented programming is more efficient since some of the elements of the objects are already configured as they wait in a dynamic link library. It avoids the necessity of having to compute the object from scratch each time it is called. In finite state machine programming, objects also have associated with them a state, which determines how sets of attributes are deployed within the game under particular conditions. A state machine is a device that ''remembers'' the status of something at a given time and can respond to or operate on an input to change that status, which is to say, cause an action. Computers are state machines with each instruction an input that changes one or more states that in turn may cause other actions to take place. At any moment in time, a computer system can be seen as a very complex set of states and each program in it as a state machine. The idea of a state machine has become a useful term for game studies. A computer game is an example of a state machine, with each of the characters or active objects being in a state responsive to an input. Jesper Juul describes state machines as follows:

The state machine of the game can be visualized as a landscape of possibilities or a branching game tree of possibilities from moment to moment during the playing of the game. To play a game is to interact with the state machine and to explore the game tree. (Juul, 2005, p. 56)

While Juul is primarily concerned with giving an account of game rules, the idea that an object exists in a state responsive to an input gives us another means for seeing recursive space. For any given frame of action, objects are called and rendered into the appropriate image. But many of the objects within the frame have a state position from which an input generated by a player can trigger a transition. The state of an object is not necessarily obvious to the player, only to the program. In this sense, the input of the gamer is mediated as a further consequence of a game's state. In the later 2000s, finite state machine programming has been a staple means of coding the reactions of non-player character (NPCs). Nonetheless, programming strategies are always under development, with a recent example goal-oriented action planning (GOAP). Diagrammatically, GOAP can also be thought of as a branching tree, but one from which a wider range of possible routes toward a goal become possible. Whether the programming involves scripting, a finite state machine or GOAP, the sense of space that we are beginning to see is not simply one in which play occurs. Instead, we have a recursive space defined as a series of objects whose relationships to one another reconfigure as a consequence of play and the input of the gamer.

4. PLAYING WITH RECURSIVE SPACE

Having a greater understanding that recursive space is configured according to the ways in which a player activates code allows us away of getting a glimpse of the agencies of technology. In a recursive space, the influence of technology is, however, always balanced with a gamer's input. There remains the question of how this interaction can be understood as an intersection between the game and gamer. How does a gamer gain the ability to reconfigure objects, anticipating how the game organizes objects, recursively intervening with an input to which the game then responds? Interactions with game space are understood to be complex, involving the recognition of patterns and kinds of configurations. These approaches are a good place to start thinking about the balance that emerges between game and gamer.

To illuminate the multiplicity of spatial configurations in the game world, Laurie Taylor describes space through the terms striated and smooth (Taylor, 2006). Striated space is delimited and defined, a highly controlled space consisting of predictable objects. Striated space is geometrically understood, already systematically organized through a system of precise definition, a system that can be generalized to other spaces too. By contrast, smooth space is given through relations between the objects that exist within the space. For Taylor, the distinction between striated and smooth space, derived from Gilles eleuze and Fe lix Guattari (1987), is useful to think through how a gamer shifts between different pockets of space, as space is experienced as a mixture between striated and smooth space. Georgia McGregor suggests that gamers relate to game space through recognition of patterns familiar from their existing knowledge of architectural and social spaces. Though she does not use the terms striated and smooth, her analysis also notes controlled and relational spaces:

Game space is based on real space. Video-games display recurrent patterns of spatial use, taken from reality, formalized and altered by the demands of gameplay. Through screen-mediated games these situations of play are made explicit. Each pattern has a particular relationship with gameplay and through this association reveals ways in which gameplay relates to game space. (McGregor, 2007, p. 544)

By invoking patterns of play, she further suggests that these patterns are not set but reinvented according to the different contexts in which they arise, they are relational engagements. Expanding on this point, she links patterns to rules: ‘‘ ules give the game a range of possibility of play, how players actually use that space can vary from what the designer anticipated. Just as real spaces can be used differently from their intended purpose, patterns of game space can change through emergent game-play (McGregor, 2007, p. 539).’’

McGregor is primarily interested in patterns familiar from the actual world, providing a means for embedding engagement with wider social and cultural conventions. The idea of patterns is also helpful in thinking about a gamer’s interaction with code. The ability to reconfigure objects defining space involves recognizing patterns of transformation that emerge from within the game, as a gamer comes to understand what inputs work and in what way. Through play, a gamer learns to anticipate how the code will be called from the way objects appear and move on the screen. In recognizing the ways in which objects appear, the gamer is able intervene in the configurations of the relations between objects. They manage their input to try and work with or against the ways in which the game recursively plays out. At the level of place, the location of interactions between game characters within the game world, any action of an avatar mediated by the gamer has the potential to alter space. Simply moving an avatar changes what is visible in the camera space, so the view within the frame of the screen, the relations between objects, must reconfigure. In a more complex way, inputs can also modify objects within the frame. As a consequence how that object operates to configure space changes too. In Assassin’s Creed, if an NPC target registers the Alta ir avatar attempting to steal from them, the NPC enters into a more reactive mode. This mode has several consequences for the avatar and hence the kinds of input that a gamer can deploy. In a reactive mode, the NPC speaks aloud accusing the avatar of being a thief and so draws attention to them. Further, the space for play is more constrained, as the gamer must be more careful of the proximity in which they locate the avatar in relation to NPCs. The spatial configuration is different in that some parts of the available play space have become harder to negotiate. It is possible to see this scenario in terms of a standoff between characters (suspicious NPC vs. pick-pocketing avatar), but it is equally about access to create space. The poor pick-pocketing skills of the avatar, inputted by the gamer as a series of clumsy movements, in turn configure a space where the range of movements is more limited. To create more space in which to act, the gamer learns to recognize spatial constrictions as they emerge through the interactions between the gamer and game.

Though sometimes used interchangeably with game rules, Miguel Sicart defines game mechanics as ‘‘methods invoked by agents, designed for interaction with the game state’’ (Sicart, 2008). Defining game mechanics as distinct from game rules draws attention to their emphasis on emergent interactions as opposed to possible ones:

Game mechanics are concerned with the actual interaction with the game state, while rules provide the possibility space where that interaction is possible, regulating as well the transition between states. In this sense, rules are modelled after agency, while mechanics are modelled for agency. (Sicart, 2008)

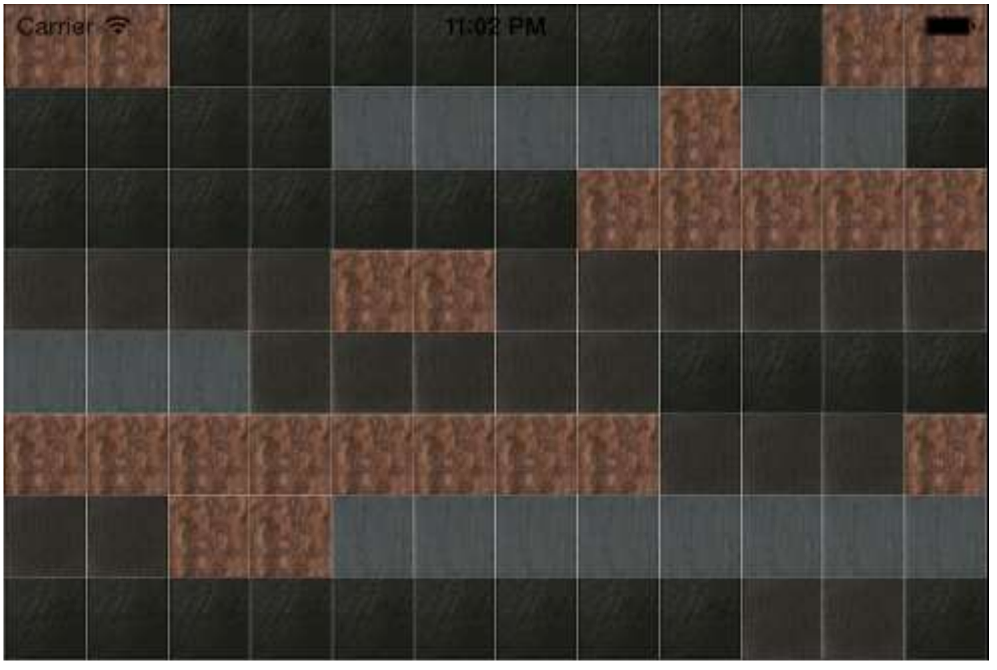
This distinction can be used in relation to recursive space. Rules set up an expectation for the gamer by establishing a set of reliable possibilities. But having a set of reliable possibilities does not determine the outcome; instead, like recursive space game mechanics emerge in the interaction between a gamer and game. Furthermore, game mechanics are available to both human-controlled avatars and NPCs function- ing as intelligent agents. This observation holds for games that use an artificial intelligence (AI) mode of programming in order to create the illusion that intelligent agents exist among the non-player characters that populate the game world. The evolution of A.I. programs, though not exclusive to first-person shooters, is most evident within this category of games. A.I. programming aims to give the impression that enemies are reactive, not simply activated en masse by a trigger point within a game level. We see this in the example above from Assassin's Creed as the NPC alters the state of the game by constraining space, limiting the way in which a gamer can effectively reconfigure objects. When an agent is able to interact with the game mechanic and alter the state of the game, then they too become a component in the reconfigurations of recursive space. The role-playing game Oblivion takes A.I.-based goal- oriented programming further. Each NPC has its own cycle of activities with which the gamer's avatar (chosen and named by the gamer) interacts and depending on the class of the NPC can influence in ways that help or hinder the gamer. This kind of programming creates more complex interactions between a gamer's avatar and NPCs. The game code does not simply provide the framing within which the inputs of the gamer are configured, but the game code too contributes to the recursive quality of space.

Thinking about patterns of play, game rules, and game mechanics gives us a way of thinking about how the gamer's input becomes entangled with the code of the game. Rules primarily refer to what a gamer might expect from their knowledge of games generally as well as the specific game being played. Rules are preformed, more like striated space. In playing, in putting the rules into action, patterns of play and game mechanics emerge as the gamer's engagement with a situation evolves. Both patterns and mechanics are relational, referring to the terrain of the game as it is unfolding, smooth spaces that reconfigure striated ones. Through the idea of recursive space, we can see that gamers not only engage with patterns based on social and cultural ones. They also engage with technological ones as they reconfigure the relationships between objects and so create space.

5. MATHEMATICAL AND PROGRAMMING APPROACH

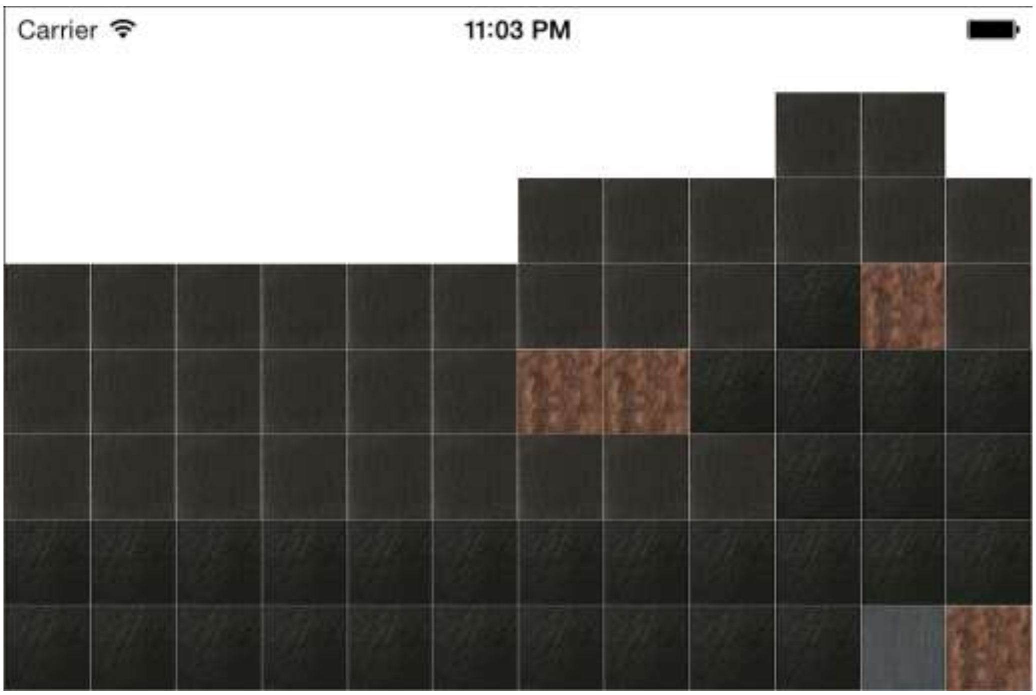
```
//for enable touch
self.isTouchEnabled = YES;
#pragma mark Create Scene
//make scene to play
-(void)addTilesToScene
{
[tilesSpriteArray removeAllObjects];
[deleteTilesSpriteArray removeAllObjects];
```

```
for (int c = 1, yy = 20; c<=8 ; c++, yy+=40) {
int randRow = 0;
int r=1, xx=20;
do {
int rand = (arc4random()%4)+1;
randRow = (arc4random()%(12-randRow))+1+randRow;
for (; r<=randRow; r++, xx+=40) {
//here [NSString stringWithFormat:@"0%d.jpg",rand] generates a random image/Tile name
CCSprite *sprite = [CCSprite spriteWithFile:[NSString stringWithFormat:@"0%d.jpg",rand]
rect:CGRectMake(0, 0, 40, 40)];
sprite.tag = rand;
sprite.position = ccp(xx, yy);
[self addChild:sprite];
[tilesSpriteArray addObject:sprite];
}
} while (randRow<12);
}
}
```




```
//when player touch the screen to remove a group of Tiles object following touch delegate will be called
#pragma mark Touches Delegate
-(void)ccTouchesBegan:(NSSet *)touches withEvent:(UIEvent *)event
{
UITouch *touch = [touches anyObject];
CGPoint touchPoint = [self convertTouchToNodeSpace:touch];
```

```
//here we check that is there a group for removing tiles from screen or not
BOOL success = [self checkForDeletionOfSelectedSprite:touchPoint];
if (success) {
[deleteTilesSpriteArray removeAllObjects];
//remove tiles from touch point
[self removeSpriteAtPosition:touchPoint];
// check that all tiles are removed or not to win or lose
[self isAllSpriteDeleted:touchPoint];
}
}
```

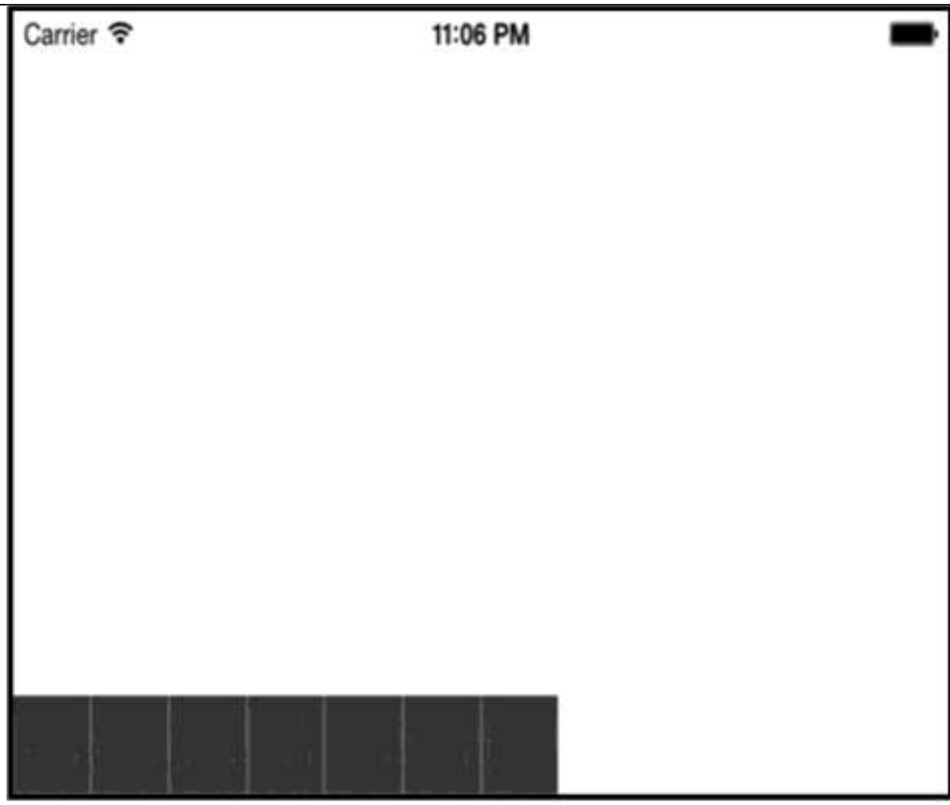



```
//check if selected sprite having group of 3 or not
-(BOOL)checkForDeletionOfSelectedSprite:(CGPoint)touchPoint
{
int nearTiles = 0;
spriteTag = -1;
[deleteTilesSpriteArray removeAllObjects];
for (CCSprite *sprite in tilesSpriteArray) {
[deleteTilesSpriteArray addObject:sprite];
}
//get tag of sprite on touch position
for (CCSprite *sprite in deleteTilesSpriteArray) {
if (CGRectContainsPoint(sprite.boundingBox, touchPoint)) {
spriteTag = sprite.tag;
```

```
[deleteTilesSpriteArray removeObject:sprite];
nearTiles++;
break;
}
}
if (spriteTag != -1) {
nearTiles = [self checkCombinationOfThreeTiles:touchPoint:nearTiles];
if (nearTiles == 3) {
return YES;
}
}
return NO;
}
//Recursive method to delete all tiles in group, that is tiles connected to touched tiles
//if selected sprite having group of 3 then delete all group
-(void)removeSpriteAtPosition:(CGPoint)touchPoint
{
if (!(touchPoint.x >= 0 && touchPoint.x <= 480 && touchPoint.y >= 0 && touchPoint.y <= 320)) {
return;
}
BOOL checkNext = YES;
for (CCSprite *sprite in tilesSpriteArray) {
if (CGRectContainsPoint(sprite.boundingBox, touchPoint)) {
if (spriteTag == sprite.tag) {
[tilesSpriteArray removeObject:sprite];
[self addYPosition:sprite.position.y];
[self removeChild:sprite cleanup:YES];
checkNext = NO;
break;
}
}
}
if (checkNext) {
return;
}
CGPoint leftTouchPoint = CGPointMake(touchPoint.x-40, touchPoint.y);
[self removeSpriteAtPosition:leftTouchPoint];
CGPoint rightTouchPoint = CGPointMake(touchPoint.x+40, touchPoint.y);
[self removeSpriteAtPosition:rightTouchPoint];
CGPoint upperTouchPoint = CGPointMake(touchPoint.x, touchPoint.y+40);
[self removeSpriteAtPosition:upperTouchPoint];
```

```
CGPoint lowerTouchPoint = CGPointMake(touchPoint.x, touchPoint.y-40);
[self removeSpriteAtPosition:lowerTouchPoint];
}
//after every move we have to check that is there any moves available or not to play next turn
-(void)checkMovesAvailOrNot
{

BOOL isMoves = NO;
for (CCSprite *sprite in tilesSpriteArray) {
if ([self checkForDeletionOfSelectedSprite:sprite.position]) {
isMoves = YES;
break;
}
}
if (!isMoves) {
CCScene *scene = [CCScene node];
GameOverScene *layer = [GameOverScene node];
layer.gameMsg = @"No More Moves. :[";
[scene addChild: layer];
[[CCDirector sharedDirector] replaceScene:scene];
}
}
```


Carrier
11:06 PM

6. CONCLUSION

Approaches to space and spatiality in digital games reveal our engagements as multifaceted and multilayered. In playing, gamers construct an experience of spatiality. They are active, participating in creating that experience through a combination of engagements with culturally embedded sound and imagery, a physical relation with a game interface, imaginative connections with the game, its rules, and objects. Immersion, play, and spatiality give us many insights into modalities of engagement.

The idea of recursive space proposes another modality of engagement. It begins from the position that a gamer is not only interacting with complex organizations of space making up the game-world but also with the technology of a game. The agencies of the gamer and game both contribute to play in the model outlined in this article. I draw a connection between the way in which a game technology organizes space as a combination of objects, and how a gamer interacts with space as a combination of objects. From this perspective, space is no longer simply a coherent representation of a game-world in which gamers play but a series of relations between objects. Accordingly, a gamer’s involvement with space is under- stood as interactions that lead to a series of reconfigurations of the objects, which in turn leads to a generation of space. Recursive space is, then, a mode of engagement in which the gamer is both embedded within a space defined by the organization of objects, and also creating that space as they configure the organization of objects. Furthermore, in a digital game, the capacity to interact is technologically mediated, as entering the game involves the gamer-making inputs that feed back into the program. In this way, a gamer’s input becomes entangled with the code of the game as they learn to recognize rules, patterns of play, and game mechanics. The modality of engagement described through recursive space allows us to see that a gamer not only engages with patterns based on social and cultural ones. As a gamer reconfigures the relationships between objects and crate space, they are engaging with the parameters of a technology too. Thinking about play through recursive space offers a way of thinking that reveals our encounters with technological patterns, just as much as social and cultural ones.

REFERENCES

1. Aarseth, E. (1997). Cybertext: Perspectives on Ergodic literature. Baltimore, MD: John Hopkins University Press.
2. Aarseth, E. (2000). Allegories of space: The question of spatiality in computer games. In M. Eskelinen & R. Koskimaa (Eds.), CyberText yearbook 2000 Jyvaskyla (pp. 152–171). Finland: Research Centre for Contemporary Culture.
3. Bogost, I. (2008). Unit operations: An approach to videogame criticism. Cambridge: MIT Press.
4. Calleja, G. (2007). Digital game involvement. Games and Culture, 2, 236–260.
5. Csikszentmihaly, M. (1990). Flow: The psychology of optimal experience. New York: Harper New York.
6. Deleuze, G., & Guattari, F. (1987). A thousand plateaus: Capitalism and schizophrenia. Minneapolis: University of Minnesota Press.
7. Dovey, J., & Kennedy, H. W. (2006). Game cultures: Computer games as new media. Maidenhead, England: Open University Press.
8. Fernandez-Vara, C., Zagal,J., P. and Mateas, M. (2005). 'Evolution of Spatial Configurations in Video Games,' In Changing Views: Worlds in Play. Proceedings of DiGRA 2005, Vancouver.
9. Galloway, A. (2006). Gaming: Essays on algorithmic culture. Minneapolis: Minnesota Press.
10. Giddings, S., & Kennedy, H. (2006). Digital games as new media. In J. Rutter & J. Bryce (Eds.), Understanding digital games (pp. 129–147). London, England: SAGE.
11. Hardwidge, B. (2009). How AI in games works (pp. 1–8). Retrieved July 10, 2009, from http://www.bit-tech.net/gaming/2009/03/05/how-ai-in-games-works/
12. Ja rvinen, A. (2009). Video games as emotional experiences. In B. Perron & M. J. P. Wolf (Eds.), The video game theory reader 2 (pp. 85–108). New York, NY: Routledge.
13. Jenkins, H. (2002). Game design as narrative architecture. In P. Harrington & N. Frup-Waldrop (Eds.), First person (pp. 141–166). Cambridge: MIT Press.
14. Jenkins, H., & Fuller, M. (1995). Nintendo and new world narrative. In S. Jones (Ed.), Cybersociety: Computer-mediated communication and community (pp. 57–72). Thousand Oaks, CA: SAGE.
15. Jenkins, H., & Squire, K. (2002). The art of contested spaces. In L. King (Ed.), Game on: The history and culture of video-games (pp. 65–75). London, England: Barbican.
16. Juul, J. (2005). Half-real: Video games between real rules and fictional worlds. Cambridge: MIT Press.
17. Keogh, J., & Giannini, M. (2004). OOP demystified. New York, NY: McGraw Hill/Osborne.
18. Latour, B. (2007). Reassembling the Social: An Introduction to Actor Network Theory. Oxford: Oxford University Press.
19. Lefebvre, H. (1991). The Production of Space. Oxford: Wiley-Blackwell.
20. Lister, M., Dovey, J., Giddings, S., Grant, I., & Kelly, K. (2008). New media: A critical introduction. London, England: Routledge.
21. McGregor, G. L. (2007). Situations of play: Patterns of spatial use in video-games. Proceedings of DiGRA Conference, Tokyo, 537–545.
22. Murray, J. (1998). Hamlet on the holodeck: The future of narrative in cyberspace. Cambridge:MIT Press.
23. Newman, J. (2004). Video-games. London, England: Routledge.
24. Nitsche, M. (2008). Video game space: Image, play and structure in 3D worlds. Cambridge:MIT Press.
25. Rabin, S. (2005). Introduction to game development. Boston, MA: Charles River Media.

26. Sicart, M. (2008). Defining game mechanics. Retrieved July 19, 2009, from http://gamestudies.org/0802/articles/sicart
27. Stockburger, A. (2007). Playing the third space: Spatial modalities in contemporary game environments. International Journal of Performance Arts and Digital Media, 3, 223–236.
28. Suchman, L. (2007). Human-machine reconfigurations: Plans and situated actions. Cambridge, England: Cambridge University Press.
29. Taylor, L. N. (2006). A ludic model? Smooth and striated space and Sid Meier's Civilization. In online component to M. Bittanti (Ed.), Civilizations: Virtual history, real fantasies. Milan: Edizioni Costa & Nolan. Retrieved from http://www.videoludica.com/graphic/ dynamic/books/pdf/21.pdf
30. Taylor, T. L. (2007). Play between worlds: Exploring online game culture. Cambridge: MIT Press.
31. Thompson, J. (2007). The computer game design course. London, England: Thames and Hudson.
32. Von Borries, F., Walz, S. P., & Bottger, M. (2007). Space/time/play: Computer games, architecture and urbanism. Basle: Birkha user Verlag AG.
33. Walz, S. (2010). Toward a ludic architecture: The space of play and games. Creative Commons, Pittsburgh, PA: ECT Press.
34. Wolf, M. J. P. (2001). Space in the Video Game. In Wolf, M. J. P. (Ed.), The Medium of the Video Game (pp. 51–75). Austin: University of Texas Press.